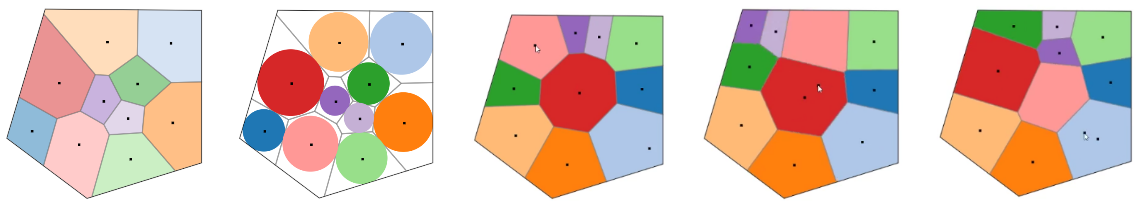
A Force-Directed Power Diagram Approach for Interactive Voronoi Treemaps

A. Abuthawabeh and M. Aupetit

Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar



(a) (b) (c) (d) (e)

Figure 1: *Voronoi treemaps are made of nested Power Diagrams (polygons (a) and grey lines (b)). Their dual circles (b) are natural handles to apply a force-directed approach for moving cells. In this work we propose to apply forces to the dual circles, and we explore the effect on area accuracy and stability. In a failed attempt (c-d-e), we modified the state-of-the-art "static" Weighted Voronoi technique to update weights and positions of all cells, except for the dragged cell (pink polygon) constrained to follow the mouse pointer (white arrow). Unfortunately, although the dragged cell tends to follow the pointer (c), too often it stays fix, seemingly slipping below the mouse pointer (d), until it suddenly jumps to an unexpected position (e) messing up neighboring cells, hence the user's mental map. We propose a force-directed approach to get over these issues.*

Abstract

Voronoi treemaps represent weighted hierarchical data as nested Voronoi diagram partitions with cells' area proportional to the weights. Current techniques to compute them propose static visualizations which can be used for reporting, or dynamic one to capture data update. However, no ideal solution exists yet to interactively rearrange the treemap layout, for instance for a data journalist to tell a story, or for a scientist to create data categorization. We propose a new way to get an interactive Voronoi treemap, where a child cell can be moved by drag-and-drop within a parent cell attempting to preserve both stability (position) and weight (area) during the move. We use a force-directed approach applied to the dual circles of the Power cells to guide the computation of the Power diagram under the hood. Our preliminary quantitative experiments show the force-directed approach provides areas with 10% *weighted average error, which is an order of magnitude higher than standard static approaches, but qualitative observations show that it gives a more predictable and smoother interaction, and a direct control over the stability of the remaining cells. Assuming the user would focus less on getting high accuracy of the areas than keeping a good and stable overview of the treemap while dragging a cell, the force-directed approach appears to be a valuable option to explore further. We also discovered a trade-off between stability and accuracy and the force-directed approach lets the user control it directly.*

CCS Concepts

• ***Human-centered computing*** → ***User interface design;*** • ***Theory of computation*** → ***Computational geometry;***

1. Introduction

Treemaps [JS91] are space-filling graphical representations of hierarchical data, which use sets of nested polygons. In Voronoi Treemaps [BD05], each polygon represents a node of the tree and is a weighted Voronoi cell of the diagram constrained by the polygon of the parent cell. The polygon's area is proportional to the node's weight – typically the number of leaves of the underlying branch of the tree. Usually, parent cells are overlaid with a graphical summary of the underlying data like a piece of text or snippet image, and

DOI: 10.2312/evs.20201057

can be color-coded to represent supplementary data. Different variants of weighted Voronoi partitions exist depending on the metric used [BD05]. We consider the Additively Weighted Power Voronoi diagram (Power cells in the sequel) [Aur87] which produces cells with straight-line boundaries [BD05, NB12]. Several criteria have been proposed to characterize a good treemap [JS91, BD05], like being space-filling while visually preserving and making distinguishable the tree structure, and having compact cells with areas matching accurately the desired weight. In many applications of treemaps, we claim it is also desirable to let the user re-arrange the layout interactively by moving cells, for instance in:

