Convex Treemaps with Bounded Aspect Ratio

Mark de Berg* Bettina Speckmann* Vincent van der Weele†

Abstract

Treemaps are a popular technique to visualize hierarchical data. The input is a weighted tree $\mathcal{T}$ where the weight of each node is the sum of the weights of its children. A treemap for $\mathcal{T}$ is a hierarchical partition of a rectangle into simply connected regions, usually rectangles. Each region represents a node of $\mathcal{T}$ and the area of each region is proportional to the weight of the corresponding node. An important quality criterium for treemaps is the aspect ratio of its regions. Unfortunately, one cannot bound the aspect ratio if the regions are restricted to be rectangles. Hence Onak and Sidiropoulos introduced *polygonal partitions*, which use convex polygons. We are the first to obtain convex partitions with optimal aspect ratio $O(\text{depth}(\mathcal{T}))$. We also consider the important special case that $\text{depth}(\mathcal{T}) = 1$, that is, single-level treemaps. We show how to construct convex single-level treemaps that use only four simple shapes for the regions and have aspect ratio at most 34/7.

1 Introduction

Treemaps are a popular technique to visualize hierarchical data [10]. The input is a weighted tree $\mathcal{T}$ where the weight of each node is the sum of the weights of its children. A treemap for $\mathcal{T}$ is a hierarchical partition of a rectangle into simply connected regions, usually rectangles. Each region represents a node of $\mathcal{T}$ and the area of each region is proportional to the weight of the corresponding node. To visualize the hierarchical structure the region associated with a node must contain the regions associated with its children. Shneiderman [11] first presented an algorithm for the automatic creation of rectangular treemaps. Treemaps are since used to visualize hierarchical data from a variety of application areas, for example, stock market portfolios [7], tennis competitions trees [6], large photo collections [3], and business data [13].

One of the most important quality criteria for treemaps is the aspect ratio of its regions [8]. Hence several approaches [3, 4] try to "squarify" the regions of a rectangular treemap. However, one cannot bound the aspect ratio if the regions are restricted to be rectangles. Hence, several types of treemaps using region shapes other than rectangles have been proposed. Balzer and Deussen [1, 2] use centroidal Voronoi tessellations. Their algorithm is iterative and can give no guarantees on the aspect ratio of the regions produced. Wattenberg [14] developed treemaps whose regions follow a space filling curve on a grid, so called Jigsaw maps. Jigsaw maps assume integer weights, which must add up to a square number. The regions of the maps are rectilinear, but highly non-(ortho)convex. However, they do have aspect ratio 4.

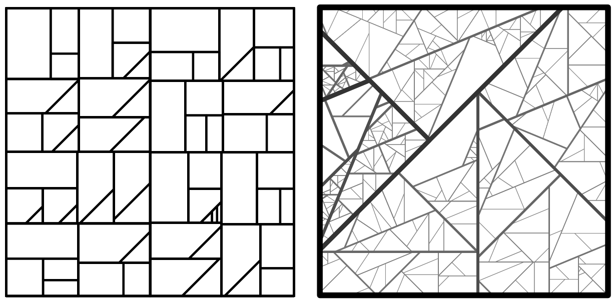


Figure 1: Treemaps constructed by our drawing algorithms: single-level convex, hierarchical convex.

Onak and Sidiropoulos [9] introduced *polygonal partitions*, which use convex polygons. They proved an aspect ratio of $O((\text{depth}(\mathcal{T}) \cdot \log n)^{17})$ for a tree $\mathcal{T}$ with n leaves. In cooperation with De Berg, this bound has since been improved to $O(\text{depth}(\mathcal{T}) + \log n)$ [5]. The latter paper also gives a lower bound construction of $\Omega(\text{depth}(\mathcal{T}))$.

Results and organization. We are the first to obtain convex partitions with optimal aspect ratio $O(\text{depth}(\mathcal{T}))$. Our recursive drawing procedure is described in Section 2. In Section 3 we also consider the important special case that $\text{depth}(\mathcal{T}) = 1$, that is, single-level treemaps. We show how to construct convex single-level treemaps that use only four shapes and have aspect ratio at most 34/7. Figure 1 shows two treemaps constructed by our drawing algorithms. In the treemap on the right the hierarchical structure is emphasized by line thickness and color: thicker, darker lines delimit nodes higher in the hierarchy. Specifically, there are four nodes on the top-level.

