Original Article

Quantifying the space-efficiency of 2D graphical representations of trees

Michael J McGuffin* and
Jean-Marc Robert

École de technologie supérieure, Montréal,
Canada.
E-mails: michael.mcguffin@etsmtl.ca,
jean-marc.robert@etsmtl.ca

*Corresponding author.

Abstract   A mathematical evaluation and comparison of the space-efficiency of various 2D graphical representations of tree structures is presented. As part of the evaluation, a novel metric called the *mean area exponent* is introduced that quantifies the distribution of area across nodes in a tree representation, and that can be applied to a broad range of different representations of trees. Several representations are analyzed and compared by calculating their mean area exponent as well as the area they allocate to nodes and labels. Our analysis inspires a set of design guidelines as well as a few novel tree representations that are also presented.
Information Visualization (2010) **9,** 115–140. doi:10.1057/ivs.2009.4;
published online 21 May 2009

Introduction

A variety of graphical representations are available for depicting tree structures (Figure 1), from 'classical' node-link diagrams,[1,2] to treemaps,[3–6] concentric circles,[7,8,9] and many others (see[10] for a survey). A major consideration when designing, evaluating or comparing such representations is how efficiently they use screen space to show information about the tree. To date, however, it is unclear how to go about evaluating space-efficiency in a way that can be applied to the large variety of tree representations and that enables a fair comparison of them. Space-efficiency might be described in terms of area, aspect ratio, label size or other measures. However, there is no accepted standard set of metrics for evaluating the space-efficiency of tree representations, and it is unclear what approach would be general enough to be applied to all the forms in Figure 1.

One basic metric of space-efficiency is the total area of a representation. Assuming the representation is bound within a 1×1 square, both icicle diagrams and treemaps (Figures 1(c) and 1(g)) have a total area of 1, and are equally efficient (and both optimal) according to this metric. Likewise, concentric circles and nested circles (Figures 1(e) and 1(f)) both have a total area of $\pi/4 \approx 0.785$ (the area of a circle of diameter 1), and are also equally efficient according to the metric of total area. However, experience suggests that the representations within each of these pairs do not scale equally well with larger, deeper trees. This article shows that there are finer ways of distinguishing efficiency, that is, that there is more to space-efficiency than total area.

Treemaps are often described as optimally space-efficient, not just because they have a total area of 1, but also because they allow for what we call a *weighted partitioning* of the area. Nodes can be allocated more or less area, depending on some attribute such as file size, population or number of species, and furthermore this weighted partitioning can be
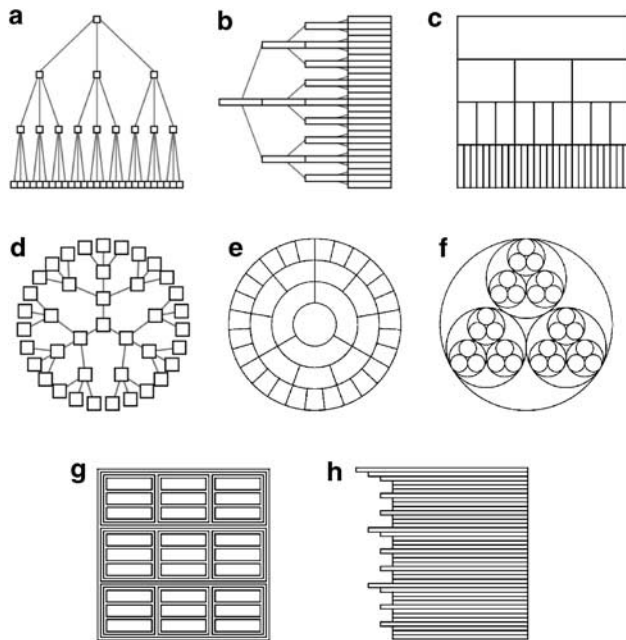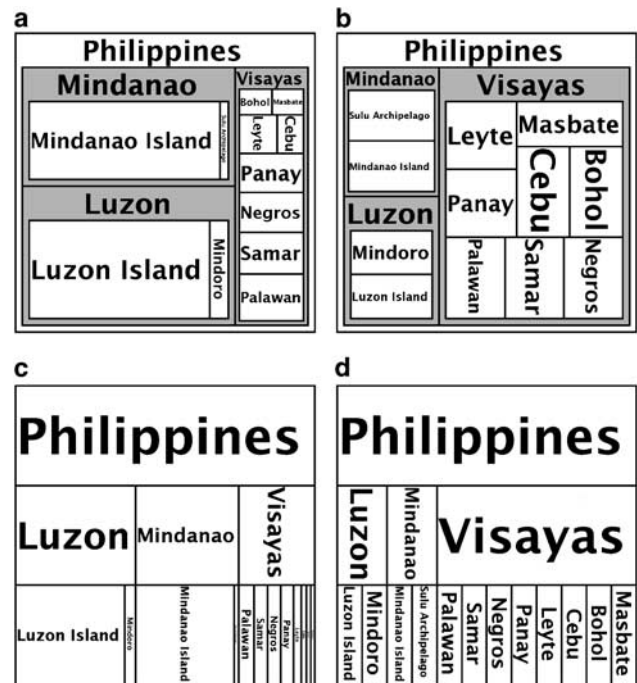
Figure 1: Several basic kinds of tree representations, here each showing a complete 3-ary tree of depth 3 as an example. All representations are drawn to just fit within a 1×1 unit square: (**a**) classical (layered) node-link,[1,2] (**b**) a variation on (**a**) in which the shape of nodes better accommodates long labels, (**c**) icicle, (**d**) radial,[11,12] (**e**) concentric circles,[7,8,9] (**f**) nested circles, similar to,[13,14] (**g**) treemap,[3,4] (**h**) indented outline, sometimes called a 'tree list', and common in file browsers such as Microsoft Explorer.

Figure 2: A tree of regions and major islands of the Philippines, drawn using squarified treemaps (top–(**a**), (**b**)) and using icicle diagrams (bottom–(**c**), (**d**)). The two diagrams on the left (**a**), (**c**) weight leaf nodes by geographic area, whereas the two diagrams on the right (**b**), (**d**) give equal area to leaf nodes. Labels are rotated when necessary to maximize their size.

done without reducing the total area used. These are indeed desirable properties, however they are not unique to treemaps. Figure 2 shows that icicle diagrams also allow for a weighted partitioning of area, and incidentally have no need for margins between the borders of nodes as treemaps often do.

Furthermore, although a weighted partitioning is useful for showing the relative sizes of nodes in Figures 2(a) and 2(c), an unfortunate side effect is that labels on small nodes are very difficult to read. If users are more interested in seeing the identity of all the nodes rather than their relative sizes, an alternative approach would be to give equal weight to each leaf node (Figures 2(b) and 2(d)), improving the overall legibility of nodes. (Although not shown in the figure, the labels could also be augmented to numerically show the 'size' attribute of each node.) In terms of label size or legibility, Figures 2(b) and 2(d) are clearly preferable, but even they still result in much whitespace around certain labels, suggesting that a more space-efficient (in terms of label size) representation might be possible.

Clearly, it would be useful to have some way to quantitatively distinguish the four possibilities in Figure 2, for example, in terms of their respective scalability and the sizes of their labels. If total area is the only metric of space-efficiency used, and 'optimal' space-efficiency is defined as a total area of 1 (possibly partitioned by weight), then we have no way of distinguishing these four cases. If alternative metrics of space-efficiency are used, such as those investigated in this article, it is not clear initially if treemaps, or any other representation, will still turn out to be optimal with respect to such alternative metrics.

This article identifies several metrics related to space-efficiency, and performs the first rigorous analysis and comparison of the space-efficiency of most of the basic tree representation styles in the information visualization literature, including all those in Figure 1. Some of the key ideas involved are (1) the use of a metric of the size of the smallest nodes (that is, the leaf nodes) in the representation, in addition to a metric of total area; (2) analyzing the area of *labels* on the nodes, which implicitly takes into account both the size and aspect ratio of the nodes, measuring how much 'useful' area they contain; and (3) analyzing how these metrics behave asymptotically, as the tree grows arbitrarily deep. Our analysis allows us to rank tree representations by their efficiency, which is useful for helping designers choose the most efficient representation allowable within other given constraints.

Our work also quantifies an interesting difference between representations in how they distribute area

across nodes. For example, the icicle diagrams in Figures 1(c), 2(c), and 2(d) allocate equal area to each level of the tree: the root node has the same area as all the leaf nodes together. Treemaps, in contrast, typically allocate more area to deeper nodes. There is a tradeoff here as we would like users to be able to see as many deep nodes as possible (which tend to also be the most numerous nodes), while at the same time providing some information about shallow nodes (for example, to give an overview of subtrees, and/or to guide the user in zooming operations). This article develops a new metric, the *mean area exponent*, that describes the distribution of area across levels of a tree representation, to quantify this tradeoff.

Finally, we also present a set of design guidelines for using tree representations, as well as a few novel tree representations, including a variation on squarified treemaps that allows for larger labels within the nodes.

Related work

Different tree representations, including classical node-link, icicle, nested enclosure and indented outline, were identified decades ago by Bertin and Knuth,[15,16] and an interactive version of the indented outline representation (now popular in file browsers such as Microsoft Explorer) was presented by Engelbart and English.[17] Subsequent years have seen variations on these representations proposed. Treemaps are a relatively recent innovation, and are a kind of nested enclosure representation. Treemaps are often described as *space-filling*, a highly desirable property for space-efficiency.

However, the term 'space-filling' can sometimes be problematic. For example, a view sometimes expressed[18] holds that tree representations can be divided into two classes: (1) node-link diagrams, that illustrate parent–child relationships with line segments or curves and (2) space-filling representations, which include treemaps and concentric circles such as Sunburst. (Sunburst was described as space-filling by Stasko and Zhang,[8] and Plaisant *et al*[18] similarly describe the work of Andrews and Heidegger[7] as space-filling.) However, these two classes seem to not be disjoint, because some node-link diagrams also 'fill space'.[19,20] The second class also ignores an interesting difference between treemaps and concentric circles, namely that parent nodes in treemaps *enclose* their children, whereas parents in concentric circle diagrams are *adjacent* to their children. Finally, the term 'space-filling' suggests increased space-efficiency, however it is easy to design a treemap layout algorithm that *occupies* all available space without making good *use* of it, for example, by using excessively thick margins, or by concentrating child nodes in only one corner of their parent, leaving the rest of the parent empty and unused. Would such a treemap cease to be considered space-filling, even though its root node covers all the available space? Without a precise definition of 'space-filling', we recommend being cautious about using this term to refer to a category of
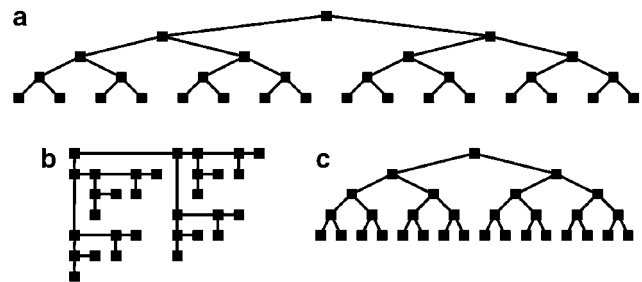
Figure 3: (**a**) and (**b**) are adapted from a comparison in Figure 5 from Anand *et al*,[21] and show two different graphical representations of the same tree in which nodes are constrained to positions with integer coordinates. (**b**) is clearly more compact than (**a**). In (**c**), however, we have redrawn the representation from (**a**) with the integer coordinate constraint relaxed, and the resulting graphical representation has a convex hull whose area is only about 5 per cent greater than that in (**b**). Notice also that the minimum horizontal spacing between nodes in (**b**) and (**c**) is the same, allowing nodes to be overlaid with horizontally oriented labels of the same size in both cases.

tree representations, because the name seems to imply that members of the category are more space-efficient than non-members. As an alternative, categories of representations could instead be based on how the nodes are drawn (for example, representations where the nodes are mapped to points, and those where the nodes are mapped to areas) or on how parent–child relationships are shown (for example, through line segments, enclosure, adjacency or relative positioning). The space-*efficiency* of a given representation can be treated as a separate matter, and evaluated by several metrics, as demonstrated in this article.

Within the graph-drawing community, a common approach for evaluating space-efficiency is to compare the total area required by different drawings (that is, representations) of the same graph or tree. As any drawing can be scaled arbitrarily in x and y, to ensure a meaningful comparison, the 'resolution' of the representations is fixed, often by requiring that nodes be positioned on a grid (that is, with integer coordinates).[12] There are problems with this general approach, however, especially when comparing representations of trees rather than graphs. For example, allowing only grid positions may be misleading, because non-integer coordinates can significantly reduce total area without compromising the clarity of the representation or the space available for labels (Figure 3). As a potential remedy, instead of positioning nodes on a grid, we might instead impose a minimum distance between nodes, or a minimum size for non-overlapping labels centered over the nodes. Unfortunately, matters are complicated by the fact that some tree representations (such as Figures 1(c), 1(e), 1(f), 1(g)) involve nodes that have an area and shape, and there

may be nodes and labels of different sizes within a single representation (for example, deeper nodes may be smaller and have smaller labels). This makes it less clear how to impose a fixed resolution in a way that is fair across tree representations. Note that this issue does not arise in traditional graph drawing, in which nodes are typically mapped to points.

In our work, rather than comparing total area with a fixed resolution, we fix the total area available, and fix its aspect ratio. Representations are drawn within a fixed, bounding 1×1 unit square, and we analyze the resulting area allocated to individual nodes as well as the resulting size of labels. In a mathematical sense, this can be made equivalent to the fixed-resolution approach, but we find our approach simpler. The aspect ratio of typical computer screens, of typical windows and of the human visual field is approximately 1. By imposing the same square-shaped total area on each representation, it is clear that comparisons will be fair, while also allowing the representation to position nodes and subdivide area between nodes in many different ways.

The graph-drawing community has developed algorithms for drawing trees, with nodes positioned on grid points, with optimal total area.[22,23] In the output of these algorithms, however, the relative positioning of sibling nodes varies greatly, making it difficult to perceive the overall tree structure. We suspect these algorithms are unlikely to gain widespread use, and focus on analyzing more conventional, visually uniform, symmetrical depictions of trees.

Several algorithms exist for generating treemap layouts, each of which has advantages and disadvantages in terms of criteria such as stability, preservation of ordering of nodes, aspect ratio of nodes and shape of nodes. One recent example with a mathematical treatment is seen in the work of Wattenberg.[6] All of these layout algorithms allow for a weighted partitioning of the total area (and, interestingly, Wattenberg[6] identifies 'space-filling' with the ability to perform a weighted partitioning). In this article, we do not consider stability or preservation of ordering, although these two criteria are trivially met for many of the tree representations we consider. Unlike the previous work, however, we do examine label area, which implicitly takes into account both the size and aspect ratio of nodes.

The efficiency of 'inclusion' tree layouts (another nested enclosure representation somewhat similar to treemaps) has also been investigated,[24] but not in a way that is easily compared to other kinds of tree representations.

Previous work on labeling drawings of graphs[25] has assumed the labels to be drawn with constant-size text, outside the elements to label. In trees, however, the number of elements to label grows exponentially with depth, and labels placed outside nodes can introduce significant clutter and area requirements. Our work assumes labels will be placed inside nodes, and allows for labels of varying size, corresponding to how nodes in trees are typically labeled in practice.

Metrics Related to Space-Efficiency

We use $n_{i,j}$ to denote the jth node at depth i in the tree. For example, if $n_{i,1}$ has P children, they will be $n_{i+1,1}, \ldots, n_{i+1,P}$; and if the next node $n_{i,2}$ has Q children, they will be $n_{i+1,P+1}, \ldots, n_{i+1,P+Q}$. We also simply use n_i to denote some node at depth i. For example, n_0 and $n_{0,1}$ both denote the root node. Finally, in the 2D representation of the tree, the region corresponding to a node has area $a(n_{i,j})$ or $a_{i,j}$, or simply a_i if all nodes at depth i have the same area.

Total area A

Total area is a basic metric of space-efficiency, which we define here more precisely. To ensure a fair comparison between representations, we require that the 2D representation be bounded by a 1×1 square. The total area A occupied is simply

$$A = \sum_{i,j} a_{i,j} = \sum_{n \, \text{node}} a(n)$$

To ensure $0 \leqslant A \leqslant 1$, we require that no area be counted twice, that is, we subtract away overlap. For example, in representations involving nested enclosure such as treemaps, we require that $a(n)$ be the area corresponding to n after subtracting away any overlapping area assigned to descendant nodes. In a treemap, if node n has thin margins surrounding its children, then $a(n)$ will be the area of the margins, and will not include the children. Not doing this would give an unfair advantage to representations with overlapping areas.