- **Exploratory Data Analysis**: a data scientist could spatially group siblings together based on visual patterns of snippet images representing the children, or semantic of their labels, thus creating categories and sub-categories of similar data and annotating them for further processing or analysis [BZSA18];
- **Information Architecture**: using a card sorting approach [SB04], collaborators could interact with a Voronoi treemap to re-order categories of items, coming to a consensus before taking decisions;
- **Graphical Design**: a graphical designer could interactively re-arrange a treemap to improve its aesthetic [YLG*14];
- **Data Journalism**: a journalist could spatially re-order siblings in a treemap to tell a story more coherent with other information manually instead of relying on full automation [WD08].

However, existing techniques which compute a static layout, cannot easily be adjusted afterward based on visual interaction, as it would result in dramatic changes in the layout. Indeed, when considering dynamically updated treemaps [SFL10], and following *Apprehension* principles stated in [TMB02] and implemented for designing animated transitions of statistical data graphics [HR07], we want an *interactive* treemap layout to be:

- **Stable**: when data or tree structure change, the positions and areas of the cells should change *minimally*;
- **Predictable**: changes should happen in a *predictable* way.

In this work, we explore a new approach to compute an interactive Voronoi Treemap able to keep stable and predictable during a *move-cell* user interaction, *i.e.* dragging a Power cell within its parent cell, with a drag-move-drop mouse interaction.

Our solution relies on the duality of the Power cells to a set of circles in the plane [Aur87]. In a nutshell, the rendered interactive treemap layout is that of the nested Power cells, but we control the position and area of the cells with a force-directed approach applied to their underlying dual circles.

In the following, we describe our solution focusing on an isolated polygonal parent cell and its weighted children. The extension to a multi-level hierarchy of nested cells is straightforward based on this elementary component. Still we discuss related foreseeable issues and possible extensions in the last section.

2. Related Work

Techniques for computing Power Voronoi treemaps, exploit the dual circles of the Power cells, by searching their optimal radius and center, so that the areas of the children's Power cells match the target proportion of the area of their parent's cell [BD05]. However, the optimal solutions may in some cases lead to dual circles not ending up intersecting their Power cell [NB12]. This is one of the obstacle to make an interactive Power Voronoi treemap because the relative positions of the circle and it dual cell are somehow correlated but in a manner hardly predictable by the user. Thus, if the user drags the cell, it is not clear how to move the underlying circle to follow that position still preserving areas. And if the user drag the circle, the cell can slide in unpredictable ways, leading to a low *degree of compatibility* [BL00] especially for a *drag* interaction.

Re-centering dual circles within their Power cell at each iteration as proposed by Nocaj and Brandes [NB12] does not solve the previous issue although it makes a move toward a better circle/cell graphical coupling. Another issue is related to the fact there is no constraint to maintain stable positions during the optimization process.It results in possibly dramatic changes of the layout converging to another local optimum if parameters (positions and weights) are slightly modified by the user.

A technique to build a dynamic additively-weighted Voronoi treemap using GPU has been studied [SFL10] to account for dynamically updated data, but not for interactively re-arranging the positions of the cells and not for the Power cells type of layout we focus on. Bubble and circular treemaps have been proposed [HL15,GSWD18] where nodes hierarchy are represented by nested circles or clusters of bubbles. They show very clearly the hierarchical data structure, but at the expense of empty space between the circles. The cells are located with a circle-packing technique based on the duality of circles and Power diagrams [HL15], or with a force-directed approach [GSWD18].

Following the semantic interaction approach [End14], we propose a way to control the Power cell during a *move-cell* interaction to get better *predictability* and *stability* of the Voronoi treemap, with the aim to better support the user steering the layout in the intended way.