Preliminaries. Our input is a rooted tree $\mathcal{T}$. Following [5] we say that $\mathcal{T}$ is *properly weighted* if each node ν of $\mathcal{T}$ has a positive weight weight(ν) that equals the sum of the weights of the children of ν. We as-

*Department of Mathematics and Computer Science, TU Eindhoven, mdberg@win.tue.nl and speckman@win.tue.nl. B. Speckmann was supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 639.022.707.

†Max-Planck-Institut für Informatik, Saarbrücken, vdweele@mpi-inf.mpg.de.

sume that weight(root($\mathcal{T}$)) $= 1$. A treemap for $\mathcal{T}$ associates a region $R(\nu)$ with each node $\nu \in \mathcal{T}$ such that (*i*) $R(\text{root}(\mathcal{T}))$ is the unit square, (*ii*) for every node we have $\text{area}(R(\nu)) = \text{weight}(\nu)$, and (*iii*) for any node ν, the regions associated with the children of ν form a partition of $R(\nu)$. We denote the set of children of a node ν by children(ν).

The aspect ratio of a treemap is the maximum aspect ratio of any of its regions. For a single region, we use the following definition from [5]: the aspect ratio of a convex region R is $\text{diam}(R)^2/\,\text{area}(R)$.

Lemma 1 *Suppose all children of node ν have weight at most $2/3 \cdot \text{weight}(\nu)$. Then we can partition* children(ν) *into two subsets S_1 and S_2, such that* $\text{weight}(S_2) \leqslant \text{weight}(S_1) \leqslant 2/3 \cdot \text{weight}(\nu)$.

2 Hierarchical treemaps

We describe a recursive algorithm for computing a polygonal partition (convex treemap) of aspect ratio $O(\text{depth}(\mathcal{T}))$ for a properly weighted tree $\mathcal{T}$. Our algorithm has two phases. We first convert $\mathcal{T}$ into a binary tree $\mathcal{T}^*$ and then construct a partition for $\mathcal{T}^*$. This approach is similar to the one taken by De Berg *et al.* [5], but we implement both phases differently.

Converting to a binary tree. We recursively convert $\mathcal{T}$ into a binary tree $\mathcal{T}^*$, replacing each node with $k > 2$ children in $\mathcal{T}$ by a binary subtree with $k-1$ nodes. During this process we assign a label $d(\nu)$ to each node ν, corresponding to the depth of ν in $\mathcal{T}$.

At every step, we treat a node ν with label $d(\nu)$ and convert the subtree rooted at ν. (Initially $\nu = \text{root}(\mathcal{T})$ with $d(\text{root}(\mathcal{T})) = 0$.) If ν is a leaf there is nothing to do. If ν has two children we recurse on these children and assign them label $d(\nu)+1$. Otherwise ν has k children, $\text{children}(\nu) = \{\nu_1, \ldots, \nu_k\}$, for some $k > 2$. We then distinguish two cases.

If there is a "heavy" child, say ν_1, with $\text{weight}(\nu_1) \geqslant \text{weight}(\nu)/2$, then we turn ν into a binary node whose children are ν_1 and a new node μ_1; the children of μ_1 are $\nu_2, \ldots, \nu_k$. We recurse on ν_1 and μ_1, with $d(\nu_1) = d(\nu)+1$ and $d(\mu_1) = d(\nu)$. Otherwise all children have weight less than $\text{weight}(\nu)/2$, and hence there is a partition of children(ν) into two subsets S_1 and S_2 such that $\text{weight}(S_i) \leqslant 2/3 \cdot \text{weight}(\nu)$ for $i \in \{1,2\}$. We turn ν into a binary node with children μ_1 and μ_2, with children from S_1 and S_2, respectively, and we recurse on μ_1 and μ_2 with $d(\mu_1) = d(\mu_2) = d(\nu)$.

Drawing a binary tree. Generalizing ϕ-separated polygons [5], we define a (k,ϕ)-*polygon* to be a convex polygon P such that

(i) P does not have parallel edges, except possibly two horizontal edges and two vertical edges. Moreover, each non-axis-parallel edge e makes an angle of at least ϕ with any other edge and also with the x-axis and the y-axis.
(ii) If P has two horizontal edges, then $\text{diam}(P)/\,\text{height}(P) \leqslant k$.
(iii) If P has two vertical edges, then $\text{diam}(P)/\,\text{width}(P) \leqslant k$.