Leaf node area

Another simple metric is based on the size of the smallest nodes in the representation, which typically means the leaf nodes. This metric could be defined as the average (or minimum) area of the leaf nodes in the representation. In the case of a tree where all the leaf nodes are at the same depth D, and they all have the same area, then the metric would simply be $a_D = a(n_D)$.

In typical representations, as a tree grows deeper, eventually the leaf nodes become so small that they are barely visible, and any labels they contain are illegible. The leaf node area $a(n_D)$ can be thought of as a metric of roughly how much zooming the user must perform to make the leaf node labels legible.

Mean area exponent ρ

Next, we define a novel metric ρ that quantifies the distribution of area across levels of a tree representation, independent of the total area A. In this section, we limit our consideration to 'homogeneous' tree representations (with constant branching factor, leaf nodes all at the

same depth, and nodes within the same level having the same area), and in a later section we generalize to other representations.

Consider a complete B-ary tree of depth $D \geqslant 1$, with $B \geqslant 2$. In other words, a tree where every non-leaf node has B children, and the path between every leaf node and the root has D edges. Assume also that nodes within the same level have the same area, that is, for each $i = 0, 1, \ldots, D$, all nodes at depth i have the same area a_i.

We first define a local metric $\rho(p; c)$ associated with a parent node p and one of its children c, or in other words associated with the edge between p and c. We seek a definition of $\rho(p; c)$ that satisfies the following conditions:

$$\rho(p; c) = \begin{cases} -1 & \text{if } a(c) = a(p) \cdot B \\ 0 & \text{if } a(c) = a(p) \\ +1 & \text{if } a(c) = a(p)/B \end{cases}$$

$$\rho(p; c) < 0 \qquad \text{if } a(c) > a(p)$$

$$\rho(p; c) > 0 \qquad \text{if } a(c) < a(p)$$

A value of -1 corresponds to a child occupying B times the area of its parent; a value of 0 corresponds to the child and parent occupying equal area; and a value of $+1$ corresponds to a child occupying $1/B$ the area of its parent. (In this third case, the B children of a parent collectively occupy the same total area as their parent, which is seen, for example, in Figure 1(c).) The definition of the local metric that we will use is

$$\rho(p; c) = \log_B \frac{a(p)}{a(c)}$$

(Intuitively, the use of a logarithm here is related to the fact that trees grow exponentially with depth.)

In the tree we are considering, because all nodes within the same level have the same area, we can associate the same ρ_i with all parent nodes at depth i and their children at depth $i + 1$, where $\rho_i = \rho(n_i; n_{i+1}) = \log_B(a_i/a_{i+1})$. We then define the global metric ρ for the entire tree simply as an arithmetic mean:

$$\rho = \frac{1}{D} \sum_{i=0}^{D-1} \rho_i = \frac{1}{D} \sum_{i=0}^{D-1} \log_B \frac{a_i}{a_{i+1}}$$

We will refer to this global metric as the *mean area exponent* ρ (this name is explained later), and we can rewrite it as

$$\rho = \frac{1}{D} \sum_{i=0}^{D-1} \log_B \frac{a_i}{a_{i+1}}$$

$$= \log_B \sqrt[D]{\prod_{i=0}^{D-1} \frac{a_i}{a_{i+1}}} \qquad (1)$$

$$= \log_B \sqrt[D]{\frac{a_0}{a_D}} \qquad (2)$$
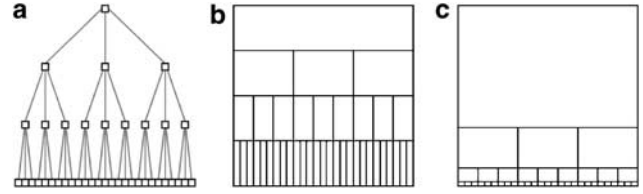
Figure 4: Example representations of a tree with $B = 3$ and $D = 3$. (**a**) Classical node-link, (**b**) icicle, (**c**) a variation on icicle, where each child node is scaled-down with respect to its parent in both the x and y directions by the same factor, allowing for a larger label on the root.

Interestingly, the expression (1) has a simple interpretation: it is the logarithm of the geometric mean of the ratios of areas of nodes on consecutive levels. Expanding the product in expression (1) allows many terms to cancel, yielding the rather simple expression (2) that can be used to compute the metric. As will be seen, such canceling of terms will not be possible in the more general case of heterogeneous tree representations.

Example analysis of Figure 4

To illustrate the calculation of the metrics A, $a(n_D)$ and ρ, we evaluate them in the cases shown in Figure 4.

Because the tree being considered is complete and B-ary, there are B^i nodes at depth i for $0 \leqslant i \leqslant D$, and $(B^{D+1} - 1)/(B - 1)$ nodes in total. Also, for each representation in Figure 4, all nodes at a given depth have congruent rectangular regions. Let w_i, h_i and $a_i = w_i h_i$ be the width, height and area of the rectangular region corresponding to a node at depth i.

In Figure 4(a), because there are B^D leaf nodes, and they are squares, they each have width and height $w_D = h_D = 1/B^D$. As all nodes are of the same size, $a_i = w_i h_i = a_D = w_D h_D = 1/B^{2D}$ and the total area occupied is

$$A = \sum_{i=0}^{D} B^i a_i = \sum_{i=0}^{D} B^i \frac{1}{B^{2D}} = \frac{(B^{D+1} - 1)}{(B - 1)} \frac{1}{B^{2D}} = \Theta(1/B^D)$$

where $\Theta()$ denotes an asymptotically tight bound. (Asymptotic analysis in this article is always with respect to D, that is, expressions that approach zero as $D \to \infty$, with B held constant. If $f(D)$ and $g(D)$ both approach zero as D approaches infinity, then the statement $f(D) = \Theta(g(D))$ means there exist constants $k_1, k_2 > 0$ such that $k_1 g(D) < f(D) < k_2 g(D)$ for all sufficiently large D.) The mean area exponent, which quantifies the distribution of area across levels of the tree, is

$$\rho = \log_B \sqrt[D]{\frac{a_0}{a_D}} = \log_B \sqrt[D]{\frac{1/B^{2D}}{1/B^{2D}}} = 0$$

Next, consider Figure 4(b). Each node has $1/B$ the area of its parent. Clearly, $w_i = 1/B^i$, $h_i = 1/(D+1)$, $a_i = 1/((D+1)B^i)$ and

$$A = \sum_{i=0}^{D} B^i a_i = \sum_{i=0}^{D} B^i \frac{1}{(D+1)B^i} = 1$$

The leaf node area $a(n_D) = 1/((D+1)B^D) = \Theta(1/(DB^D))$, and the mean area exponent is

$$\rho = \log_B \sqrt[D]{\frac{a_0}{a_D}} = \log_B \sqrt[D]{\frac{1/((D+1)B^0)}{1/((D+1)B^D)}} = \log_B \sqrt[D]{B^D} = 1$$

Finally, consider Figure 4(c). Although this representation might not be used in practice, it illustrates a situation where the distribution of area is more heavily towards the root. Each node now has $1/B$ the width of its parent, $1/B$ the height of its parent and $1/B^2$ the area of its parent. Specifically, $w_i = 1/B^i$, $h_i = (B-1)B^{D-i}/(B^{D+1}-1)$ and $a_i = (B-1)B^{D-2i}/(B^{D+1}-1)$. It is easy to check that $A = 1$, $a(n_D) = (B-1)/((B^D)(B^{D+1}-1)) = \Theta(1/B^{2D})$, and $\rho = 2$.

To summarize the results:

| | *Figure* | | |
	4(a)	4(b)	4(c)
total area A	$\Theta(1/B^D)$	1	1
leaf node area $a(n_D)$	$1/B^{2D}$	$\Theta(1/(DB^D))$	$\Theta(1/B^{2D})$
mean area exponent ρ	0	1	2

In terms purely of total area A, Figures 4(b) and 4(c) are preferable to 4(a). However, total area does not help us distinguish Figures 4(b) and 4(c). This distinction is provided by the other two metrics. The representation with the slowest shrinking leaf area (as a function of D) is Figure 4(b). Notice also that Figures 4(a) and 4(c) have the same leaf area (in terms of Θ-complexity, ignoring hidden constants) but different ρ values, because the value of ρ is based on the areas of leaf nodes *and* of the root node. Larger values of ρ mean that more area is allocated to the root. In other words, ρ is a measure of how heavily the representation's total area is weighted toward the root, or of how quickly node area decreases with depth.

Other metrics related to ρ

A deeper understanding of ρ can be achieved by briefly considering some closely related metrics of area distribution. We can define an alternative local metric $R(p; c) = a(p)/a(c)$ that is simply a ratio of areas, and a global metric $R = (\prod_{i=0}^{D-1} a_i/a_{i+1})^{1/D}$ that is the geometric mean of these ratios. It is easy to check that $\rho = \log_B R$, that is, $B^\rho = R$, which motivated calling ρ the mean area *exponent*. Furthermore, whereas R and ρ are defined in terms of ratios of areas, we can define analogous metrics that are ratios of lengths. The four resulting global metrics R, ρ, S

and σ are summarized hence:

	ratios	*logarithms of ratios*
2D area	mean area ratio $R = B^\rho$	mean area exponent $\rho = \log_B R$
1D length	mean scale factor $S = B^\sigma = \sqrt{R}$	mean scale exponent $\sigma = \log_B S = \rho/2$

These four metrices all measure the distribution of area in the tree representation. Of these, $S = \sqrt{R}$ has perhaps the most familiar geometric interpretation, because it is simply an average scale factor. However, we prefer to use ρ in our analyses, firstly because it is more often independent of the particular branching factor B than R or S are, and secondly because ρ is more often an integer than σ (owing to the fact that we are analyzing 2D tree representations).

Incidentally, we also experimented with using an *arithmetic* mean of ratios, that is, $(1/D)\sum_{i=0}^{D-1} a_i/a_{i+1}$, and found that such a formulation resulted in less elegant properties than will be presented in the next section.

Generalizing ρ to heterogeneous tree representations

We now extend the definition of ρ to complete B-ary trees where the areas of sibling nodes are not equal. (Later we extend to incomplete trees with variable branching factor.) In such a tree, let $n_{i,j}$ denote the jth node at depth i, where $0 \leqslant i \leqslant D$ and $1 \leqslant j \leqslant B^i$. The children of $n_{i,j}$ are $n_{i+1,(j-1)B+1}, \ldots, n_{i+1,(j-1)B+B}$. We define the value of ρ associated with any non-leaf node as

$$\rho(n_{i,j}) = \frac{1}{B} \sum_{k=1}^{B} \rho(n_{i,j}; n_{i+1,(j-1)B+k})$$

This is simply the arithmetic mean of the local $\rho(p; c)$ defined earlier, over all the node's children.

Next, the value ρ_i characterizing the distribution of area across the levels i and $i+1$ of a tree representation is itself an arithmetic mean of the previous quantity:

$$\rho_i = \frac{1}{B^i} \sum_{j=1}^{B^i} \rho(n_{i,j})$$

Finally, the global metric ρ is defined as

$$\rho = \frac{1}{D} \sum_{i=0}^{D-1} \rho_i \qquad (3)$$

Combining these three equations yields an expression that can be rewritten in a few different ways. (In reading the following, it helps to keep in mind that i is an index

over levels, j an index over parents in a given level and k an index over children.)

$$\rho = \frac{1}{D}\sum_{i=0}^{D-1}\frac{1}{B^i}\sum_{j=1}^{B^i}\frac{1}{B}\sum_{k=1}^{B}\log_B\frac{a_{i,j}}{a_{i+1,(j-1)B+k}} \qquad (4)$$

$$= \frac{1}{D}\sum_{i=0}^{D-1}\frac{1}{B^i}\sum_{j=1}^{B^i}\log_B\sqrt[B]{\prod_{k=1}^{B}\frac{a_{i,j}}{a_{i+1,(j-1)B+k}}} \qquad (5)$$

$$= \frac{1}{D}\sum_{i=0}^{D-1}\frac{1}{B^i}\sum_{j=1}^{B^i}\log_B\frac{a_{i,j}}{\sqrt[B]{\prod_{k=1}^{B}a_{i+1,(j-1)B+k}}} \qquad (6)$$

$$= \frac{1}{D}\sum_{i=0}^{D-1}\log_B\frac{\sqrt[B^i]{\prod_{j=1}^{B^i}a_{i,j}}}{\sqrt[B^{i+1}]{\prod_{j=1}^{B^{i+1}}a_{i+1,j}}} \qquad (7)$$

$$= \log_B\sqrt[D]{\frac{a_{0,1}}{\sqrt[B^D]{\prod_{j=1}^{B^D}a_{D,j}}}} \qquad (8)$$

Each of the above expressions has a fairly simple interpretation. The radical in expression (5) is the geometric mean of the ratio of the (i,j)th node's area and each of its children's areas. The inner ratio in expression (6) is between the (i,j)th node's area and the geometric mean of its children's areas. The inner ratio in expression (7) is between the geometric mean of areas of all parents on level i and the geometric mean of areas of all children on level $i+1$. Expanding expression (7) results in many terms canceling, yielding expression (8), which involves a ratio between the root's area and the geometric mean of the leaf areas. Note that an alternatively defined metric, based on *arithmetic* means of areas or *arithmetic* means of ratios, would not lead to the kind of multiple, equivalent expressions shown above.

Notice that in equation (3) (and expression (7)), each level's ρ_i contributes with equal weight, even though deeper levels have many more nodes. The root's area has a much larger impact on ρ than any of the B^D leaf nodes. To make the unequal weighting of nodes more apparent, we can rewrite equation (4) as

$$\rho = \frac{1}{DB^D}\sum_{i=0}^{D-1}\sum_{j=1}^{B^i}\sum_{k=1}^{B}B^{D-(i+1)}\log_B\frac{a_{i,j}}{a_{i+1,(j-1)B+k}} \qquad (9)$$

that is simply the *weighted* arithmetic mean of the local metric $\rho(n_{i,j}; n_{i+1,(j-1)B+k})$, where the local metric for an

edge between levels i and $i+1$ has weight $B^{D-(i+1)}$, and the sum of the weights is DB^D.

There is another way we can define ρ, that is equivalent to the previous definitions, but that is more easily extended to other kinds of trees such as incomplete trees. In equation (9), the weight $B^{D-(i+1)}$ assigned to an edge between levels i and $i+1$ is equal to the number of paths through that edge that connect the root node to a leaf node. If we consider all paths between the root and leaf nodes, and for each path we find the arithmetic mean of $\rho(p,c)$ along that path, and then find the (unweighted) arithmetic mean over all the paths, we end up with the same quantity ρ defined earlier. The weighting of edges seen in equation (9) arises 'for free' because many paths pass through shallow edges, causing them to contribute many times.

To make this precise, let path(a,b) be the set of nodes along the path from node a to node b, and let $p(n)$ be the parent of node n. If n is one of the B^D leaf nodes, then path(n_0,n) is the path from the root to leaf n. We define

$$\rho = \frac{1}{B^D}\sum_{n\ \text{leaf}}\frac{1}{D}\sum_{m\in\text{path}(n_0,n)\setminus\{n_0\}}\log_B\frac{a(p(m))}{a(m)} \qquad (10)$$

Not only is equation (10) equivalent to equation (9), it is also straightforward to generalize to incomplete trees with varying branching factor. Given a node n, let $B(n)$ be the branching factor at n, that is, the number of children of n, and let $D(n)$ be the depth of n, that is, the number of edges in the path from n to the root n_0. Also, let $\mathscr{L}$ be the set of leaf nodes in the tree. Then we define a more general mean area exponent as

$$\rho=\frac{1}{|\mathscr{L}|}\sum_{n\in\mathscr{L}}\frac{1}{D(n)}\sum_{m\in\text{path}(n_0,n)\setminus\{n_0\}}\log_{B(p(m))}\frac{a(p(m))}{a(m)} \quad (11)$$

which subsumes the previous definitions as special cases, and which can be applied to a fully heterogeneous tree representation. (A minor caveat: due to the base of the logarithm, ρ is not defined if any node n has only one child, that is, $B(n)=1$. This problem does not arise in our analyses, but if it had to be circumvented, the metrics R or S (in the previous section) could be defined in the fully heterogeneous case without any logarithms, and used instead of ρ.) We note that the canceling of terms that resulted in simpler expressions (2) and (8) cannot be performed here, partly because the logarithm in equation (11) does not have a constant base.

The multiple equivalent formulations and interpretations we have presented show that the mean area exponent ρ exhibits a promising degree of conceptual versatility as a metric.