3. Moving Power cells using forces on dual circles

3.1. Failed attempt

We tried initially to modify the technique by Nocaj and Brandes [NB12] which has been implemented in `D3.js` [Wei], by forcing the center of the circle dual to the Power cell dragged by the user, to follow the mouse pointer. Figure 1cde shows a series of snapshots. Quite often, the dragged cells just stay stuck in position (seemingly slipping below the pointer), because the underlying dual circle actually moves with the mouse, but all other circle positions and radii are adapted to preserve the optimal area of the Power cell structure and keep it static for a while, until it suddenly jumps to a better local optimum.

3.2. A force-directed approach

Our solution builds on the duality between the Power diagram and a set of circles [Aur87]. For any set of circles it exists a Power diagram, and when all circle centers are mutually outside other circles, the corresponding Power cells overlap most of their dual circles, contain their respective circle's center, and are approximately matching their circle's area and position (small/big circles correspond to small/big power cells). This can be seen for typical circle packing layout [HL15] and in Figure 1b. It is then tempting to control the circles directly and let the Power cells follow, but we need to maintain low overlapping of the circles to keep these good conditions during the *move-cell* interaction.

Hence we propose to use the best of both worlds to get an interactive Power diagram: only the Power diagram is visualized, but forces applied on its dual circles are used under the hood to get a more predictable interaction and stable treemap during the drag (Figure 3).

3.3. Forces under the hood

Force directed models assume graphical objects (typically nodes of a network) are part of a physical space ((x,y) position) and get mechanical, gravity and electrostatic forces applied to them. We considered the following forces from the `d3-force` API [For] (Physical equations and models are described in full details in the documentation therein):

- **Centering**: translate all objects to maintain the center of gravity of the whole set at the desired position; (Centering can be either `on` or `off`)
- **Many-Body**: simulate gravity (attraction) or electrostatic charges (repulsion) between all pairs of objects (Signed `Strength` can be set).
- **Anti-Collision**: prevent objects from overlapping assigning them an individual `Radius` and a repulsive `Strength`.
- **Positioning**: simulate pulling/pushing forces (springs) for each object to/from its individual desired position (x,y) with some `Strength`.

We assume each child of the parent cell is assigned a circle with two position parameters $c_i(x,y)$ and a radius R_i. At any time, we render the Power diagram of this set of circles. To decide about the position and radius of the circles, we apply the above forces with different parameters depending on the following two modes (Parameters used are summarized in Table 1):

- **Static mode**: We need to find an initial position for the Power cells, hence for the circles, which provides correct area and maintain each circle overlapping its dual cell. We explore the use of the force-directed approach to get such a "static" Power Voronoi treemap. This requires *centering*, *many-body* attraction force, and gradually growing all the circles radius R_i at a factor α of their target weight w_i ($R_i(\alpha) = \sqrt{\frac{\alpha \times A \times w_i}{\pi}}$ with A the area of the parent cell). We also add a soft *anti-collision* force to avoid overlapping of the circles, ensuring existence of the Power cells, and we maintain all circles within the parent cell. We control the growth α of the circles and find α_{opt} which minimizes the error between the area of the Power cells and the target area. The optimal radius for every node $R_i(\alpha_{opt})$ is then used in the Move mode.
- **Move mode**: When the user drags a cell, we first turn off the *centering* force, and turn on the *positioning* force with some strength to pull each circle back to its current "initial" position (the one at the time of entering the drag interaction), enforcing stability of the layout. We also put a stronger strength on the *anti-collision* force so the dragged circle always maintains space for its Power cell to exist by pushing apart the other circles. At the end of the drag action, we instantly set up the new *positioning* forces to maintain each circle at its new current final position, and come back to a stand-by static mode.