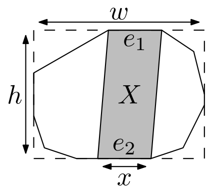
It follows from the definition of ϕ-separated polygons that a (k,ϕ)-polygon P is a ϕ-separated polygon, if it respects the following:

- if P has two horizontal edges, then $\text{height}(P) \geqslant \text{width}(P)$;
- if P has two vertical edges, then $\text{width}(P) \geqslant \text{height}(P)$.

Note that a (k,ϕ)-polygon P is ϕ-separated if its bounding box is square.

Lemma 2 *Any (k,ϕ)-polygon has aspect ratio* $O(\max(k, 1/\phi))$.

Proof. Let P be a (k,ϕ)-polygon with $w = \text{width}(P)$ and $h = \text{height}(P)$. Assume that $w \geqslant h$. Let e_1 and e_2 be the horizontal edges (possibly of length 0), let $x = \min(|e_1|, |e_2|)$, and let X be a parallellogram of width x. We distinguish two cases.



Case 1: $x > w/2$. P has two horizontal edges, so $h \geqslant \text{diam}(P)/k$. Clearly, the area of P is at least the area of X which is $xh > w \cdot \text{diam}(P)/(2k)$. The diameter of P is at most the diameter of the enclosing rectangle, hence $\text{diam}(P) \leqslant \sqrt{w^2+h^2} \leqslant w\sqrt{2}$. Combined:

$$\text{asp}(P) = \frac{\text{diam}(P)^2}{\text{area}(P)} \leqslant \frac{2k \cdot \text{diam}(P)}{w} \leqslant 2\sqrt{2}k = O(k).$$

Case 2: $x \leqslant w/2$. We obtain polygon P' from P by reducing the length of e_1 and e_2 with $\min(x, w-h)$. Clearly, $\text{area}(P') \leqslant \text{area}(P)$. Observe that P' is a ϕ-separated polygon since either it has at most 1 horizontal edge (and $w - x \geqslant h$), or the bounding box of P' is square. Therefore, $\text{asp}(P') = O(1/\phi)$ [5]. Using $\text{diam}(P) \leqslant \sqrt{2}w$ and

$$\text{diam}(P') \geqslant w - \min(x, w-h) \geqslant w - x \geqslant w/2\ ,$$

we calculate

$$\begin{aligned}\text{asp}(P) = \frac{\text{diam}(P)^2}{\text{area}(P)} &\leqslant \frac{2w^2}{\text{area}(P)} \\ &\leqslant 8 \cdot \frac{\text{diam}(P')^2}{\text{area}(P')} = 8 \cdot \text{asp}(P') = O(1/\phi).\end{aligned}$$

□

We construct the partition for $\mathcal{T}^*$ in a top-down manner. Each node ν in $\mathcal{T}^*$ has an associated region $R(\nu)$; initially $\nu = \text{root}(\mathcal{T}^*)$ and $R(\nu)$ is the unit square. We write $n(\nu)$ for the number of non-axis-parallel edges in $R(\nu)$. We maintain the following invariants:

(Inv-1) $n(\nu) \leqslant d(\nu) + 4$;

(Inv-2) $R(\nu)$ is a $(k, \phi(\nu))$-separated polygon for $k = \sqrt{17}$ and $\phi(\nu) = \pi/(2(d(\nu)+6))$.

The invariants are satisfied for $\nu = \text{root}(\mathcal{T}^*)$. Now consider a node ν that is not $\text{root}(\mathcal{T}^*)$. If ν is a leaf, there is nothing to do. Otherwise, let ν_1 and ν_2 be the two children of ν. Assume that $\text{weight}(\nu_1) \geqslant \text{weight}(\nu_2)$. We distinguish two cases.