As a minor additional extension to our definitions, if the root node's area is zero and all the leaf nodes have positive area, then we define $\rho=-\infty$. In the opposite case, if the root node has positive area and all the leaf nodes have zero area, then we define $\rho=+\infty$.
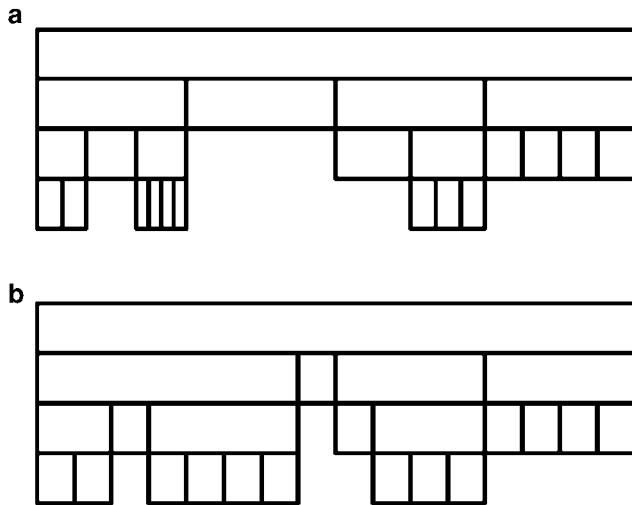
Figure 5: Two icicle diagrams of the same tree. The top diagram (**a**) allocates more of its total area to shallow nodes, as quantified by its higher value of ρ.

We conclude this section with a simple example. Consider the icicle diagram in Figure 5(a), where the width of each parent node is evenly divided among its children. As can be confirmed by the reader, applying equation (11) reveals that $\rho = 1$ (exactly). In Figure 5(b), however, each node is given a width proportional to the number of leaf nodes under it, causing most leaf nodes to be larger than in the former case, hence the metric $\rho \approx 0.884$ is lower.

Additional metrics, and example analysis of Figure 2

So far, our metrics have considered the area of nodes in the representation. In many situations, however, the labels within the nodes (whether they consist of strings, images or some combination) are at least as important to the user as the structure of the tree. The legibility of these labels depends on their size, which is not generally equal to the size of the nodes they appear on. As can be seen in Figure 2, a mismatch in the aspect ratios of a label and its node can leave much whitespace unused within the node. Hence, for each of the three metrics seen so far (total area, leaf node area and mean area exponent), we define a variant based on the areas of labels, resulting in six metrics: the three original metrics based on node areas, and three label-oriented metrics.

Let $a_l(n) \leqslant a(n)$ be the area of (the rectangular bounding box of) the label embedded in the area of node n. The first label-oriented metric we define is the total label area

$$A_l = \sum_{n \text{ node}} a_l(n) \leqslant A \leqslant 1$$

which is of course analogous to total area A.

The second label-oriented metric quantifies the size of the smallest labels in the representation, which typically means the leaf labels (that is, labels on the leaf nodes). This second metric could be defined as the average (or minimum) leaf label area, or in the case of a homogeneous representation, it would simply be $a_l(n_D)$.

The third label-oriented metric is a variant of ρ that quantifies how quickly the labels become illegibly small with depth. This variant ρ_l is defined simply by replacing the areas of nodes with the areas of labels in either equation (11) for the general case or equation (2) for the homogeneous case. For example, in the homogeneous case,

$$\rho_l = \log_B \sqrt[D]{\frac{a_l(n_0)}{a_l(n_D)}}$$

Table 1 lists the values of all six metrics for the four heterogeneous cases in Figure 2. In the table, rows 1–3 are for metrics involving node area, and rows 4–6 are for the analogous label-oriented metrics. Row 5 contains three different metrics of the size of the smallest labels, which will be explained shortly.

The metrics based purely on node area (rows 1–3) are of limited value and can even be misleading. For example, the total area $A = 1$ in all four cases, failing to identify any difference in space-efficiency between them. Also, the average area of leaf nodes is exactly equal between cases 2(c) to 2(d), even though it is plain from Figure 2 that there is a difference in how easily the labels can be read.

We recommend instead using metrics based on labels (rows 4–6). Examining these metrics we notice that, as expected, from Figure 2(a) to 2(b), and again from 2(c) to 2(d), there is an increase in A_l as well as in the average area of leaf labels and also a decrease in ρ_l. These effects are desirable if we want to improve legibility, rather than show the relative sizes of nodes (as is usually done in treemaps, and done in Figures 2(a) and 2(c)).

Comparing all four cases (2(a), 2(b), 2(c) and 2(d)) in terms of label size, we see that 2(d) has the best A_l value, but 2(b) has the best value for average area of leaf labels. The higher A_l value for 2(d) seems to owe largely to the larger labels it has on shallow nodes, which do not contribute to the legibility of the deep nodes. On the one hand, shallow nodes might be considered more important than deep nodes, because they can give the user a high-level overview of the tree structure when there is not enough space to give every individual label a legible size. On the other hand, most nodes in a typical tree are deep, and the shallow nodes only constitute a minority, so making them bigger will impair the legibility of a majority of nodes. Clearly, there is a tradeoff (quantified by ρ_l) in allocating area to shallow or deep labels.

Nevertheless, if our goal is to make *all* the labels as legible as possible (which should be the case at least for static, non-zoomable tree representations), then we should prefer representations in which the smallest labels

Table 1: A comparison of the space-efficiency of representations in Figure 2 according to six kinds of metrics

		2(a)	2(b)	2(c)	2(d)
1	Total area A of nodes	1.000	1.000	1.000	1.000
2	Average area of leaf nodes	0.0465	0.0505	0.0278	0.0278
3	Mean area exponent ρ	0.842	0.462	1.262	0.895
4	Total area A_l of labels	0.366	0.419	0.368	0.591
	Average area of leaf labels	0.0171	0.0219	0.00524	0.0177
5	Average leaf label height	0.0586	0.0791	0.0294	0.0699
	Minimum leaf label height	0.0217	0.0360	0.00821	0.0418
6	Mean area exponent ρ_l for labels	0.755	0.498	1.797	0.871

Each of the six kinds of metrics corresponds to one row. Three instances of the fifth kind of metric are considered, hence the fifth row is split into three sub-rows.

are as large as possible (as quantified in the first sub-row within row 5 of Table 1). We can further refine this idea for *text* labels (as opposed to labels consisting of images): the legibility of a label is not best quantified by its area, but rather by its area per character, or alternatively by its font size (hence the other two sub-rows within row 5, concerning 'label height'). With this in mind, the most space-efficient representation in the table would either be 2(b) (whose leaf labels have the best *average* height) or 2(d) (which has the best *minimum* leaf label height).

Stepping back from the cases in Figure 2, we recapitulate the six kinds of metrics, numbered the same way they are in Table 1:

1. *Total node efficiency*: The amount of available space that is occupied, as quantified by the total area A. Treemaps and icicle diagrams, for example, are optimal in this sense, as $A = 1$ for them.
2. *Smallest node efficiency*: The degree to which the smallest node(s) are made as large as possible. As leaf nodes are typically the smallest, this efficiency can normally be quantified by the average (or minimum) area of leaf nodes. In the case of a tree where all leaf nodes n_D have the same area, it can be quantified by their area $a_D = a(n_D)$.
3. *Node area distribution*: A measure of the area tradeoff between shallow and deep nodes. We quantify this with ρ, although the other global metrics in the section 'Other metrics related to ρ' could play the same role.
4. *Total label efficiency*: The amount of available space devoted to labels, as quantified by the total label area A_l.
5. *Smallest label efficiency*: The degree to which the smallest label(s) are made as legible as possible. This can be quantified by the average (or minimum) area of leaf labels, or, in the case of a tree in which all leaf labels have the same area, by $a_l(n_D)$. In the case of labels that are strings of text of variable length, we can instead quantify this with the average (or minimum) area of leaf labels per character, or the average (or minimum) font height of leaf labels.
6. *Label area distribution*: A measure of the area tradeoff between shallow and deep labels, which we quantify with the mean area exponent ρ_l for labels. A value close to zero indicates that all labels are the same size, which is often desirable in static, non-zoomable representations. However, in zoomable representations of deep trees, a value between 1 and 2 is better (the section 'Observations concerning ρ and ρ_l' discusses why).

For completeness, we discuss and report all six kinds of metrics in this section and the later sections of the article. However, as already mentioned, we recommend using label-oriented metrics (numbers 4–6 in the above list) because they implicitly take into account the aspect ratio of the node area and how well it matches the label's aspect ratio. An important lesson from our work is that only considering total area A, without regard to labels, is a crude way of quantifying space-efficiency, that fails to distinguish between any of the representations in Figure 2.

Within the label-oriented metrics, even A_l alone is not the best way to quantify the space-efficiency of labels, since a large A_l may be because of a few large labels on shallow nodes while most deep labels are very tiny. Metrics of *smallest label efficiency* should arguably take precedence over A_l. Finally, the mean area exponent ρ_l for labels is not a measure of space-efficiency *per se*, but rather a measure of a tradeoff, and will be discussed more in the later sections of the article.

The metrics listed above can each be evaluated for any given instance of a tree's representation (that is, for any given *drawing*). However, we would like to gain insight into the behavior of these metrics for large trees (that is, as $D \rightarrow \infty$), for a variety of different representation styles (or drawing *conventions*, to use graph drawing terminology), as a function of the aspect ratio of the labels involved. The next section performs an analysis of such behavior, which requires some simplifying assumptions, but which yields more general results than could be obtained by analyzing individual instances of representations such as those in Figure 2.

As we will be interested in trends in the performance of a representation for trees whose depth D is large, we finally define

$$A_\infty = \lim_{D\to\infty} A; \quad \rho_\infty = \lim_{D\to\infty} \rho$$
$$A_{l,\infty} = \lim_{D\to\infty} A_l; \quad \rho_{l,\infty} = \lim_{D\to\infty} \rho_l$$

Of course, in practical settings, D is always limited by the size of data sets and the resolution of output media. However, the resolution and physical size of common display devices have increased steadily over recent years, and many tree data sets are deep enough to require some amount of zooming before viewing leaf nodes. It seems reasonable to us to evaluate the limit of various metrics as $D \to \infty$, because representations with better asymptotic efficiency should perform better at a given display resolution and require less zooming.

Asymptotic Analysis of Tree Representations

This section analyses the representation styles in Figure 1 and a few other figures. The analysis considers complete B-ary trees with depth $D \geqslant 1$ and $B \geqslant 2$, in which all nodes at the same depth i have the same area $a_i = a(n_i)$. In addition, we assume that each node contains a rectangular label with aspect ratio $L \geqslant 1$, in which the label must fit within the confines of the node without overlapping descendants, and the label is made as large as possible, rotating the label by an arbitrary angle if necessary to better fit the node. For example, in the leaf nodes of Figure 1(c), labels would be oriented vertically, and in the leaf nodes of Figure 1(e) they would be oriented along radial lines. Note also that L is constant for the entire tree, hence it might be thought of as the maximum length of strings, or the maximum aspect ratio of images.

Of course, in practical settings, real trees seldom have the properties of being complete, with a fixed branching factor B, and labels of fixed aspect ratio. However, these assumptions will allow our analysis to shed significant light on basic differences between the various styles of tree representations.

The results are presented in Table 2. For each representation, the table reports the mean area exponent, area associated with leaf nodes and total area, for both nodes and labels, including limits as $D \to \infty$. Tree representations are also ranked by the various metrics.

The last row in Table 2 lists some additional simplifying assumptions that were used in certain cases. These simplify the expressions appearing in the table, focusing attention on behavior in the most important cases. In almost all cases, it would be an easy matter to remove these assumptions and work out a more complete analysis, however this would clutter the table with more complicated results and would probably not yield any important insights.

In the following subsections, we discuss our choice of the representations that were analyzed, and then present the mathematical details of the analyses, case by case. Readers uninterested in these details may prefer to skip ahead to the discussion section and the later sections, in which the results and their implications are discussed.

Choice of representations to analyze

Initially, we chose to analyze the representations in Figures 1 and 4, because they include key representations from previous visualization literature, as well as some interesting variants. In particular, Figure 1(b) is a variant of Figure 1(a) that better accommodates long labels, Figure 4(c) serves as an example where ρ has a high value (and turns out to have an excellent $A_{l,\infty}$ value, as will be seen), and Figure 1(g) is an 'extreme' treemap in which the margins have zero thickness and labels only appear on leaf nodes. (Although such a treemap is ambiguous for homogeneous trees, treemaps without margins and without labels on non-leaf nodes are used in practice, for example in software such as SequoiaView.)

The initial analysis of this set of representations had two outcomes that led to analyzing additional representations. First, it became apparent that leaf nodes are larger if they are laid out along a longer curve (for example, the leaf nodes in Figure 1(e), and their labels, are larger than those in Figures 1(c)/4(b), because in the former case they are laid out along a curve of length π, and in the latter case they are laid out along a curve of length 1). This led us to invent and analyze the representation in Figure 8, in which leaf nodes are laid out along a curve of length almost 4 (making its leaf nodes and leaf labels even larger), and whose total area A is nearly optimal. Second, it became apparent that treemaps have promising space-efficiency properties. (Although it has long been acknowledged that treemaps are optimal in terms of total area A, it was initially unclear to us how efficient treemaps would be in terms of label size, and our evaluation of multiple metrics clarified this.) It became clear that we should analyze a treemap in which there are margins between nodes and labels on every node, hence we added the case in Figure 9. Finally, we added the case in Figure 10, which is inspired by a kind of fractal, and which we suspected would have excellent space-efficiency and be interesting to compare to treemaps and other representations.

Case-by-case analysis

Case 1(a)/4(a)
The section 'Example analysis of Figure 4' already found the values of ρ, a_i, $a(n_D)$ and A for Figure 1(a)/4(a), from which the values of ρ_∞ and A_∞ follow immediately, allowing us to fill several cells in the first column of Table 2. (Note that the asymptotic expressions in the table sometimes explicitly show the (normally hidden) coefficient of the leading term, for example, showing

Table 2: The space-efficiency of several tree representations. In each row of rankings, 1 means best. Rankings are based first on upper bounds and limits, and then on Θ-complexity, comparing 'hidden' constants (shown explicitly in many cases to aid comparison) to break ties.

Figure	1(a)/4(a) classical (layered)	1(b) classical variant	1(c)/4(b) icicle	4(c) icicle variant	1(d) radial	1(e) concentric circles	1(f) nested circles	1(g) treemap without margins	1(h) indented outline	8 concentric squares	9 treemap with margins and labels	10 Sierpiński carpet fractal
ρ	0	0	1	2	0	$1-\Theta(\frac{1}{D}\log_B D)$	$\rho_\infty - \Theta(\frac{1}{D})$	$-\infty$	$\Theta(\frac{1}{B^D})$	$1-\Theta(\frac{1}{D}\log_B D)$	$\rho_\infty - \Theta(\frac{1}{D})$	$\log_8 9 - \Theta(\frac{1}{D})$
ρ_∞	0	0	1	2	0	1	$2\log_B(1/\phi)\le 2$	$-\infty$	0	1	$\log_B\frac{B+1/L}{1-m}>1$	$\log_8 9 \approx 1.057$
$a(n_D)$	$\frac{1}{LB^{2D}}$	$\Theta(\frac{1}{DB^D})$	$\Theta(\frac{1}{DB^D})$	$\Theta(\frac{B-1}{B B^{2D}})$	$\Theta(\frac{\pi^2}{2B^{2D}})$	$\Theta(\frac{\pi}{2DB^D})$	$\frac{\pi}{4(1/\phi)^{2D}}$	$\frac{1}{B^D}$	$\Theta(\frac{B-1}{B B^D})$	$\Theta(\frac{2}{DB^D})$	$(\frac{1-m}{B+1/L})^D$	$\frac{1}{(B+1)^D}$
Rank	9	5	5	10	8	4	7	1	2	3	6	6
A	$\Theta(\frac{B}{L(B-1)B^D})$	$\Theta(\frac{B}{(B-1)D})$	1	1	$\Theta(\frac{\pi^2 B}{2(B-1)B^D})$	$\pi/4\approx 0.785$	$\pi/4\approx 0.785$	1	$1-\Theta(\frac{K(B-1)D}{B B^D})$	$1-\Theta(\frac{1}{D^2})$	1	1
A_∞	0	0	1	1	0	$\pi/4\approx 0.785$	$\pi/4\approx 0.785$	1	1	1	1	1
Rank	7	5	1	1	6	4	4	1	2	3	1	1
ρ_l	0	0	$2-\Theta(\frac{2}{D}\log_B D)$	2	0	$2-\Theta(\frac{2}{D}\log_B D)$	$\rho_{l,\infty}-\Theta(\frac{1}{D})$	$-\infty$	0	$2-\Theta(\frac{2}{D}\log_B D)$	$\rho_{l,\infty}-\Theta(\frac{1}{D})$	$\log_8 9 - \Theta(\frac{1}{D})$
$\rho_{l,\infty}$	0	0	2	2	0	2	$2\log_B(1/\phi)\le 2$	$-\infty$	0	2	$\log_B\frac{B+1/L}{1-m}>1$	$\log_8 9 \approx 1.057$
$a_l(n_D)$	$\frac{1}{LB^{2D}}$	$\frac{L}{B^{2D}}$	$\frac{L}{B^{2D}}$	$\frac{1}{LB^{2D}}$	$\Theta(\frac{\pi^2}{2LB^{2D}})$	$\Theta(\frac{\pi^2 L}{B^{2D}})$	$\frac{L}{(L^2+1)(1/\phi)^{2D}}$	$\frac{1}{LB^D}$	$\Theta(\frac{L(B-1)^2}{B^2 B^{2D}})$	$\Theta(\frac{16L}{B^{2D}})$	$\frac{1}{L}\left(\frac{1-m}{B+1/L}\right)^D$	$\frac{1}{L(B+1)^D}$
Rank	9	6	6	9	8	5	3	1	7	4	2	2
A_l	$\Theta(\frac{B}{L(B-1)B^D})$	$\Theta(\frac{BL}{(B-1)B^D})$	$\Theta((2-\frac{1}{L})(\frac{B+1}{B-1})\frac{1}{D})$	$A_{l,\infty}-\Theta(\frac{1}{B^D})$	$\Theta(\frac{\pi^2 B}{2L(B-1)B^D})$	$\Theta(\frac{\pi}{D^2})$	$A_{l,\infty}+\Theta((B\phi^2)^D)$	$1/L$	$\Theta(\frac{L(B-1)}{B B^D})$	$\Theta(\frac{4}{D^2})$	$A_{l,\infty}+\Theta(k^D)$	$1/L$
$A_{l,\infty}$	0	0	0	$\frac{B}{(B-1)L}\le 1$	0	0	$\frac{L}{(L^2+1)}\frac{(1-2\phi)^2}{(1-B\phi^2)}<\frac{1}{2}$	$1/L\le 1$	0	0	$\frac{(1-m)^2(B+1)L}{(BL+1)^2-(BL(1-m))^2}<\frac{3}{4}$	$1/L\le 1$
Rank	11	8	5	1	10	7	4	2	9	6		2
Notes				assumes $L\ge 2$			assumes $B\ge 4$	assumes D even		assumes B multiple of 4	assumes D even	$B=8$