Table 1: *Parameters used in the Static and Move modes*

Mode	Centering On/Off	Many-Body strength	Collision strength	Positioning strength
Static	On	1000	1	0
Move	Off	0	2	10

4. Experiments

4.1. A low area accuracy in Static mode

We ran a quantitative experiment to evaluate the quality of the single level treemap we can obtain only using the force-directed approach, and compared it to the `D3.js` implementation [Wei] of Nocaj and Brandes technique [NB12]. We consider a square parent cell and a set of 10 polygonal parent cells for each of $N \in \{10,30\}$ children cells (Figure 2c). For each N, we ran 10 trials. Each trial is a set of N random weights w_i in the range $[0,1]$, assigned to the children, and for each trial we ran 10 random initial position (x,y) for the circles c_i near the center of the parent cell.

Each run consists in searching the optimal α_{opt} scaling factor applying the forces of the Static mode. α lies in the range $[0.1,2]$. It is updated by 0.1 increments in a first pass, then by 0.01 increments and decrements during a second fine-tuned pass, moving upward then backward from the first-pass' optimum. We let the layout stabilizes before trying the next α value. The figures 2ab show for the square parent cell, the distribution of the weighted average $E = \sum_i w_i \varepsilon_i$ of the local errors $\varepsilon_i = \frac{|w_i - A_i(\alpha)|}{w_i}$ where $A_i(\alpha)$ is the area of the Power cell of the dual circle c_i whose radius $R_i(\alpha)$ depends on α. We observe that the error reaches a minimum for $\alpha_{opt} \sim 0.9$, which means when the total area of the circles is 90% of the area of the parent cell. The minimum error is greater than the one of Nocaj and Brandes technique by at least a factor 10. Figures 2de show a similar error difference for the 10 polygonal parent cells when measured at the optimal parameter setting α_{opt} of the force-directed approach.

4.2. A controllable stability/accuracy trade-off in Move mode

The figure 3 shows screenshots of the force-directed Voronoi approach during the Move mode. It can be compared to Figure 1cde showing the modified Weighted Voronoi technique described in section 3.1. The force-directed approach can provide a smoother interaction where the Power cell always follows the mouse pointer so is more *predictable*.

Interestingly, we set up a parameter $\beta \in [0,1]$ to let the user control the trade-off between *stability* and area *accuracy*: $\alpha = \beta \times \alpha_{opt}$.

For low β value (Figure 3 top), all radii $R_i(\alpha)$ are small and the Power diagram tends toward a standard unweighted Voronoi diagram, the areas of the Power cells cannot match the target weights,

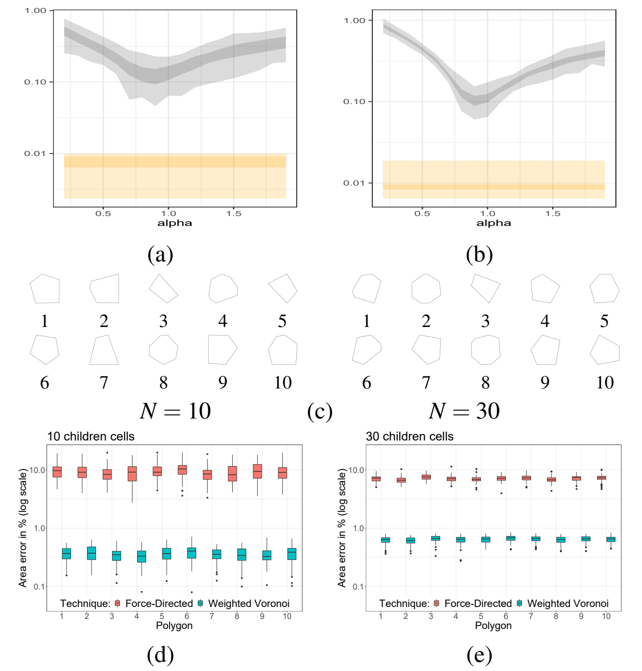


Figure 2: *Areas of the best Force-Directed Power Diagrams are at least* 10 *times less accurate than when computed with the Weighted Voronoi approach in Static mode. On top,* $0-1$ *and* $.25-.75$ *quantile area chart of the weighted average area error E (log scale) of the force-directed approach (grey) versus the standard Weighted Voronoi diagram (orange) for* $N = 10$ *(a) and* $N = 30$ *(b) children of a square parent cell. Optimal Force-directed approach makes about* 10% *error in area on average (See minimum of the top curve reached at* α_{opt}*), while Weighted Voronoi is less than* 1%*. (c) 10 polygons for* $N = 10$ *and* $N = 30$ *children are used as parent cells instead of a square. Their weighted average error distributions for optimal Force-Directed and Weighted Voronoi techniques for* $N = 10$ *(d) and* $N = 30$ *(e) show the same order-of-magnitude difference in accuracy as for the square parent cell (a-b).*