Case 1: $d(\nu_1) = d(\nu) + 1$**.** Since $R(\nu)$ uses at most $d(\nu) + 4$ non-axis-parallel edges, there is a line ℓ that makes an angle of at least $\pi/(2(d(\nu) + 6))$ with each of the edges of $R(\nu)$ and with the x- and the y-axis. Imagine placing ℓ such that it splits $R(\nu)$ into two halves of equal area, and let R' be the half with the smallest number of non-axis-parallel edges. Now partition $R(\nu)$ into subpolygons $R(\nu_1)$ and $R(\nu_2)$ of the appropriate area with a cut c that is parallel to ℓ such that $R(\nu_2) \subset R'$. (Thus c lies inside R'.) We claim that both $R(\nu_1)$ and $R(\nu_2)$ satisfy the invariants.

Clearly $R(\nu_1)$ uses at most one edge more than $R(\nu)$. Since $d(\nu_1) = d(\nu)+1$, this implies that (Inv-1) is satisfied for $R(\nu_1)$. Now consider the number of non-axis-parallel edges of $R(\nu_2)$. This is no more than the number of non-axis-parallel edges of R'. At most two non-axis-parallel edges are on both sides of ℓ, hence this number is bounded by

$$\begin{aligned} n(\nu_2) &\leqslant \left\lfloor \frac{n(\nu)+2}{2} \right\rfloor + 1 \leqslant \left\lfloor \frac{d(\nu)+6}{2} \right\rfloor + 1 \\ &= \left\lfloor \frac{d(\nu)}{2} \right\rfloor + 4 \leqslant d(\nu) + 4 \leqslant d(\nu_2) + 4. \end{aligned}$$

Given the choice of ℓ, and because $d(\nu_i) \geqslant d(\nu)$ and $R(\nu)$ satisfies (Inv-2), we know that the minimum angle between any two non-parallel edges of $R(\nu_i)$ ($i \in \{1, 2\}$) is at least $\pi/(2(d(\nu_i) + 6))$. The following lemma, that we prove in the full version, suffices to show that $R(\nu_1)$ and $R(\nu_2)$ satisfy (Inv-2).

Lemma 3 *If $R(\nu_i)$ has two horizontal edges, then* $\text{diam}(R(\nu_i))/\text{height}(R(\nu_i)) \leqslant k$ *and if $R(\nu_i)$ has two vertical edges, then* $\text{diam}(R(\nu_i))/\text{width}(R(\nu_i)) \leqslant k$, *for $i \in \{1, 2\}$.*

Case 2: $d(\nu_1) = d(\nu)$**.** By construction of $\mathcal{T}^*$, $1/3 \cdot \text{weight}(\nu) \leqslant \text{weight}(\nu_1) \leqslant 2/3 \cdot \text{weight}(\nu)$. We now partition $R(\nu)$ into two subpolygons of the appropriate area with an axis-parallel cut orthogonal to the longest side of the axis-parallel bounding box of $R(\nu)$. The possible positions of this cut are limited by convexity, as specified in the following lemma.

Lemma 4 *Let P be a convex polygon with* $\text{width}(P) \geqslant \text{height}(P)$. *We can partition P with a vertical cut into two subpolygons P_1, P_2, where* $\text{area}(P)/3 \leqslant \text{area}(P_i) \leqslant 2/3 \cdot \text{area}(P)$ *(for $i \in \{1, 2\}$), such that* $\text{width}(P)/4 \leqslant \text{width}(P_i) \leqslant 3/4 \cdot \text{width}(P)$.

The number of non-axis-parallel edges of $R(\nu_1)$ and $R(\nu_2)$ is no more than the number of non-axis-parallel edges of $R(\nu)$. Since $d(\nu_i) \geqslant d(\nu)$, this implies $R(\nu_1)$ and $R(\nu_2)$ satisfy (Inv-1). As for (Inv-2), note that the cut does not introduce any new non-axis-parallel edges. It is thus met by the following lemma.

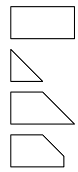
Lemma 5 *If $R(\nu_i)$ has two horizontal edges (for $i \in \{1, 2\}$),* $\text{diam}(R(\nu_i))/\text{height}(R(\nu_i)) \leqslant \sqrt{17}$. *Similarly, if $R(\nu_i)$ has two vertical edges,* $\text{diam}(R(\nu_i))/\text{width}(R(\nu_i)) \leqslant \sqrt{17}$.