Figure 6: Labels are shown in dashed red, enclosed by nodes in solid blue. Labels have a fixed aspect ratio L (equal to 1 in the examples of the bottom row, and greater than 1 in the top row). If the aspect ratio of nodes grows arbitrarily large with depth D (the examples at left), then label area $a_l(n)$ will be proportional to L. On the other hand, if the aspect ratio of nodes stays fixed as D increases (the examples at right), then label area $a_l(n)$ will be inversely proportional to L.

$A = \Theta(B/((B-1)B^D))$ rather than $A = \Theta(1/B^D)$ to aid in comparing the metrics of various representations.)

We must next determine the area $a_l(n_i)$ of a label on a node at depth i. Because each node is a square with area $a(n_i)$, a label with aspect ratio $L \geqslant 1$ within the square will have area $a(n_i)/L$ (see Figure 6, top right and bottom right, for examples). Hence $a_l(n_i) = a(n_i)/L = 1/(LB^{2D})$, and the values of ρ_l, $\rho_{l,\infty}$, $a_l(n_D)$, A_l and $A_{l,\infty}$ all follow by straightforward application of their definitions, completing the entries in the first column of Table 2.

Case 1(b)
In Figure 1(b), as with the previous case, we have nodes all of equal area, and leaf nodes tiling one edge of the unit square (hence $h(i) = 1/B^D$), but now the nodes are rectangles with width $w(i) = 1/(D+1)$, allowing more area to be filled. So $a_i = \frac{1}{(D+1)B^D}$ and

$$A = \sum_{i=0}^{D} B^i \frac{1}{(D+1)B^D} = \frac{(B^{D+1}-1)}{(B-1)} \frac{1}{(D+1)B^D}$$
$$= \Theta\left(\frac{B}{(B-1)D}\right)$$

and $A_\infty = 0$. Also, because all nodes have the same area, we again have $\rho = \rho_\infty = 0$.

To find the area of labels, we note that, unlike the previous case the aspect ratio of nodes grows arbitrarily large as D increases. This is because the height $h(i)$ of each node decreases much faster than the $w(i)$. Hence, labels will look like those in Figure 6, top left and bottom left, and will have an area proportional to L (assuming a sufficiently large D to ensure that the node's aspect ratio is greater than L). In particular, $a_l(n_i) = L(h(i))^2 = L/B^{2D}$. Because all labels have the same size, it follows that

$\rho_l = \rho_{l,\infty} = 0$. Finally

$$A_l = \sum_{i=0}^{D} B^i \frac{L}{B^{2D}} = \frac{(B^{D+1}-1)}{(B-1)} \frac{L}{B^{2D}} = \Theta\left(\frac{BL}{(B-1)B^D}\right)$$

and $A_{l,\infty} = 0$.

Case 1(c)/4(b)
The section 'Example analysis of Figure 4' found the values of ρ, a_i, $a(n_D)$ and A for this case, from which the values of ρ_∞ and A_∞ follow immediately. Similar to the previous case, nodes become arbitrarily 'skinny' as D increases. For sufficiently large D, the labels on leaf nodes will be rotated to be oriented vertically and have area $a_l(n_D) = L(w(D))^2 = L/B^{2D}$, whereas the label on the root node will be oriented horizontally and have area $a_l(n_0) = L(h(0))^2 = L/(D+1)^2$. Hence,

$$\rho_l = \log_B\left(\frac{a_l(n_0)}{a_l(n_D)}\right)^{1/D} = \log_B\left(\frac{B^{2D}}{(D+1)^2}\right)^{1/D}$$
$$= \log_B B^2 - \log_B(D+1)^{2/D} = 2 - \Theta\left(\frac{2}{D}\log_B D\right)$$

and $\rho_{l,\infty} = 2$.

Next, we must calculate A_l, which requires considering carefully the labels that can fit within the nodes. Let $i_1 = \log_B((D+1)/L)$, $i_2 = \log_B(D+1)$ and $i_3 = \log_B(L(D+1))$. If i_2 is an integer, nodes at depth $i = i_2$ are squares. At depth $i < i_2$, nodes are wider than high (that is, $w(i) > h(i)$), and labels are oriented horizontally. At depth $i > i_2$, we have the opposite situation, and labels are rotated to be oriented vertically. And at depth $i < i_1$ or $i > i_3$, the aspect ratio of the nodes is greater than the aspect ratio L of the label. This leads to the following piece-wise expression for label area:

$$a_l(n_i) = \begin{cases} L(h(i))^2 = L/(D+1)^2 & \text{if } i \leqslant i_1 \\ (w(i))^2/L = 1/(LB^{2i}) & \text{if } i_1 \leqslant i \leqslant i_2 \\ (h(i))^2/L = 1/(L(D+1)^2) & \text{if } i_2 \leqslant i \leqslant i_3 \\ L(w(i))^2 = L/B^{2i} & \text{if } i_3 \leqslant i \end{cases}$$

Hence, the summation for calculating A_l can be split into four summations, one for each of the pieces defining $a_l(n_i)$, which we summarize here without the full details:

$$A_l = \sum_{i=0}^{D} B^i a_l(n_i)$$
$$= \sum_{0 \leqslant i \leqslant i_1} B^i \frac{L}{(D+1)^2} + \sum_{i_1 < i \leqslant i_2} B^i \frac{1}{LB^{2i}}$$
$$+ \sum_{i_2 < i \leqslant i_3} B^i \frac{1}{L(D+1)^2} + \sum_{i_3 < i \leqslant D} B^i \frac{L}{B^{2i}}$$
$$\approx \frac{(B+1)(2-1/L)}{(B-1)(D+1)} - \frac{L}{B-1}\left(\frac{1}{(D+1)^2} + \frac{1}{B^D}\right)$$
$$= \Theta\left(\left(2-\frac{1}{L}\right)\left(\frac{B+1}{B-1}\right)\frac{1}{D}\right)$$

The step in the above calculation that involves an approximation is exact if i_1, i_2 and i_3 are integers and D is sufficiently large.

Finally, we clearly have $A_{l,\infty} = 0$.

Case 4(c)
The section 'Example analysis of Figure 4' found the values of ρ, a_i, $a(n_D)$ and A for this case, from which the values of ρ_∞ and A_∞ follow immediately. If we assume that $L \geqslant 2$ (the more typical case), we find $a_l(n_i) = (w(i))^2/L = 1/(LB^{2i})$. It is easy to confirm, then, that

$$A_l = \sum_{i=0}^{D} B^i a_l(n_i) = \frac{B}{(B-1)L} - \frac{1}{(B-1)LB^D}$$

$$= \frac{B}{(B-1)L} - \Theta\left(\frac{1}{B^D}\right)$$

hence, $A_{l,\infty} = B/((B-1)L)$. The values for ρ_l, $\rho_{l,\infty}$ and $a_l(n_D)$ listed in the table are also easy to verify.

Actually, the above results (that are reported in Table 2) do not only hold for $L \geqslant 2$, but also hold more generally if the label's aspect ratio L is greater than or equal to the node's aspect ratio (that is, $L \geqslant (B^{D+1} - 1)/((B-1)B^D) < 2$). In the opposite case, where $1 \leqslant L \leqslant (B^{D+1} - 1)/((B-1)B^D)$, we instead find that $a_l(n_D) = \Theta(L(B-1)^2/(B^2 B^{2D}))$ and $A_l = L(B-1)/B + \Theta(1/B^D)$. The results for this opposite case are omitted from Table 2 for brevity, but are reported in Table 3 in which we need them to rank representations as a function of L.

Case 1(d)
All nodes in this case are squares of equal size. The leaf nodes have their centers on a circle with radius almost $1/2$ and circumference almost π (with this approximation becoming arbitrarily good as D increases). The side length of each node is well approximated by $\pi/((\sqrt{2})B^D)$, where the factor of $\sqrt{2}$ ensures that corners of adjacent leaf nodes need never overlap. It follows that $a(n_i) \approx \pi^2/(2B^{2D})$, from which it is easy to confirm the values of the metrics listed in Table 2 for this case.

Case 1(e)
In a circle of radius r, a sector (that is, a 'pie slice'-shaped region) with angle θ radians has area $(\theta/2\pi)\pi r^2 = (1/2)\theta r^2$. The area of each node in Figure 1(e) can be expressed as the difference in area between two overlapping sectors with the same angle but different radii. The angle corresponding to a node at depth i is $2\pi/B^i$, and the radii of the corresponding sectors are $i/2(D+1)$ and $i+1/2(D+1)$. So the area of the node is

$$a(n_i) = \frac{1}{2}\frac{2\pi}{B^i}\left(\left(\frac{i+1}{2(D+1)}\right)^2 - \left(\frac{i}{2(D+1)}\right)^2\right) = \frac{\pi(2i+1)}{4(D+1)^2 B^i}$$

It is then straightforward to verify that $A_\infty = A = \sum_{i=0}^{D} B^i a_i = \pi/4$, as we should expect from the figure, and also that

Table 3: A comparison of representations relevant for practical applications. Rankings are by $a_l(n_D)$ (1 being best), using A_l to break ties.

	Nested enclosure of nodes			Leaf nodes have an unbounded aspect ratio				Leaf nodes have a fixed aspect ratio		
Figure	9	1(f)	8	1(e)	1(c)/4(b)	1(b)	1(h)	1(d)	4(c)	1(a)/4(a)
	treemap with margins and labels	nested circles	concentric squares	concentric circles	icicle	classical variant	indented outline	radial	icicle variant	classical (layered)
leaf label area $a_l(n_D)$	$\frac{1}{L}\left(\frac{1-m}{B+1/L}\right)^D$	$\frac{L}{(L^2+1)(1/\phi)^{2D}}$	$\Theta\left(\frac{16L}{B^{2D}}\right)$	$\Theta\left(\frac{\pi^2 L}{B^{2D}}\right)$	$\frac{L}{B^{2D}}$	$\frac{L}{B^{2D}}$	$\Theta\left(\frac{L(B-1)^2}{B^2 B^{2D}}\right)$	$\Theta\left(\frac{\pi^2}{2LB^{2D}}\right)$	$1/(LB^{2D})$, $L \geq 2$ $\Theta\left(\frac{(B-1)^2}{B^2 B^{2D}}\right)$, $L = 1$	$\frac{1}{LB^{2D}}$
total label area A_l	$A_{l,\infty} + \Theta(k^D)$	$A_{l,\infty} + \Theta((B\phi^2)^D)$	$\Theta\left(\frac{4}{D^2}\right)$	$\Theta\left(\frac{\pi}{D^2}\right)$	$\Theta\left((2-\frac{1}{L})(\frac{B+1}{B-1})\frac{1}{D}\right)$	$\Theta\left(\frac{BL}{(B-1)B^D}\right)$	$\Theta\left(\frac{L(B-1)}{B\ B^D}\right)$	$\Theta\left(\frac{\pi^2 B}{2L(B-1)B^D}\right)$	$A_{l,\infty} \pm \Theta\left(\frac{1}{B^D}\right)$	$\Theta\left(\frac{B}{L(B-1)B^D}\right)$
total label area $A_{l,\infty}$ as $D \to \infty$	$\frac{(1-m)^2(B+1)L}{(BL+1)^2-(BL(1-m))^2} < \frac{3}{4}$	$\frac{L}{(L^2+1)}\frac{(1-2\phi)^2}{(1-B\phi)^2} < \frac{1}{2}$	0	0	0	0	0	0	$\frac{B}{B-1}L \leq 1, L \geq 2$ $\frac{B}{B-1} < 1$, $L = 1$	0
Rank if $L > 5$ ("skinny" labels)	1	2	3	4	5	6	7	8	9	10
Rank if $L = 1$ ("squarish" labels)	1	2	3	4	6	7	9	5	8	7
Allows all leaf labels to be oriented the same way, for easier reading	YES (see text for qualifications)	YES	no	no	YES	YES	YES	YES	YES	YES
Allows weighted partitioning without reducing total area A	YES	YES	YES	YES	YES	no	no	no	YES	no

$a(n_D) = \Theta(\pi/(2DB^D))$. Next, we find

$$\rho = \log_B\left(\frac{a(n_0)}{a(n_D)}\right)^{1/D} = \log_B\left(\frac{B^D}{2D+1}\right)^{1/D}$$
$$= 1 - \log_B(2D+1)^{1/D} = 1 - \Theta\left(\frac{1}{D}\log_B D\right)$$

Unlike the previous cases where ρ is a constant, here ρ is a function of D, with $\rho < 1$. Evaluating the limit, we find $\rho_\infty = 1$.

Next we analyze the size of labels. For the label on the root node, we make use of the fact that the area of a rectangle (such as a rectangular label) of aspect ratio $L \geqslant 1$ inscribed within a circle of radius r is $4Lr^2/(L^2+1)$. As the radius of the root node is $1/(2(D+1))$, we deduce that the area of the root node's label is $a_l(n_0) = L/((L^2+1)(D+1)^2)$. As for leaf labels, we know that if D is large, the leaf nodes can be well approximated as skinny rectangles that are much longer than wide, with width $(D/(D+1))(\pi/B^D)$. Hence $a_l(n_D) \approx L\pi^2/B^{2D}$ for large D. Calculating ρ_l and $\rho_{l,\infty}$ is then straightforward.

To find A_l, we first note that A_l is bounded below by the area of the root label $a_l(n_0) = L/((L^2+1)(D+1)^2) = \Theta(1/D^2)$. Next, an upper bound on A_l can be found by approximating all nodes (except for the root) with rectangles containing appropriately oriented labels. We can then find an approximating upper bound on $a_l(n_i)$ defined in four pieces, closely analogous to that used in the section 'Case 1(c)/4(b)'. Once again, we can split A_l into four summations, to find an upper bound on A_l that is $\Theta(1/D^2)$. Because the lower and upper bounds have the same Θ-complexity, this establishes the asymptotically tight bound of $A_l = \Theta(1/D^2)$. Furthermore, the detailed calculations involved in the approximating upper bound suggest that the hidden constant is π, hence A_l is given as $\Theta(\pi/D^2)$ in Table 2.

Note that, although the value of the hidden constant for A_l may only be approximate, its value only makes a difference when comparatively ranking representations with the same complexity. In Table 2, the only other representation whose A_l has the same complexity is that in Figure 8, whose A_l value was found with an approximation so similar to Figure 1(e)'s that any error in the hidden constants would not change their relative ranking.

Case 1(f)

In this representation, we assume that the circles corresponding to sibling nodes are laid out with their centers in a circle, so they are inside, and tangent to, the parent node's circle.