the *accuracy is low*. But as the radii are small, their is fewer chance of collisions, hence the layout *stability is maximum*.

For high β value (Figure 3 bottom), all radii R_i are optimum (depending on α_{opt}), the Power cells areas are better matching the target weights still with 10% average error (see section 4.1), *accuracy is higher*, but now the chances of collision and dramatic change of the layout are far higher, hence the *stability is lower*.

5. Discussion and Future Work

We proposed to use a force-directed approach to compute and update the layout of an interactive Power Voronoi treemap.

We showed this approach is not competitive in terms of areas accuracy compared to Nocaj and Brandes technique [NB12] (10% error instead of less than 1%), so we cannot recommend this setting for rendering a static treemap. However, a way to improve on the area preservation would be to run Nocaj and Brandes weighted

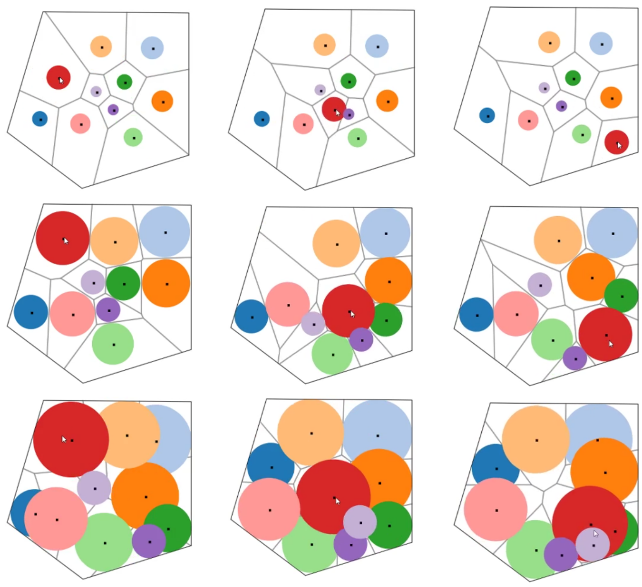


Figure 3: *With the Force-Directed approach, the user can control the stability-accuracy trade-off in Move mode. The red circle/cell is dragged from top left to bottom right in all views. In the top row, small radii* ($\beta = 0.1$) *of the dual circles lead to very stable layout but inaccurate areas of the Power cells. In the bottom row, large radii* ($\beta = 1$) *of the dual circles lead to unstable layout but more accurate areas of the Power cells. The middle row shows intermediate radii* ($\beta = 0.5$) *with better accuracy and still good stability.*

Voronoi technique after final optimal position has been reached in Static or Move mode, by maintaining the positioning forces to enforce stability, and optimizing only for the circles' radii. This is let as a future work.

Still, this approach provides interesting improvement for rendering interactive layout when dragging and moving a cell. We identified the existence of a trade-off between stability and area accuracy during a move, which can be controlled by the user with a slider for instance.

Further quantitative studies of this trade-off would allow determining a default value depending on measures of perceptual area acuity during a *move-cell* interaction in such Power Voronoi treemap. Indeed, it is likely that users are less sensitive to inaccurate areas than to stability of other treemap cells during a *move-cell* interaction, but to which degree is it so? And how can we quantify the stability of siblings during a move? Also, the stability of the children during the move of a parent cell should be considered too as the shape of the parent is changing. An approach has been proposed for dynamic Voronoi treemaps [SFL10], but not yet for Power Voronoi treemaps. At last, implementing a *move-cell* interaction in a multilevel hierarchy, would require to consider moving cells vertically across several levels in addition to the horizontal move proposed here. Hence, many directions are yet to be explored.