Lemma 2, together with the fact that $\max_{\nu \in \mathcal{T}^*} d(\nu) = \text{depth}(\mathcal{T})$ and Inv-2, implies the result.

Theorem 6 *Every properly weighted tree of depth d can be represented by a polygonal partition (convex treemap) which has aspect ratio $O(d)$.*

3 Single-level treemaps

Rectangle:
Triangle:
Chisel:
Pentagon:



We now consider the special case that $\text{depth}(\mathcal{T}) = 1$. Our input is hence a set of positive weights. We describe a recursive drawing procedure that creates a treemap of aspect ratio at most 34/7 and uses only the four shapes depicted on the right. We do not recurse on pentagons, these are used only for single high weights.

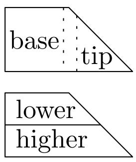
We denote the bounding rectangle of a region R by $\rho(R)$. The aspect ratio of a rectangle ρ is defined as $\text{long}(\rho)/\text{short}(\rho)$, where $\text{long}(\rho)$ is the maximum of $\text{width}(\rho)$ and $\text{height}(\rho)$ and $\text{short}(\rho)$ is the minimum. This is equivalent to $\text{long}(\rho)^2/\text{area}(\rho)$. We write $\text{short}(R)$ for $\text{short}(\rho(R))$ and $\text{long}(R)$ for $\text{long}(\rho(R))$. We frequently use the aspect ratio of the bounding rectangle of a region and write $\text{asp}_\rho(R)$ for $\text{long}(R)/\text{short}(R)$. Our drawing procedure keeps the following invariant:

(Inv) $\text{asp}_\rho(R) \leqslant 4$ for all regions R.

We convert $\mathcal{T}$ into a binary tree $\mathcal{T}^*$ as before and create a drawing for $\mathcal{T}^*$ in a top-down manner. Each node ν has again an associated region $R(\nu)$; initially $\nu = \text{root}(\mathcal{T}^*)$ and $R(\nu)$ is the unit square. If ν is a leaf, we are done. Otherwise, let ν_1 and ν_2 be the children of ν and let ν_1 be the heavier child. We distinguish three cases according to the shape of $R(\nu)$.

Case 1: Rectangle. If $\text{weight}(\nu_1)$ and $\text{weight}(\nu_2)$ are roughly equal, $\text{weight}(\nu)/(\text{asp}_\rho(R(\nu)) \cdot \text{weight}(\nu_2)) \leqslant 4$, we cut $R(\nu)$ through its longer side into two rectangles and have $\text{short}(R(\nu))^2/\text{area}(R(\nu_2)) \leqslant 4$ and hence $\text{asp}_\rho(R(\nu_i)) \leqslant 4$. Otherwise we draw $R(\nu_2)$ as a equilateral right-angled triangle in a corner of $R(\nu)$. Since we use only equilateral right-angled triangles the bounding rectangle of any triangle is a square. Furthermore, since $\rho(R(\nu_1))$ equals $\rho(R(\nu)$ we also have $\mathrm{asp}_\rho(R(\nu_1)) \leqslant 4$.

Case 2: Triangle. We cut a triangle into a chisel and a triangle. The bounding rectangle of the chisel has aspect ratio at most 4 if its width is at least 1/4 times the width of the triangle. This is the case if $\mathrm{weight}(\nu_1)/\mathrm{weight}(\nu) \geqslant 7/16$. Since ν_1 is the heavier child, this always holds.



Case 3: Chisel. Cutting the longer side of a chisel, by an *orthogonal cut*, yields two regions *base* and *tip*. Cutting the shorter side of a chisel, by a *parallel cut*, yields regions *higher* and *lower*. We first give two lemmas for analyzing the two types of cuts. For brevity, we write $\mathrm{rel}(\nu_i) = \mathrm{weight}(\nu_i)/\mathrm{weight}(\nu)$, $i \in \{1, 2\}$.

Lemma 7 *For an orthogonal cut, the regions meet the invariant if* $\mathrm{short}(R(\nu))^2/\,\mathrm{area}(\mathit{base}) \leqslant 4$.