It can be shown that the ratio $\phi < 1$ of a node's circle's radius to its parent's circle's radius is

$$\phi = \frac{\sin(\pi/B)}{1 + \sin(\pi/B)}$$

(This is shown in Koike and Yoshihara[26]). Note that as $B \geqslant 2$, we have $0 < \phi \leqslant 0.5$. As the radius of the root node's

circle is 0.5, the radius $r(i)$ of a node at depth i is $r(i) = 0.5\phi^i$, and its area is

$$\pi(r(i))^2 = \frac{1}{4}\pi\phi^{2i}$$

Recall that $a(n_i)$ is defined as the area of the node after subtracting away the overlapping area of descendants, which in this case means subtracting away the area of the B children whenever $i < D$:

$$a(n_i) = \begin{cases} \frac{1}{4}\pi\phi^{2i} & \text{if } i = D \\ \frac{1}{4}\pi\phi^{2i}(1 - B\phi^2) & \text{otherwise} \end{cases}$$

It is easy to check that $A = A_\infty = \pi/4$. The mean area exponent is

$$\rho = \log_B\left(\frac{a(n_0)}{a(n_D)}\right)^{1/D} = \log_B\left(\frac{1 - B\phi^2}{\phi^{2D}}\right)^{1/D}$$
$$= 2\log_B\frac{1}{\phi} - \frac{1}{D}\log_B\frac{1}{1 - B\phi^2} = 2\log_B\frac{1}{\phi} - \Theta\left(\frac{1}{D}\right)$$

As with the previous case, ρ is a function of D, now with $\rho < 2$. We note that $\phi \approx \pi/B$ for large B, and

$$\rho_\infty = \lim_{D\to\infty}\rho = \log_B\frac{1}{\phi^2} = 2\log_B\left(1 + \frac{1}{\sin(\pi/B)}\right) \leqslant 2$$

The formula used in the previous subsection (Case 1(e)), for the area of a rectangle inscribed within a circle, can again be applied to show that the area of leaf labels in this case is $a_l(n_D) = L/((L^2+1)(1/\phi)^{2D})$. For labels on non-leaf nodes, we simplify calculations by assuming that labels are inscribed within a circle that is itself inscribed within the child nodes (Figure 7). The circle inscribed within the children, that is, the dashed red circle in Figure 7, has radius $r(i) - 2r(i+1) = \phi^i(0.5 - \phi)$, hence the label inscribed within it has area $a_l(n_i) = 4L\phi^{2i}(0.5 - \phi)^2/(L^2+1)$. This is not the optimal label size, but is an excellent approximation of the optimal size for large B. It is then a straightforward exercise to calculate ρ_l, $\rho_{l,\infty}$, A_l and $A_{l,\infty}$.

(The results for case 1(f) in Table 2 assume $B \geqslant 4$, as indicated in the last row of the table. For $B = 2$ and $B = 3$, a different choice of label position makes the non-leaf labels significantly larger, but this would not change $a_l(n_D)$ nor the upper bound of $A_{l,\infty} \leqslant 1/2$ that we give in Table 2, and therefore would not change the rankings given in Table 2.)

Case 1(g)

The representation shown in Figure 1(g) is a slice-and-dice treemap[3]. The figure shows thin margins around each set of child nodes for illustration purposes. However, here we consider the extreme case where the margins have thickness zero and can be ignored. Because a_i is defined as the area of a node after subtracting away the area of descendant nodes, the leaf nodes are the only nodes with
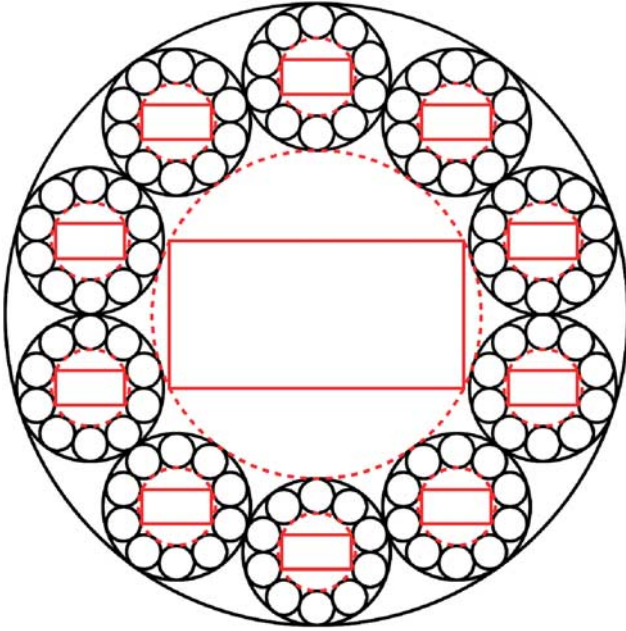
Figure 7: The representation from Figure 1(**f**) with $B = 10$ and $D = 2$. Nodes are shown in black. Red rectangles, inscribed within the dashed red circles, show where we assume the labels are for the root node and for the root's children. The labels' aspect ratio $L = 2$ in this example. (For simplicity, labels on leaf nodes are not shown.)

non-zero area, and the leaf nodes furthermore cover all the available area:

$$a_i = \begin{cases} 1/B^D & \text{if } i = D \\ 0 & \text{otherwise} \end{cases}$$

It follows that $A = A_\infty = 1$. Also, because $a_0 = 0$ and $a_D > 0$, we have $\rho = \rho_\infty = -\infty$.

As indicated by $\rho = -\infty$, the treemap allocates all area to its leaf nodes. We could of course modify the treemap to have margins, which would increase the value of ρ. We consider such a case later.

The analysis of label area is simple if we assume D is even (ensuring leaf nodes are square), yielding the results in Table 2. For D odd, the results are slightly different, and omitted for brevity.

Case 1(h)
This representation corresponds to the 'indented outline' tree views used in many file browsers and other software. Nodes are rectangles of constant height, stacked in a single column. The top-to-bottom ordering of rectangles corresponds to a depth-first traversal of the tree. Each rectangle is indented to the right by a distance proportional to its depth.

The height of each node is $h(i) = (B-1)/(B^{D+1} - 1)$. Let the indentation per level be $Kh(i)$, a constant multiple of the height, where $K > 0$. Then the width of the nodes is
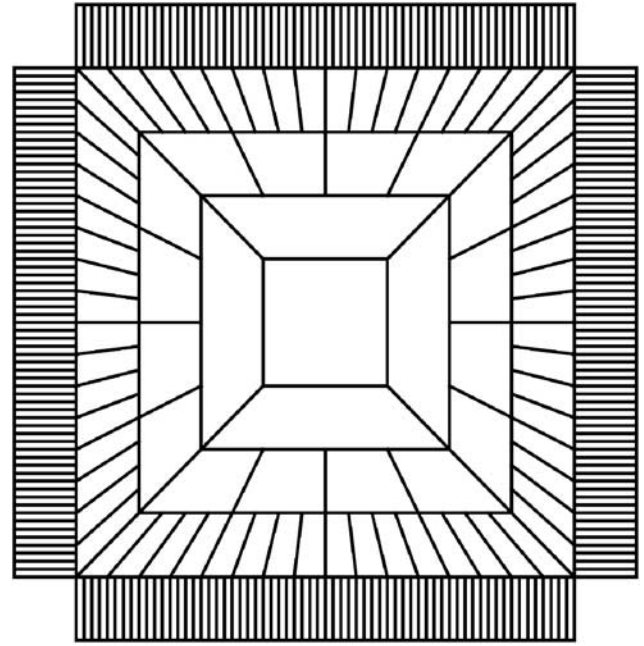
Figure 8: A concentric squares representation, with $B = 4$ and $D = 4$. Notice that $A < 1$ because of the small square regions in the corners that are not used.

$w(i) = 1 - iKh(i)$, and

$$a_i = \frac{(B-1)(1 - iKh(i))}{B^{D+1} - 1} = \frac{(B-1)(B^{D+1} - 1 - K(B-1)i)}{(B^{D+1} - 1)^2}$$

We omit the details of calculating A, and simply give the result:

$$A = 1 - \frac{KB(DB^D(B-1) - B^D + 1)}{(B^{D+1} - 1)^2} = 1 - \Theta\left(\frac{K(B-1)D}{B \, B^D}\right)$$

hence $A_\infty = 1$. Also, it is straightforward to show that

$$\rho = -\frac{1}{D}\log_B\left(1 - \frac{KD(B-1)}{B^{D+1} - 1}\right) \geqslant 0$$

Next, consider that for any quantity x that approaches zero, we have $\log_B(1 + x) = \Theta(x/\ln B) = \Theta(x)$. This fact can be used to simplify the above expression for ρ, yielding $\rho = \Theta(1/B^D)$. We also have $\rho_\infty = 0$.

Finally, it is straightforward to also calculate the metrics related to label area, reported in Table 2.

Case 8
One result of our analysis, which we noticed early on is that the leaf nodes (and labels) in Figure 1(e) are larger than those in Figure 1(c), because the leaf nodes are laid out along a curve of length π rather than a curve of length 1. This led us to invent the representation in Figure 8, where leaf nodes are laid out along (most of) the four edges of the bounding unit square, having a length

of almost 4. This *concentric squares* representation leaves small corners of the unit square unused, but nevertheless outperforms the Figure 1(e) representation in terms of all the area metrics.

To simplify the analysis of this representation, we assume that B is a multiple of four (note that any B value could be used with concentric squares, albeit perhaps less efficiently). We also assume the concentric squares increase in size with a constant step, that is, the side length of the squares is $(i+1)/(D+1)$ for $i=0, \ldots, D$, analogous to the concentric circles in Figure 1(e). It follows that the area of the root node is $a(n_0) = 1/(D+1)^2$, the area of the four unused corners is equal to $a(n_0)$ and the area of each leaf node is $a(n_D) = 2D/((D+1)^2 B^D)$. We omit the further details of the calculations, which are simple for most of the metrics. The calculation of A_l is non-trivial, but can be done in a very similar manner to how A_l was calculated in the sections 'Case 1(c)/4(b)' and 'Case 1(e)', that is, with a summation split into four parts. The non-leaf nodes are trapezoids in this case, but can again be approximated with rectangles, leading to an upper bound on A_l, as was done in the section 'Case 1(e)', and a lower bound of the same complexity is found by simply considering the root node's label.

Case 9

Because treemaps (for example, Figure 1(g)) seem promising, we analyze a treemap with margins and with labels on all its nodes. We consider the kind of treemap shown in Figure 9, in which labels alternate in orientation from one level to the next, to save space and simplify analysis. Nodes at even depths $0, 2, 4, \ldots$ are squares (hence $w(i) = h(i)$ for even i) and have horizontally oriented labels. Nodes at odd depths are rectangles and have vertically oriented labels. All labels have aspect ratio L. Given i even, the labels at depths i and $i+1$ are the same size, and the size of labels is chosen so that each label at depth $i+1$ matches the height of the nodes at depth $i+2$. The sizes and relative positions of labels and margins within nodes at depths i and $i+1$ are simply scaled down for each of the nodes at depths $i+2$ and $i+3$, respectively. The only exception to this is at $i = D$, where labels are at maximum size within leaf nodes.

Let $m \geqslant 0$ be the small fraction of the height (or width) of each node at even depth that is devoted to margins, with m independent of i or L. Next, consider the labels at depth 0 and 1, and how their dimensions along the vertical axis must add up to the total length $1-m$ not used for margins. There is one label at depth 0, with vertical size $h_l(0)$, and B labels at depth 1, each with vertical size $w_l(0)$ (because they have the same dimensions as the root label, only rotated). Hence, $h_l(0) + Bw_l(0) = 1-m$. As $w_l(0) = Lh_l(0)$, it follows that $h_l(0) = (1-m)/(1+BL)$. More generally, for i even, it can be shown that

$$\frac{h_l(i)}{h(i)} = \frac{1-m}{BL+1}; \quad \frac{h(i+2)}{h(i)} = \frac{w(i+2)}{w(i)} = \frac{1-m}{B+1/L}$$
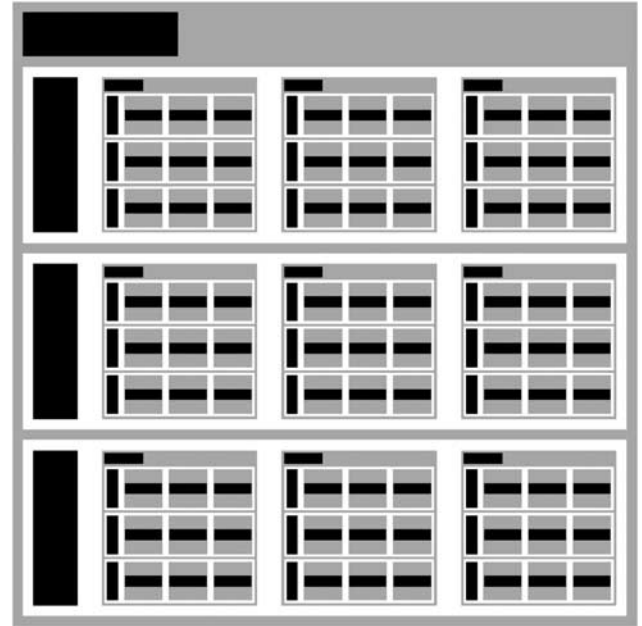
Figure 9: A treemap with margins and labels, with branching factor $B = 3$ and depth $D = 4$. Rather than show actual labels, the bounding rectangles for the labels are shown in black, and have an aspect ratio of $L = 3.5$ in this example. The root node's label is at upper left; there are also $3^4 = 81$ leaf nodes, each with their own label.

The area of each square corresponding to a node at even depth i is

$$w(i)h(i) = (w(i))^2 = \left(\frac{1-m}{B+1/L}\right)^i$$

(Note that the above expression is not equal to a_i, because a_i is defined as the area after subtracting away the area of descendants.) Assuming D even, it is straightforward to confirm the values given in Table 2 for ρ, ρ_∞ and $a(n_D)$, and clearly $A = A_\infty = 1$.

The area of a label is

$$a_l(n_i) = \begin{cases} a_l(n_{i-1}) & \text{if } i \text{ odd or } i = D \\ \frac{1}{L}\left(\frac{1-m}{B+1/L}\right)^{i+2} & \text{otherwise} \end{cases}$$

Assuming D even, the values of ρ_l, $\rho_{l,\infty}$ and $a_l(n_D)$ given in Table 2 follow. Although we omit the demonstration, for D even it can also be shown that

$$A_l = \frac{(1-m)^2(B+1)L}{(BL+1)^2 - (BL(1-m))^2} + k^D\left(\frac{1}{L} - \frac{(1-m)^2(B+1)L}{(BL+1)^2 - (BL(1-m))^2}\right)$$

where $k = (1-m)/(1+1/(BL)) < 1$. (Note that A_l is close to 1 for m small, B large, D small, and $L = 1$.) Evaluating the

limit, we find

$$A_{l,\infty} = \lim_{D \to \infty} A_l = \frac{(1-m)^2(B+1)L}{(BL+1)^2 - (BL(1-m))^2}$$

Keeping in mind that $B \geqslant 2$, $L \geqslant 1$ and $m \geqslant 0$, it can be shown that the above is bounded by $A_{l,\infty} < 3/4$. (Note that $A_{l,\infty}$ is close to 3/4 for m small, B small and L large). Thus, the treemap in Figure 9 fills all available area $(A = 1)$, and has excellent leaf label area with $a_l(n_D) = \frac{1}{L}\left(\frac{1-m}{B+1/L}\right)^D$; but in the limit, as $D \to \infty$, its labels use less than 75 per cent of the available area $(A_{l,\infty} < 3/4)$. In contrast, the treemap in Figure 1(g) also has $A = 1$, has better leaf label area $a_l(n_D) = \Theta\left(\frac{1}{LB^D}\right)$ and its $A_{l,\infty}$ can be as high as 1 (depending on L), but it has the significant disadvantage of having no labels on non-leaf nodes.

As a side note, there is an interesting variation on 1(g) that allows all nodes to be labeled. The idea is to temporarily add an extra child to each non-leaf node, yielding a $(B+1)$-ary tree, and then generate a layout using the representation in Figure 1(g), and then use the space allocated for the temporary children to display labels of their parent nodes. This allows all of the original nodes to be labeled, and yields a leaf label area $a_l(n_D) = (1/L)a(n_D) = \frac{1}{L(B+1)^D}$ (assuming D even). Further analysis also shows that $A_{l,\infty} \leqslant 11/15$ (with $A_{l,\infty} = 11/15$ when $B = 2$ and $L = 3$). The bound on $A_{l,\infty}$ here is not as high as in Figure 9, so we did not include this case in Table 2, but it is interesting that 11/15 is so close to 3/4.