References

[Aur87] AURENHAMMER F.: Power diagrams: properties, algorithms and applications. *SIAM journal on computing 16*, 1 (1987), 78–96. doi:10.1137/0216006. 2

[BD05] BALZER M., DEUSSEN O.: Voronoi treemaps. In *Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization* (USA, 2005), INFOVIS '05, IEEE Computer Society, p. 7. doi:10.1109/INFOVIS.2005.40. 1, 2

[BL00] BEAUDOUIN-LAFON M.: Instrumental interaction: An interaction model for designing post-wimp user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), CHI '00, Association for Computing Machinery, p. 446–453. doi:10.1145/332040.332473. 2

[BZSA18] BERNARD J., ZEPPELZAUER M., SEDLMAIR M., AIGNER W.: VIAL: a unified process for visual interactive labeling. *The Visual Computer 34*, 9 (2018), 1189–1207. doi:10.1007/s00371-018-1500-3. 2

[End14] ENDERT A.: Semantic interaction for visual analytics: Toward coupling cognition and computation. *IEEE Computer Graphics and Applications 34*, 4 (2014), 8–15. doi:10.1109/MCG.2014.73. 2

[For] D3.js d3-force API reference. https://github.com/d3/d3-force. Accessed: 2020-04-08. 3

[GSWD18] GÖRTLER J., SCHULZ C., WEISKOPF D., DEUSSEN O.: Bubble treemaps for uncertainty visualization. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (Jan 2018), 719–728. doi:10.1109/TVCG.2017.2743959. 2

[HL15] HAISEN ZHAO, LU L.: Variational circular treemaps for interactive visualization of hierarchical data. In *2015 IEEE Pacific Visualization Symposium (PacificVis)* (April 2015), pp. 81–85. doi:10.1109/PACIFICVIS.2015.7156360. 2, 3

[HR07] HEER J., ROBERTSON G.: Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (Nov. 2007), 1240–1247. doi:10.1109/TVCG.2007.70539. 2

[JS91] JOHNSON B., SHNEIDERMAN B.: Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceeding Visualization '91* (Oct 1991), pp. 284–291. doi:10.1109/VISUAL.1991.175815. 1, 2

[NB12] NOCAJ A., BRANDES U.: Computing voronoi treemaps: Faster, simpler, and resolution-independent. *Computer Graphics Forum 31*, 3pt1 (2012), 855–864. doi:10.1111/j.1467-8659.2012.03078.x. 2, 3, 4

[SB04] SINHA R., BOUTELLE J.: Rapid information architecture prototyping. In *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques* (New York, NY, USA, 2004), DIS '04, Association for Computing Machinery, p. 349–352. doi:10.1145/1013115.1013177. 2

[SFL10] SUD A., FISHER D., LEE H.: Fast dynamic voronoi treemaps. In *2010 International Symposium on Voronoi Diagrams in Science and Engineering* (June 2010), pp. 85–94. doi:10.1109/ISVD.2010.16. 2, 4

[TMB02] TVERSKY B., MORRISON J. B., BETRANCOURT M.: Animation: can it facilitate? *International Journal of Human-Computer Studies 57*, 4 (2002), 247 – 262. doi:https://doi.org/10.1006/ijhc.2002.1017. 2

[WD08] WOOD J., DYKES J.: Spatially ordered treemaps. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1348–1355. 2

[Wei] D3.js Weighted Voronoi Treemap API reference. https://github.com/Kcnarf/d3-voronoi-map. Accessed: 2020-04-08. 2, 3

[YLG*14] YU Z., LU L., GUO Y., FAN R., LIU M., WANG W.: Content-aware photo collage using circle packing. *IEEE Transactions on Visualization and Computer Graphics 20*, 2 (Feb 2014), 182–195. doi:10.1109/TVCG.2013.106. 2