Lemma 8 *For a parallel cut, with higher* $= R(\nu_1)$, *we have* $\mathrm{asp}_\rho(\mathit{lower}) \leqslant \mathrm{asp}_\rho(\mathit{higher})$ *iff* $\mathrm{rel}(\nu_1) \leqslant \mathrm{asp}_\rho(R(\nu))/(2 \cdot \mathrm{asp}_\rho(R(\nu)) - 1)$.

We distinguish three cases.

Case 1: ν_1 is a leaf. We let $R(\nu_1)$ be the base and $R(\nu_2)$ be the tip of an orthogonal cut. Since $\mathrm{rel}(\nu_1) \geqslant 1/2$, we can easily show that $\mathrm{short}(R(\nu))^2/\,\mathrm{area(base)} \leqslant 4$. By Lemma 7, the subregions meet the invariant.

Case 2: ν_1 is not a leaf and $\mathrm{asp}_\rho(R(\nu)) \geqslant 3/2$. We let $R(\nu_1)$ be the tip and $R(\nu_2)$ be the base of an orthogonal cut. The base is certainly a rectangle, since the rectangular part of the chisel is at least half of its total area and ν_2 is the lighter child. Using $\mathrm{rel}(\nu_2) \geqslant 1/3$ (Lemma 1), we can show that $\mathrm{short}(R(\nu))^2/\,\mathrm{area(base)} \leqslant 4$. Then, the subregions meet the invariant by Lemma 7.

Case 3: ν_1 is not a leaf and $\mathrm{asp}_\rho(R(\nu)) < 3/2$. We let $R(\nu_1)$ be the higher and $R(\nu_2)$ be the lower of a parallel cut. It is not hard to see that $\mathrm{asp}_\rho(\mathrm{higher})$ meets the invariant if $\mathrm{short(higher)} \geqslant 3/8 \cdot \mathrm{short}(R(\nu))$, which follows from ν_1 being the heavier child. Moreover, since $\mathrm{rel}(\nu_1) \leqslant 2/3$ by Lemma 1, it follows from Lemma 8 that lower meets the invariant as well.

The aspect ratio of a region R, compared to $\mathrm{asp}_\rho(R)$, is maximal if R is a chisel. Together with the invariant, this implies the following result.

Theorem 9 *Every properly weighted single-level tree can be represented by a convex treemap which uses only four simple shapes and has aspect ratio at most 34/7.*

References

[1] M. Balzer and O. Deussen. Voronoi treemaps. In *Proc. IEEE Symposium on Information Visualization*, pages 7–14, 2005.

[2] M. Balzer, O. Deussen, and C. Lewerentz. Voronoi treemaps for the visualization of software metrics. In *Proc. ACM Symposium on Software Visualization*, pages 165–172, 2005.

[3] B. B. Bederson, B. Shneiderman, and M. Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, 21(4):833–854, 2002.

[4] M. Bruls, K. Huizing, and J. van Wijk. Squarified treemaps. In *Proc. Joint Eurographics and IEEE TCVG Symposium on Visualization*, pages 33–42. Springer, 2000.

[5] M. de Berg, K. Onak, and A. Sidiropoulos. Fat polygonal partitions with applications to visualization and embeddings. In preparation. `http://arxiv.org/abs/1009.1866v1`, 2010.

[6] L. Jin and D. C. Banks. Tennisviewer: A browser for competition trees. *IEEE Computer Graphics and Applications*, 17(4):63–65, 1997.

[7] W. Jungmeister and D. Turo. Adapting treemaps to stock portfolio visualization. Technical report UMCP-CSD CS-TR-2996, University of Maryland, 1992.

[8] N. Kong, J. Heer, and M. Agrawala. Perceptual guidelines for creating rectangular treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):990–998, 2010.

[9] K. Onak and A. Sidiropoulos. Circular partitions with applications to visualization and embeddings. In *Proc. 24th Symposium on Computational Geometry*, pages 28–37, 2008.

[10] B. Shneiderman. Treemaps for space-constrained visualization of hierarchies. `http://www.cs.umd.edu/hcil/treemap-history/index.shtml`.

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

[12] E. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, 2001.

[13] R. Vliegen, J. J. van Wijk, and E.-J. van der Linden. Visualizing business data with generalized treemaps. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):789–796, 2006.

[14] M. Wattenberg. A note on space-filling visualizations and space-filling curves. In *Proc. IEEE Symposium on Information Visualization*, pages 181–185, 2005.