Case 10
Is it possible for a representation to allow labels on all nodes $(a_l(n_i) > 0$ for all $i)$, and not only occupy all available area $(A = 1)$, but also use all available area by filling it with labels $(A_l = A = 1)$? Figure 10 shows an example of a tree representation with these properties for the special case $B = 8$ and $L = 1$. Consider first the case where $B = 8$ and $L \geqslant 1$ is variable. All nodes in the representation are squares, the total area $A = A_\infty = 1$, total label area $A_l = A_{l,\infty} = 1/L$, and the leaf label area $a_l(n_D) = (1/L)a(n_D) = 1/(L(B+1)^D)$ is comparable to that in Figure 9. It can also be shown that $\rho = \rho_l = (D - 1/D)\log_8 9$ and $\rho_\infty = \rho_{l,\infty} = \log_8 9 \approx 1.057$. If $L = 1$, then $A_l = A = 1$, as we sought.

Note that the representation in Figure 10 can be extended to cases where $B = (a^2 - 1)b^2$, for integers $a \geqslant 2$ and $b \geqslant 1$. To see how, imagine a 'generating pattern' for the representation in Figure 10 based on a grid of 3×3 cells, where the central cell is the root node and each of the other eight cells are recursively subdivided. Now, imagine instead a grid of $a \times a$ cells, where one cell is the root node and each of the other $(a^2 - 1)$ cells are further divided into a $b \times b$ smaller cells, each of which is then recursively subdivided. Similar tricks can be used to extend the representation to other values of B if rectangular nodes are used (appropriate for labels with $L > 1$) and/or a small amount of area is left unused after tiling most of the grid with nodes.
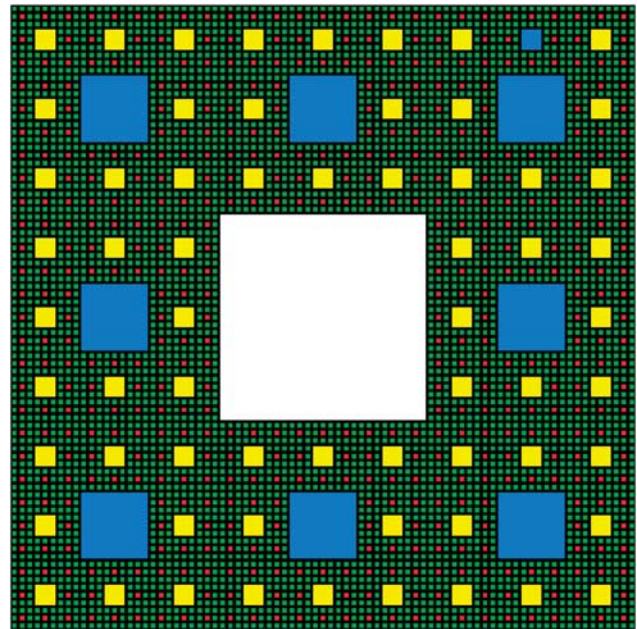
Figure 10: A tree representation with $B = 8$ and $D = 4$ based on the Sierpinski carpet fractal. Each non-leaf node is surrounded by its eight children. The root node is white, surrounded by its eight children in blue, and the leaf nodes are green, surrounding their parents in red. Notice that the bottom two levels consist of nodes of the same size.

Discussion

General observations

Perhaps the most important thing that Table 2 allows us to do is to compare different representations. When Stasko and Zhang[8] described Sunburst (essentially 1(e)) as 'space-filling', with the qualification that it does 'not completely occupy the display space as does the Treemap', they were presumably referring to the fact that $A = \pi/4 < 1$ for 1(e). Our analysis allows us to compare 1(e) with many other representations (including treemaps), not only in terms of total area A, but also according to all the metrics we have considered. Doing so reveals some subtle issues that go beyond the fact that $A < 1$ for 1(e). For example, we could compare 1(e) with 1(c) which, like treemaps, have optimal total area $A = 1$. (1(e) and 1(c) are also interesting to compare because the former is a polar coordinate version of the latter.) We find that 1(e) has leaf nodes that are (asymptotically) $\approx \pi/2$ times larger and leaf labels $\approx \pi^2$ times larger than those in 1(c), despite the fact that both representations must fit within the same 1×1 square, and 1(e) does not even take up all the available space! So, there are clearly some important advantages in 1(e) that might make it preferable over 1(c), despite the fact that 1(e) 'does not completely occupy the display space' like 1(c) does.

It is worth noting how misleading metrics that ignore label size can be. For example, cases 1(c), 1(h) and 8 all

have optimal asymptotic total area $A_\infty = 1$, however their asymptotic total *label* area $A_{l,\infty} = 0$ is as bad as can be. Case 1(h) is a particularly dramatic example: in Table 2, it is ranked second in terms of both leaf area $a(n_D)$ and total area A, and the complexity of its leaf area $a(n_D) = \Theta(1/B^D)$ is as good as can be (ignoring leading coefficients); yet, this representation is ranked ninth in terms of total *label* area A_l. More generally, any representations that have leaf nodes that become arbitrarily 'skinny' with depth D (that is, 1(b), 1(c), 1(e), 1(h), 8) will have leaf area $a(n_D)$ values much larger than their leaf *label* area $a_l(n_D)$, because the mismatch between the aspect ratios of the leaf nodes and labels grows arbitrarily bad with depth D, hence most of the area within each leaf node will not be used by the label. This clearly demonstrates the importance of taking into account label size to evaluate the space-efficiency of tree representations.

One pattern that can be seen in the leaf label area values is that representations with arbitrarily skinny leaf nodes (1(b), 1(c), 1(e), 1(h), 8) have $a_l(n_D)$ proportional to L because they correspond to the situation in Figure 6, top left and bottom left. On the other hand, representations whose leaf nodes have a roughly fixed aspect ratio correspond to the situation in Figure 6, top right and bottom right, and have $a_l(n_D)$ *inversely* proportional to L – this is true, or approximately true, of all the other representations analyzed.

It is also interesting that, within the set of representations with arbitrarily skinny leaf nodes (1(b), 1(c), 1(e), 1(h), 8), the leaf nodes are laid out, or tiled, along a 1D curve. In cases 1(b), 1(c) and 1(h), this curve is an edge of the 1×1 bounding square, and has length 1. In case 1(e), the curve is the circumference of an inscribed circle, and has length π. In case 8, the curve is (most of) the four edges of the 1×1 bounding square, with length almost 4. In all these cases, the labels on the leaf nodes are oriented perpendicular to the direction of the tiling, and as $D \to \infty$ there is plenty of room for the labels in this perpendicular direction. The size of the labels is only constrained along the direction of tiling, and hence the *length* of the curve is the limiting factor for the size of leaf labels. The complexity of $a_l(n_D) = \Theta(L/B^{2D})$ is the same (ignoring leading coefficients) for all these cases, but the leading coefficients reveal that the representations with longer curves have larger $a_l(n_D)$ values. This explains why 1(e) has a better $a_l(n_D)$ value than 1(b) or 1(c), and also why the representation in Figure 8 is the best of the set. In fact, our invention and analysis of representation 8 was motivated by a desire to tile the leaf nodes along a longer curve than in 1(e), as well as a desire to fill more of the corners of the available area than does 1(e). Finally, it is also clear why 1(h) has the worst $a_l(n_D)$ value of the set of representations: it tiles *all* nodes along the same curve of length 1 rather than just tiling leaf nodes.

Certain other representations (1(a), 4(c), 1(d)) also tile their leaf nodes along a curve, but are worse because their leaf nodes have a fixed aspect ratio. Hence, labels are constrained both along the direction of tiling and
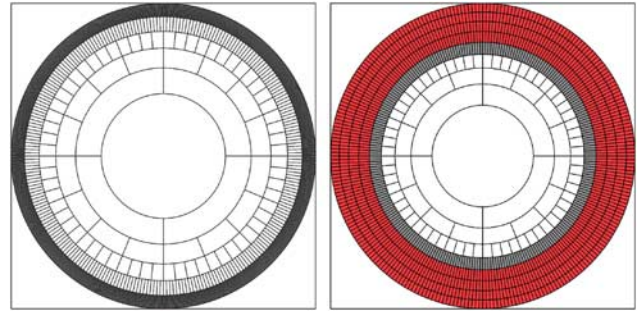
Figure 11: Two different representations with $B = 4$ and $D = 5$. At left is a normal case, with radii of circles chosen so $\rho = 1$. As can be seen, the leaf nodes appearing in the outermost ring are crowded beyond legibility, making the ring appear almost black. At right, nodes on levels 0 through 4 are shown as usual with the central circle and the inner four rings, however the leaf nodes (in red) are then distributed over the subsequent four outer rings. Each set of B leaf nodes with a common parent (in gray) is effectively 'rotated' and laid along a radial line from its parent. This allows leaf nodes to retain a useful size, unlike the case on the left.

perpendicular to it, and $a_l(n_D) = \Theta(1/(LB^{2D}))$ is inversely proportional to L for these representations.

In addition to tiling nodes along a longer curve (Figure 8), another approach we experimented with to create more space for leaf nodes is shown in Figure 11, right. This is a variation on 1(e) where the last level of nodes is 'rotated' to lie along radial lines instead of along a circumference; we call this an *asymmetrical* representation. (Such a rotation of nodes is comparable to the 'folding' in Section 4.1 of Card and Nation[27]). To avoid confusion, in an incomplete tree, such rotation should only be applied to the deepest level of leaf nodes, and not to leaf nodes on more shallow levels.

Comments on 1(h)

The indented outline representation (1(h)) is interesting for a few reasons. Before our analysis, we expected this representation to perform rather well, because it allows nodes to be tiled vertically while also allowing long labels to extend horizontally without introducing any extra space between nodes. Now, in a practical setting, the nodes in such a representation are typically not nearly as wide as the representation is tall; for example, the tree might contain 200 nodes, each of which has a label that is a mere 25 characters long, so the aspect ratio of a rectangle bounding the representation might be $\approx 200/25$. For the sake of our analysis, however, to enable fair comparison with other representations, we imposed the same 1×1 bounding square as with all the representations, and therefore we made the nodes in 1(h) extend horizontally as far as possible to fill the available space. The results partially confirmed our expectations: $a_l(n_D)$

is proportional to L (indicating that labels have room to 'grow' without any penalty) and, as $D \to \infty$, we have $A_\infty = 1$, that is, the space left unused by indentation becomes negligible and the nodes fill all the available space. (Interestingly, 1(h) is the only representation we have found for which $A_\infty = 1$ while all the nodes have asymptotically the same size, that is, $\rho_\infty = 0$). However, somewhat to our surprise, the total space used by labels goes to zero ($A_{l,\infty} = 0$) and 1(h) ranks relatively poorly in terms of leaf label area and total label area. As explained in the previous section, this is because the mismatch in aspect ratios of labels and nodes grows without bound as $D \to \infty$.

Now, what if we had instead modeled this representation as it is typically used in practice, that is, with nodes that are just wide enough to contain their labels instead of extending horizontally as far as possible? The bounding rectangle around the representation would then have been tall and narrow instead of a square, and in the limit (as $D \to \infty$), all of the bounding rectangle's area would be used by the labels. But, in the limit, the aspect ratio of the bounding rectangle would also go to infinity, so if a user zoomed out to fit the entire representation within their (square) screen, most of the available screen space would be wasted. So, the end result of the analysis would be the same, but the analysis would be complicated by having to consider the aspect ratio of the bounding rectangle of the representation. This illustrates why it is simpler and easier to bound all representations within a 1×1 bounding square from the start of the analysis: if a tree representation performs poorly within such a square (that is, $A_{l,\infty}$ is small), there is no point in relaxing the constraint of a bounding *square* and re-analyzing, because the poor performance may simply be re-expressed as a poor aspect ratio of the bounding rectangle, resulting in yet another metric to consider.

Despite the relatively poor space-efficiency we found for 1(h), this indented outline representation is popular in user interfaces such as file browsers. This popularity may partly be because of the fact that when users are using an indented outline, they are typically zoomed in on some subregion of the tree rather than looking at the entire tree, and during scrolling they mostly only need to scroll vertically. It may also have something to do with how users can expand or collapse subtrees, and work their way down a path by clicking on nodes and performing small, similar, repeated mouse motions, for example, motions down and right. Such details are not captured in our analysis (though the incorporation of such interactive aspects would be an interesting direction for future research). However, in examining the results of our analysis, we found a way to improve the space-efficiency of 1(h) (in terms of $a_l(n_D)$), while remaining true to the basic idea of an indented layout. The key was to combine 1(b) with 1(h) in a way that is exactly as space-efficient as 1(b) (Figure 12). Such a hybrid might prove to be useful in interactive software, combining advantages of 1(h) while being more space-efficient.
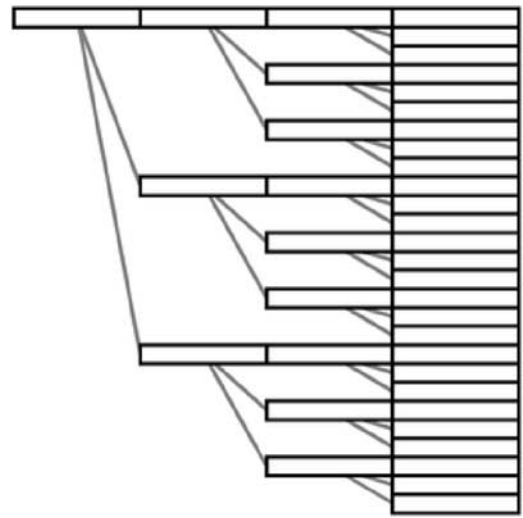
Figure 12: A hybrid of the representations in Figures 1(**b**) and 1(**h**). Each non-leaf node in 1(**b**) is shifted upwards to be aligned with its top-most child. The result imitates the indentation in 1(**h**), but requires no change in node (or label) sizes, hence the labels have the same, larger size they have in 1(**b**).

Optimal space-efficiency?

Examining the rankings by label-oriented metrics in Table 2, we see that the most space-efficient representations are 1(g), 9 and 10. The only representations that allow $A_{l,\infty}$ to achieve a value of 1 are 1(g), 10 and 4(c) (the last of which has one of the worst $a_l(n_D)$ values of all). Figure 1(g) is unique in having the best possible $a(n_D)$, but has the important disadvantage of not allowing labels on non-leaf nodes. Ten has itself the severe disadvantage of only being efficient for certain values of B, such as $B = 8$.

The remaining representation, 9, is perhaps the most space-efficient 'usable' representation. It also has a disadvantage, though arguably less severe: our analysis showed that $A_{l,\infty} < 3/4$ for this representation. Thus, in the limit, it is impossible to fully use all the available space for labels. The value of the bound $3/4$ is perhaps an artifact of the size and placement we chose for nodes and labels in our treemap with margins. However, the fact that there is a bound less than 1 probably reflects a deeper limitation of treemaps with margins, and achieving $A_{l,\infty} = 1$ may be impossible without 'filling up' the space to the right of labels with nodes. It is an intriguing challenge to try to find an alternative way to layout a treemap that would improve on our bound of $3/4$, as this could have both theoretical and practical implications.

A more ambitious goal would be to find a tree representation for which $A_{l,\infty} = 1$ and $a_l(n_D) = \Theta(1/B^D)$, that (unlike 1(g)) allows labels on all nodes. Ideally, such a
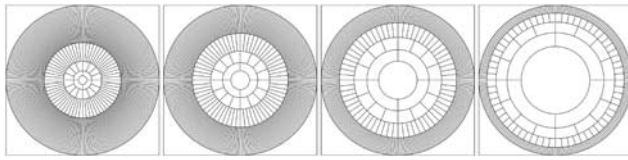
Figure 13: Changing the radii of circles in Figure 1(**e**) allows us to create a representation with any ρ value. Here, $B = 4$, $D = 4$, and from left to right, ρ is 0, 0.33, 0.66 and 1, respectively, gradually allocating more area to shallow nodes.

representation would also be equally efficient for all values of B and L.

One objection that might be raised against seeking optimal space-efficiency is the following: even if some representation can be found that has optimal $A_{l,\infty}$ and $a_l(n_D)$, there will likely be a perceptual benefit in introducing some whitespace or margins between elements of the representation, to help users better perceive appropriate groupings of elements. This is a valid point, however, it would be preferable for the amount of whitespace to be under the control of the user (or the designer) and to be something that can be added *to* the representation (for example in the form of margins), rather than there being a minimal amount of whitespace imposed *by* the representation. Hence we still see value in seeking representations that are optimal, that is, that use all available space.

Observations concerning ρ and ρ_l

Regarding the mean area exponent ρ, we point out that representations can often be adjusted to yield different ρ values. For example, the horizontal lines in 1(c)/4(b) are shifted downward to produce 4(c), changing the heights of nodes and the ρ value from 1 to 2. In an analogous manner, Figure 13 shows how 1(e) can be adjusted to yield different ρ values. Thus, one could imagine exposing ρ as a controllable parameter to a designer or user, to 'tune' a tree representation.

In terms of label-oriented metrics, the most space-efficient representations in Table 2 that allow labels on all nodes (namely, 9, 10 and 1(f), but not 1(g) because it only allows labels on leaf nodes) have $\rho_{l,\infty}$ values between 1 and 2. Representations with $\rho_{l,\infty} < 1$ or $\rho_{l,\infty} \geqslant 2$ are ranked as less efficient.

Why would $1 \leqslant \rho_{l,\infty} < 2$ correspond to more efficient representations? Well, as $D \rightarrow \infty$, the number of leaf nodes increases exponentially, so their size (and label legibility) decreases exponentially. If $\rho_{l,\infty}$ is low ($\rho_{l,\infty} = 0$, for example), then this exponential decrease in size will also happen to labels on shallow nodes, making the representation less efficient in terms of $A_{l,\infty}$. In contrast, if $\rho_{l,\infty} \approx 1$, shallow labels are more likely to remain legible as D increases, which would make a zoomed-out overview of the tree more useful, with shallow labels

guiding the user during zoom-in operations. Finally, if $\rho_{l,\infty}$ is high ($\rho_{l,\infty} \geqslant 2$), then the exponential decrease in size of deep labels will be faster, making the representation less efficient in terms of $a_l(n_D)$. Representations 9 and 10, arguably the two best representations overall, have $\rho_{l,\infty}$ only slightly larger than 1. This is not surprising, because the best leaf label size $a_l(n_D)$ we could hope for would be something slightly less than $1/B^D$, for which we expect $\rho_{l,\infty}$ to be close to 1 (assuming the root label's area $a_l(n_0)$ does not go to zero exponentially fast as $D \rightarrow \infty$). So, $\rho_{l,\infty}$ values close to 1 seem to characterize the most space-efficient representations in terms of label size.

We also note that neither $a_l(n_D)$ nor A_l alone is always a good indicator of space-efficiency. For example, 1(g) has optimal $a_l(n_D)$, but does not allow labels on any non-leaf nodes, and 4(c) has the best $A_{l,\infty}$ value among the representations analyzed, but has a poor $a_l(n_D)$ value. In contrast, the condition $1 \leqslant \rho_{l,\infty} < 2$ consistently corresponds only to the most space-efficient forms we have analyzed. So, if the space-efficiency of a representation is to be assessed with just a *single* metric, it may be feasible to use $\rho_{l,\infty}$ (or ρ_l, whose value as a function implies the value of $\rho_{l,\infty}$) as the metric of choice, rather than $a_l(n_D)$ or A_l. However, if *two* metrics can be used we recommend using $a_l(n_D)$ and A_l (with the former taking precedence over the latter) because together they provide more information than ρ_l alone. This is the approach taken in Table 3, which is described next.

Design Implications and Guidelines

Further comparisons of representations

In Table 3, we have attempted to distill the most important information in Table 2. We eliminated representations 10 (which we judge to be impractical) and 1(g) (which, while used in practice in software such as SequoiaView, has very limited support for labels). Representation 4(c), which we initially analyzed to illustrate a high-ρ case, turns out to have an excellent total label area A_l and could be useful in a zoomable user interface (ZUI),[28] and so is not eliminated in Table 3. We also report only the most important label-oriented metrics in Table 3, and give two rankings for different values of L. In both rankings representations are ranked by $a_l(n_D)$, using A_l only to break ties. In the first ranking, L is assumed to be significantly larger than 1, for example, $L > 5$. In the second ranking the labels are squares (such as image thumbnails) with $L = 1$. There's nothing particularly special about the value 5 in the $L > 5$ case; 5 is simply chosen to make the ranking for that case more clear, as the relative performance of certain representations can be more subtle for values of L between 1 and 5.

Interestingly in the $L > 5$ ranking we find that the best representations are all those involving nested enclosure of nodes, followed by all those with 'skinny' leaf nodes (that

is, leaf nodes with unbounded aspect ratio), followed by all those with 'squarish' leaf nodes (that is, leaf nodes with a fixed aspect ratio). These three categories are indicated in the column headings in Table 3, and representations are arranged left-to-right, from most to least space-efficient, according to the $L > 5$ ranking. In the $L = 1$ ranking, again the best representations are those involving nested enclosure, but after that point the ranking alternates somewhat between the other two categories of representations.

The last two rows in Table 3 contain some additional information of practical value. The second last row concerns the orientation of labels, which can impact readability. The last row indicates which representations lend themselves to a weighted partitioning. As already explained, treemaps are commonly used to create a weighted partitioning of the total area available, however other representations also allow this, such as 1(c) (as shown in Figure 2), or 1(e) and 8 (by changing the angles of wedges according to weight). In other representations, such as 1(d) or 1(a), it would be possible to vary the size of nodes according to their weight, but this might reduce the total area A of the representation, so they do not lend themselves as well to weighted partitioning.

As a side note, the second last row of Table 3 indicates that representation 9 allows all labels on leaf nodes to be oriented the same way. This is not necessarily true if the tree is incomplete (that is, if leaf nodes are not all at the same depth). However, it is easy to modify the treemap in Figure 9 so that *all* labels are oriented horizontally, even if this comes at the cost of a reduction in space-efficiency. Because of this, the information in the second last row is a useful approximation of the truth for treemaps with margins and labels.

Design guidelines

We now enumerate some guidelines that may be useful to designers. For brevity, we use the terms *skinny label* to refer to a label whose aspect ratio L is significantly greater than 1, *squarish label* to refer to a label where $L \approx 1$, *skinny leaf* to refer to a leaf node whose aspect ratio grows arbitrarily large as $D \to \infty$ and *squarish leaf* to refer to a leaf node whose aspect ratio remains fixed or converges to a constant value as $D \to \infty$.

1. *Choose a representation whose leaf node aspect ratio matches the aspect ratio of labels to be used*. If the labels to be placed on nodes are skinny, the designer could look in Tables 2 or 3 for representations whose leaf nodes are skinny (indicated in the tables by $a_l(n_D)$ proportional to L), and consider using those first. On the other hand, if the labels are squarish, the designer could consider first using representations with squarish leaf nodes (indicated by $a_l(n_D)$ inversely proportional to L).
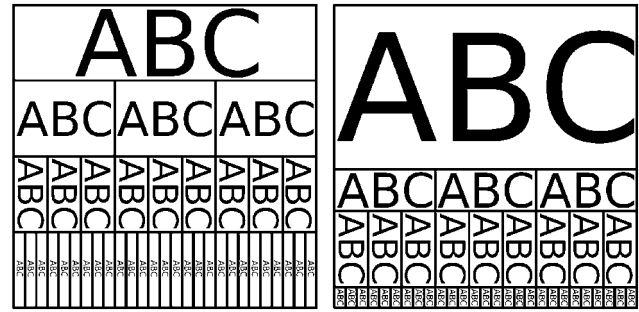
Figure 14: At left, an icicle diagram like Figure 1(c), with labels added. At right, the icicle diagram has been modified to allow shallow labels to be larger, without reducing the size of deep labels.

2. *Modify labels to match the aspect ratio of their nodes*. If a designer has chosen a representation whose leaf nodes are squarish, then labels consisting of long strings of text should be wrapped into squarish multi-line blocks of text. On the other hand, if the representation has skinny leaf nodes, then labels consisting of long strings of text should of course *not* be wrapped into multiple lines.

3. *Modify nodes to match the aspect ratio of their labels*. In the case of representations with skinny leaf nodes, even with skinny labels, eventually most of the available space will not be used by the labels because the leaf nodes' aspect ratio grows without bound. One possible way to create a better match between aspect ratios is shown in Figure 14. The horizontal lines separating levels of nodes are shifted downward as far as possible without making deep labels any smaller. This allows shallow labels to be made larger, increasing A_l. Analogously, the circles and squares separating levels within Figures 1(e) and 8 can be shifted away from the root node to increase A_l without reducing $a_l(n_D)$.

Another example of modifying nodes to match the aspect ratio of labels is presented in the section 'Rectified treemaps', in which nodes in a treemap are 'rectified' rather than squarified to create rectangles whose aspect ratio is closer to the aspect ratio of their labels.

4. *In classical representations, orient labels to be perpendicular to the levels*. It is interesting that 1(a), the most 'classical' of representations, is ranked last in space-efficiency in Table 3 when $L > 5$. Representation 1(b) is significantly superior because it uses up more of the space available between the levels of nodes. Rotating skinny labels so their length is perpendicular to the levels of the tree imitates 1(b) and is more space-efficient. Although there is a strict 1×1 bounding rectangle in Figures 1(a) and 1(b), this guideline also holds when the representation is allowed to have a bounding rectangle that is not 1 × 1: in such a case, applying the guideline simply improves the aspect ratio of the bounding rectangle of the representation. Figure 15 illustrates.
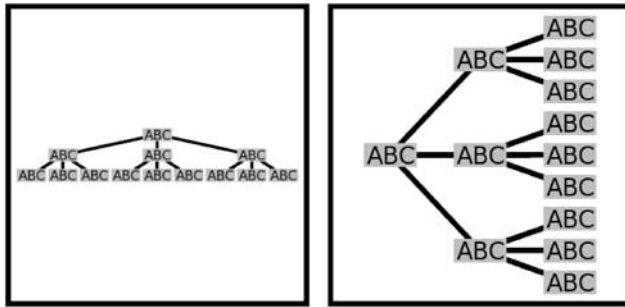
Figure 15: At right, labels have been rotated (as has the entire representation), allowing labels to be shown at a larger size within the same square viewport.

5. *In radial representations, orient labels to be along radial lines*. For reasons analogous to those in the previous guideline, if designers wish to use a radial representation they should consider orienting labels so their length is along radial lines, in imitation of 1(e) rather than 1(d). This will allow labels to be larger within the allowed space. (Note, however, that this can also negatively impact readability, as the user may need to rotate the view to read certain labels.)

6. *In representations that do not use nested enclosure, modify the representation to lay out skinny leaf nodes along a curve that is as long as possible*. As already discussed, laying out skinny leaf nodes along a longer curve will allow the leaf labels to be larger. This motivated our concentric squares representation (Figure 8). Designers may find other ways to tile skinny nodes along longer curves, perhaps by taking inspiration from space-filling curves.

7. *Within the given constraints, choose a representation that maximizes space-efficiency*. Designers typically work within several constraints beyond just space-efficiency, which might make them choose any of the representations considered. However, we can still identify simple criteria and rules of thumb for choosing a representation that increases space-efficiency, which we now discuss.

First, the most basic criterion is that there be labels on all nodes that decrease in size with depth, that is, $a_l(n_i) > a_l(n_{i+1}) > 0$ or $\rho_l > 0$. Second, the representation should have a leaf label size $a_l(n_D)$ that shrinks as slowly as possible (ideally, $a_l(n_D) = \Theta(1/B^D)$ or $\rho_l \approx 1$). Third, the representation should have a total label area A_l as high as possible (ideally, $A_l = 1$). The representations in Table 3 all satisfy the first criterion, and are ranked according to the degree to which they satisfy the second and third.

Other criteria may also be considered, producing a decision procedure for choosing a representation. For example, a representation that uses nested enclosure may be deemed too confusing in certain contexts, or the designer may want to improve legibility of labels on leaf nodes by having them all oriented the same way – such criteria eliminate subsets of representations from further consideration. The following is an example decision procedure based on the information in Table 3:

if okay to use nested enclosure *then*
    use treemap (Figure 9)
else if leaf labels needn't be all oriented the same way *then*
    use concentric squares (Figure 8)
else if L significantly larger than 1 *then*
    use icicle (Figure 1(c))
else if need support for weighted partition *then*
    use icicle (Figure 1(c))
else
    use radial (Figure 1(d))

The first step in the above procedure is to ask if the use of nested enclosure is acceptable. If it is, no further questions need be asked: a treemap should be used because it is the most space-efficient and because it also allows labels to all be oriented the same way, and allows for weighted partitioning of area. If not, then further questions are required to narrow down the choices.

Designers may combine the above criteria with their own criteria that may have little to do with label size, and create their own rules of thumb or decision procedure. Such a decision procedure could also be extended to include novel tree representations not considered here, by performing the same kind of analysis of space-efficiency that we have done in this article.

Using the metrics to evaluate other representations

Given a tree representation not considered in our analysis, such as one being considered by a designer, or a novel representation being proposed by a researcher, which metrics should be used to evaluate its space-efficiency and how?

The section 'Additional metrics, and example analysis of Figure 2' listed six kinds of metrics, and sections 'Additional metrics, and example analysis of Figure 2' and 'General observations' showed how misleading metrics that do not take into account label size can be. We recommend evaluating, most importantly, a metric of *smallest label efficiency* (for example, $a_l(n_D)$ or the average font height in leaf labels) and preferably also a metric of *total label efficiency* (that is, the total area A_l devoted to labels).

These two kinds of metrics can be applied to any given instance of a tree representation, such as those in Figure 2, and so could be used to evaluate and compare different static diagrams of the same tree.

However, if one wishes to evaluate a given representation style (drawing convention), then these metrics should ideally be evaluated analytically, as is done in the section 'Asymptotic analysis of tree representations.' If an analytical approach is too difficult or requires introducing unrealistic assumptions, then a numerical evaluation may be performed over many randomly generated trees to find average values of the metrics. To evaluate

representation styles, we also recommend finding the asymptotic behavior of these metrics for large values of depth D, ideally evaluating $A_{l,\infty}$, as well as the mean area exponent ρ_l and its limit $\rho_{l,\infty}$. As discussed in the section 'Observations concerning ρ and ρ_l,' having $1 \leqslant \rho_{l,\infty} < 2$, with $\rho_{l,\infty}$ close to 1, is a condition associated with highly efficient representations.

Finally, we note that there is no need to consider any explicit metrics of size or aspect ratio of nodes, because these are both implicitly taken into account when evaluating the size of labels.

Rectified Treemaps

Thinking about the third design guideline in the section 'Design guidelines' led us to create a variant of squarified treemaps that increases the size of labels. The idea is that, rather than 'squarifying' nodes to give them an aspect ratio closer to 1, we 'rectify' them (that is, make them into rectangles) to give them an aspect ratio closer to the aspect ratio of their labels. We implemented this idea with a simple change to the layout algorithm described in Section 3.2. of Bruls et al.[5] Let R be a list of rectangles that are tiled together to form a single row of total length w and total area s, and let $r(i)$ be the area of the ith rectangle in R. The row has thickness s/w, hence the ith rectangle has dimensions $wr(i)/s$ and s/w, and has aspect ratio $\lambda(w^2 r(i)/s^2) \geqslant 1$ where we define the function $\lambda(x) = \max(x, 1/x) \geqslant 1$ to always have aspect ratios greater than or equal to 1. As defined in Bruls et al's,[5] the worst (that is, largest) aspect ratio of the rectangles in the row is

$$\text{worst}(R) = \max_i \lambda(w^2 r(i)/s^2)$$

The function *worst* is used within Bruls et al.'s algorithm[5] to generate a squarified layout. Our one modification to the algorithm is to change the function to

$$\text{worst}(R) = \max_i \begin{cases} \lambda(\lambda(w^2 r(i)/s^2)/L(i)) & i\text{th rectangle is a leaf node} \\ \lambda(w^2 r(i)/s^2) & \text{otherwise} \end{cases}$$

where $L(i) \geqslant 1$ is the aspect ratio of the label on the ith rectangle. In other words, the effective aspect ratio of leaf nodes is relative to the aspect ratio $L(i)$ of their labels. Bringing this *relative* aspect ratio closer to 1, as the squarified layout algorithm does, results in nodes that better match their label's aspect ratio.

Figures 16 and 17 show the output before and after this change, for a randomly generated tree with random labels. In both cases, each leaf node was given equal weight for the treemap algorithm. As can be seen, the nodes in Figure 16 are often more squarish and contain much unused space, especially when they contain a long label.

Figure 18 shows the output after a second minor change, whereby leaf nodes are given weight proportional to the number of characters in their label to give the same weight to each character. The goal, in this case, is to make each
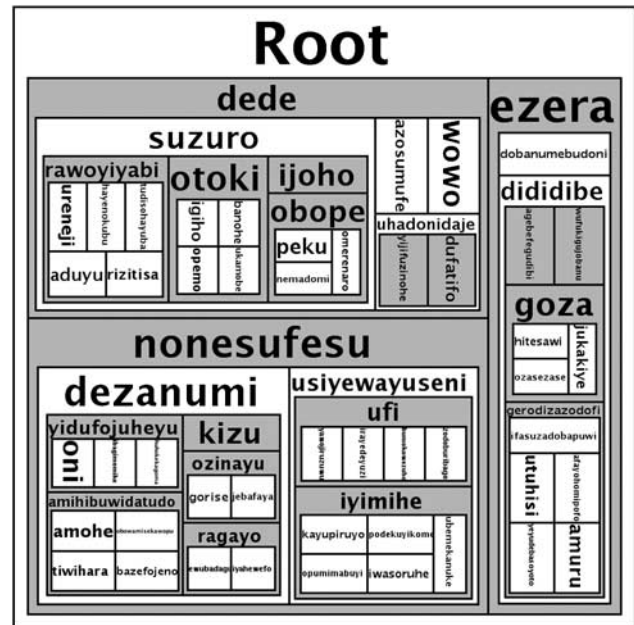

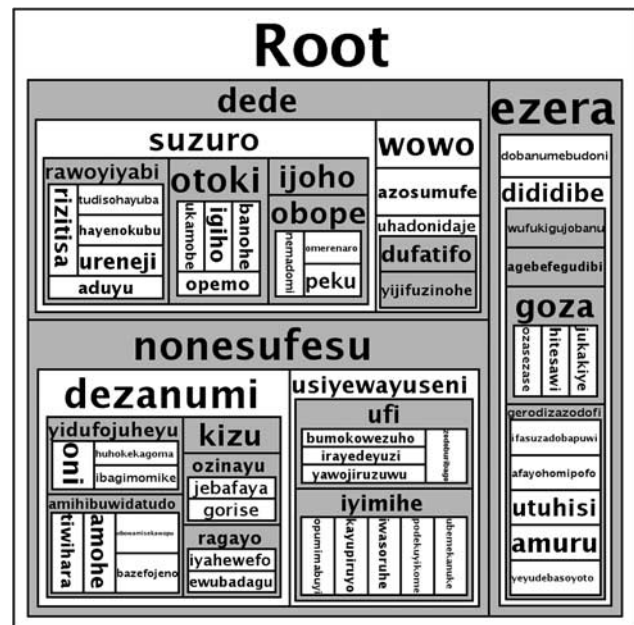**Figure 16:** A squarified treemap of a randomly generated tree.

Figure 17: A rectified treemap of the same tree in Figure 16, in which leaf nodes are made to more closely match the aspect ratio of their labels.

label as legible as possible. However, it would also be possible to give leaf nodes a different weight that is a function of some 'size' attribute of the nodes (such as file size), and in general, the rectified treemap will have larger, more legible labels than those in a normal squarified treemap. (However, such rectified treemaps with a weighted partitioning will also have a disadvantage: the skinnier the
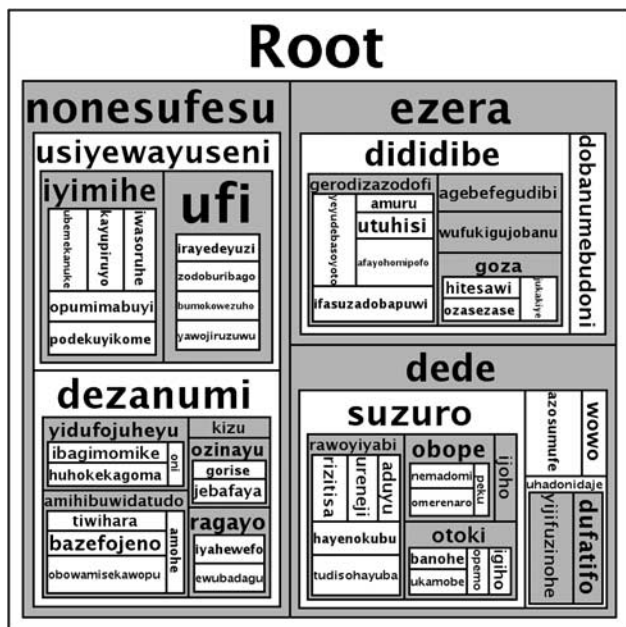
Figure 18: The same as Figure 17, but now with each leaf node given weight proportional to the number of characters in its label to give equal priority to each character.

labels are, the skinnier the nodes will be, and hence the more difficult it will be for a user to accurately compare the areas of nodes. So there is a tradeoff between label size and comparability of node areas.)

The following statistics show that our modifications yield an improvement over Figure 16 in all the label-oriented metrics. Comparing Figures 17 and 18, we see that although Figure 18 has smaller average leaf label height, it also has larger minimum leaf label height, and a smaller standard deviation of leaf label height, that is, the heights are more consistent.

Figure	16	17	18
Total area A of nodes	1.000	1.000	1.000
Mean area exponent ρ	0.811	0.811	0.799
Average area of leaf nodes	0.00688	0.00688	0.00671
Total area A_l of labels	0.331	0.396	0.402
Mean area exponent ρ_l for labels	0.657	0.540	0.496
Average area of leaf labels	0.00244	0.00375	0.00342
Average leaf label height	0.0238	0.0297	0.0265
Minimum leaf label height	0.0105	0.0112	0.0172
Standard deviation of leaf label height	0.0115	0.0115	0.00598

Conclusions

We have presented a rigorous, quantitative analysis and comparison of the space-efficiency of several 2D graphical representations of trees, including most of the basic kinds found in the information visualization literature (or, at least, simplified versions of them). We are unaware of any previous similar analysis performed of *any* single tree representation, let alone all the ones considered here collectively. Our analysis makes clear the relative space-efficiency of each representation with respect to various metrics. This allows us to rank the tree representations, which is important for enabling designers to choose the most space-efficient possibility among those not eliminated by various design constraints. One theoretically interesting result of our work is that even treemaps (as modeled in Figure 9) are suboptimal with respect to certain metrics, having $A_{l,\infty} < 3/4$ for example.

Our analysis is performed in terms of both node size and label size, and investigates multiple facets of space-efficiency – a list of six kinds of metrics is given in the section 'Additional metrics and example analysis of Figure 2.' We have demonstrated that metrics that do not take into account label size can be misleading; for example, concentric circle representations were shown to be more space-efficient (in terms of leaf label size) than icicle diagrams, contrary to what would be expected from simply comparing their total area A. Fortunately, simply evaluating the area of labels can be done without separately evaluating the size or aspect ratio of nodes, as the latter two are both implicitly taken into account in analyzing the size of labels. Hence, we recommend using label-oriented metrics for evaluating representations. We propose that future work in this area go beyond claiming that a new representation is 'space-filling' merely because of the total area A that is occupied, and that researchers apply the recommendations in the section 'Using the metrics to evaluate other representations' to evaluate the performance of new tree representations.

We have also presented a novel metric, the mean area exponent ρ, that quantifies the distribution of area in a representation, that is, that quantifies the tradeoff between shallow and deep nodes. This metric can be calculated for both homogeneous and heterogeneous tree representations. We have also applied variants of the metric (ρ_l and its limit $\rho_{l,\infty}$) in our analysis of label areas, and discussed how certain values are associated with highly space-efficient representations ($\rho_{l,\infty}$ close to, or slightly greater than, 1). These metrics could be used to classify representations (for example, $\rho_{l,\infty}$ could be a dimension within a taxonomy of representations), or could be used as tools to summarize and better understand the performance of a representation, or as input parameters for tuning a given representation (Figure 13). Finally, the definition of ρ is not arbitrary, but affords multiple interpretations (sections 'Mean area exponent ρ' and 'Generalizing ρ to heterogeneous tree representations') that we find fairly compelling.

For designers, we have presented the most immediately applicable practical results of our work in the section 'Design implications and guidelines,' which includes a set

of design guidelines and an example decision procedure for choosing a representation.

Finally, we have shown that our work has generative power, in the novel tree representations developed through this work: concentric squares (Figure 8), asymmetrical concentric circles (Figure 11), a hybrid of indented outline and classical layered (Figure 12) and the rectified treemap (Figures 17 and 18). The concentric squares representation is more space-efficient than many commonly used representations, such as classical layered and concentric circles, and the rectified treemap improves the legibility of labels on *status quo* treemaps.

Future Directions

Future work could perform further analytical or numerical evaluation of the metrics, possibly on a corpus of real-world trees, and/or on other tree representations that we have not considered such as the 'space-optimized' representation of Nguyen and Huang[19] or hybrid mixtures of representations[29] or 3D representations such as cone trees.[30]

Representations with greater space-efficiency might also be useful to develop, such as treemaps with margins that do not have the limitation of $A_{l,\infty} < 3/4$ of representation 9 (probably by placing some nodes to the right of their parent's label), or a variation on Figure 10 that generalizes to $B \neq 8$. It is also possible that the efficiency of concentric squares (Figure 8) could be improved by laying out leaf nodes along a curve whose length is greater than 4, perhaps even along a space-filling curve. (Interestingly, space-filling curves have recently been used for graph layout.[31])

A significant issue for future metrics of tree representations is to take into account human perceptual abilities and/or how well various user tasks, such as those described in the work of Wang *et al*,[32] are supported by each representation. For example, with a classical layout (1(a)) it is easier to estimate the distance between two nodes and easier to see if two nodes are on the same level than in a treemap – this is ignored by our metrics. As more experimental data emerges from researchers about user performance at various tasks with various tree representations, it would probably be reasonable to extend Table 3 with additional rows indicating the relative support for different tasks, and to refine the decision procedure in the 'Design guidelines' section.

Our metrics also assume that there is no value in whitespace, whereas in practice whitespace can be very valuable in making groupings of elements easier to perceive. It may be possible to develop new metrics that weigh the value of label size along with the value of some whitespace to aid in the perception of structure and groupings.

Another issue not yet addressed by the metrics presented is how to model interactive displays that change over time, in which nodes and labels may appear, move, change in size or be elided, depending on the user's input.

References

1 Reingold, E.M., & Tilford, J.S. (1981). Tidier drawings of trees. *IEEE Transactions on Software Engineering* 7(2): 223–228.

2 Buchheim, C., Jünger, M. and Leipert, S. (2002). *Improving Walker's algorithm to run in linear time*. In: S.G. Kobourov and M.T. Goodrich (eds.) Proceedings of Symposium on Graph Drawing (GD), Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, pp. 344–353.

3 Johnson, B. and Shneiderman, B. (1991). *Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures*. In: G.M. Nielson and L. Rosenmblum (eds.) Proceedings of IEEE Visualization (VIS); Piscataway, NJ, USA: IEEE Computer Society Press, pp. 284–291.

4 Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics (TOG)* 11(1): 92–99.

5 Bruls, M., Huizing, K. and van Wijk, J. J (2000) *Squarified Treemaps*. In: W. de Leeuw and R. van Liere (eds.) Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization; Vienna, Berlin, Germany: Springer–Verlag, pp. 33–42.

6 Wattenberg, M. (2005) *A Note on Space-Filling Visualizations and Space-Filling Curves*. In: J. Stasko and M. Ward (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis); Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 181–186.

7 Andrews, K. and Heidegger, H. (1998) *Information Slices: Visualising and Exploring Large Hierarchies Using Cascading, Semi-circular Discs*. In: G. Wills and J. Dill (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis), Late Breaking Hot Topics, Los Alamitos, CA: IEEE Computer Society Press, pp. 9–12.

8 Stasko, J. and Zhang, E. (2000) *Focus + Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations*. In: S.F. Roth and D.A. Keim (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis), Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 57–65.

9 Yang, J., Ward, M.O. and Rundensteiner, E.A. (2002) *InterRing: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures*. In: P.C. Wong and K. Andrews (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis), Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 77–84.

10 Herman, I., Melançon, G., and Marshall, M.S. (January 2000). Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* 6(1): 24–43.

11 Eades, P.D. (1992). Drawing free trees. *Bulletin of the Institute for Combinatorics and its Applications* 5: 10–36.

12 Di Battista, G., Eades, P., Tamassia, R., & Tollis, I.G. (1999). *Graph Drawing: Algorithms for the Visualization of Graphs*. Englewood Cliffs, NJ: Prentice-Hall.

13 Boardman, R. (2000). *Bubble trees: The Visualization of Hierarchical Information Structures*. In: T. Turner and G. Szwillus (eds.) Extended Abstracts of ACM Conference on Human Factors in Computing Systems (CHI); New York, USA: ACM Press, pp. 315–316.

14 Teoh, S.T. and Ma, K.-L. (2002). *RINGS: A Technique for Visualizing Large Hierarchies*. In: S.G. Kobourov and M.T. Goodrich (eds.)

Proceedings of Graph Drawing (GD), Berlin, Germany: Springer-Verlag, pp. 268–275.

15 Bertin, J. (1967). *Sémiologie graphique: Les diagrammes, Les réseaux, Les cartes*. Paris: Éditions Gauthier-Villars, (2nd ed. 1973, English translation 1983).

16 Knuth, D.E. (1968) *The Art of Computer Programming*, Volume I: Fundamental Algorithms. Reading, MA, USA: Addison-Wesley, pp. 309–310.

17 Engelbart, D.C. and English, W.K. (1968) A research center for augmenting human intellect. (Video of public demonstration of NLS). Available from http://www.1968demo.org/.

18 Plaisant, C., Grosjean, J. and Bederson, B.B. (2002) *SpaceTree*: *Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation*. In: P.C. Wong and K. Andrews (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis); Los Alamitos, California, USA: IEEE Computer Society Press, pp. 57–64.

19 Nguyen, Q.V. and Huang, M.L. (2002) *A Space-Optimized Tree Visualization*. In: P.C. Wong and K. Andrews (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis); Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 85–92.

20 Nguyen, Q.V., & Huang, M.L. (2005). EncCon: An approach to constructing interactive visualization of large hierarchical data. *Information Visualization* 4: 1–21.

21 Anand, V., Hansen, K., Jianu, R. and Rusu, A. (2004) *Area-Efficient Visualization of Web Data*. In: O.D.H.R. Arabnia and S. Chatterjee (eds.) Proceedings of International Conference on Internet Computing (IC), Vol. 1, Las Vegas, Nevada, USA: CSREA Press, pp. 83–89.

22 Garg, A., Goodrich, M.T., & Tamassia, R. (1996). Planar upward tree drawings with optimal area. *International Journal of Computational Geometry and Applications* 6: 333–356.

23 Shin, C.-S., Kim, S.K. and Chwa, K.-Y. (1996) *Area-Efficient Algorithms for Upward Straight-line Tree Drawings*. In: J.-Y. Cai and C.K. Wong (eds.) Proceedings of International Computing and Combinatorics (COCOON), Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, pp. 106–116.

24 Pulo, K. and Takatsuka, M. (2003) *Inclusion Tree Layout Convention*: *An Empirical Investigation*. In: T. Pattison and B. Thomas (eds.) Proceedings of Asia-Pacific Symposium on Information Visualisation, Adelaide, Australia: Australian Computer Society, pp. 27–35.

25 Kakoulis, K.G. and Tollis, I.G. (1998) *A unified Approach to Labeling Graphical Features*. In: R. Janardan (ed.) Proceedings of Symposium on Computational Geometry, New York, USA: ACM Press, pp. 347–356.

26 Koike, H. and Yoshihara, H. (eds.) (1993) *Fractal Approaches for Visualizing Huge Hierarchies*. Proceedings of IEEE Symposium on Visual Languages (VL), Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 55–60.

27 Card, S.K. and Nation, D. (2002) *Degree-of-Interest Trees*: *A Component of an Attention-Reactive User Interface*. In: S.L.M. De Marsico and E. Panizzi (eds.) Proceedings of ACM Advanced Visual Interfaces (AVI); New York, USA: ACM Press.

28 Bederson, B.B. and Hollan, J.D. (1994) *Pad++*: A Zooming Graphical Interface for Exploring Alternate Interface Physics. In: P. Szekely (ed.) Proceedings of ACM Symposium on User Interface Software and Technology (UIST); New York, USA: ACM Press, pp. 17–26.

29 Zhao, S., McGuffin, M.J. and Chignell, M.H. (2005) *Elastic Hierarchies*: *Combining Treemaps and Node-Link Diagrams*. In: J. Stasko and M. Ward (eds). Proceedings of IEEE Symposium on Information Visualization (InfoVis), Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 57–64.

30 Robertson, G.G., Mackinlay, J.D. and Card, S.K. (1991) *Cone Trees: Animated 3D Visualizations of Hierarchical Information*. In: S.P. Robertson, G.M. Olson and J.S. Olson (eds.) Proceedings of ACM 1991 Conference on Human Factors in Computing Systems (CHI); New York, USA: ACM Press, pp. 189–194.

31 Muelder, C. and Ma, K.L. (2008) *Rapid Graph Layout using Space Filling Curves*. In: C. North and S. Carpendale (eds.) Proceedings of IEEE Symposium on Information Visualization (InfoVis); Los Alamitos, CA, USA: IEEE Computer Society Press.

32 Wang, Y., Teoh, S.T. and Ma, K.-L. (2006) *Evaluating the Effectiveness of Tree Visualization Systems for Knowledge Discovery*. In: J.A. Jorge (ed.) Proceedings of Eurographics/IEEE-VGTC Symposium on Visualization; Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 67–74.