Voronoi Treemaps

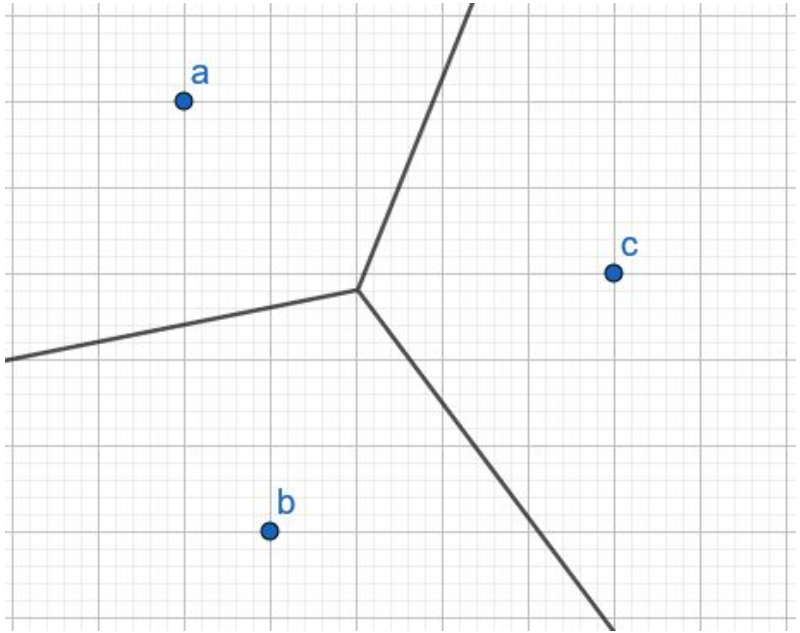
An introduction & explanation.

Christopher Oser

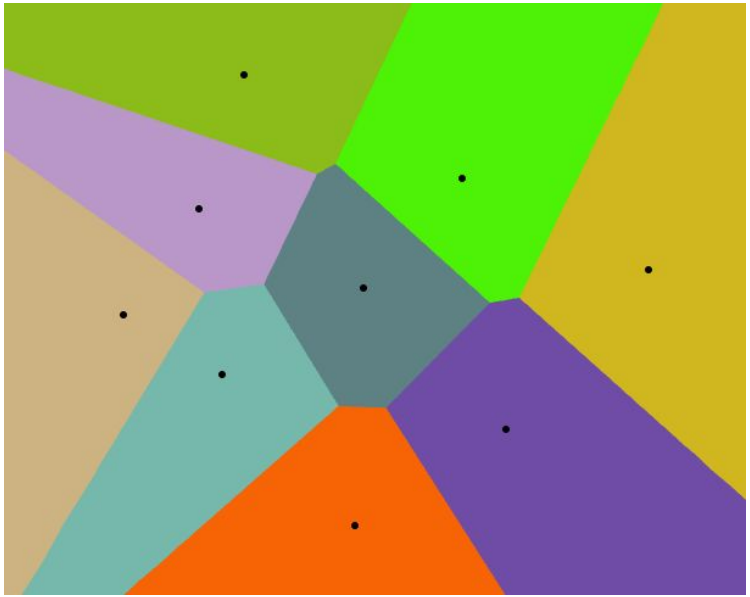
05 May 2021

DiplomandInnen Seminar Presentation

What are Voronoi diagrams ?



Constructed and captured by Christopher Oser using GeoGebra.



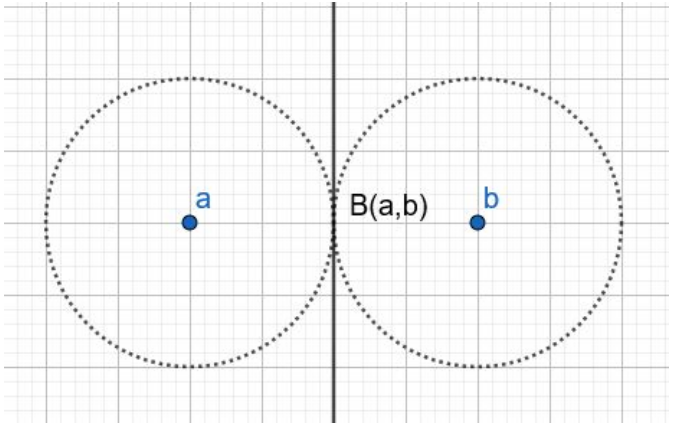
Constructed and captured by Christopher Oser using Alex Beutel's Interactive Voronoi Diagram Generator.

Basic Theory

- Set S of n sites, whereby n ≥ 3.
- Sites are on an Euclidean plane.
- Euclidean distance between two points, a and b, given as:

$$d(a,b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

- Bisector B(a,b) is the locus of all points at equal distance of a and b.



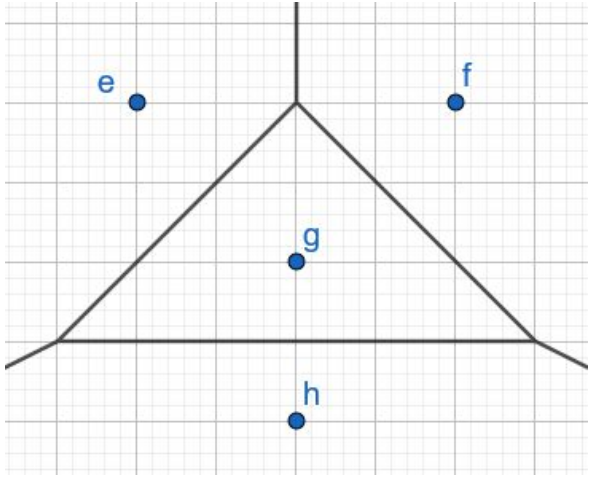
Constructed and captured by Christopher Oser using GeoGebra.

Voronoi Region

- The bisector distinguishes the half plane:
  $H(a,b) = \{x \mid d(a,x) \leq d(b,x)\}$

- The Voronoi region is the intersection of all n-1 half planes H(a,x), where x ranges over all other points in the set S.

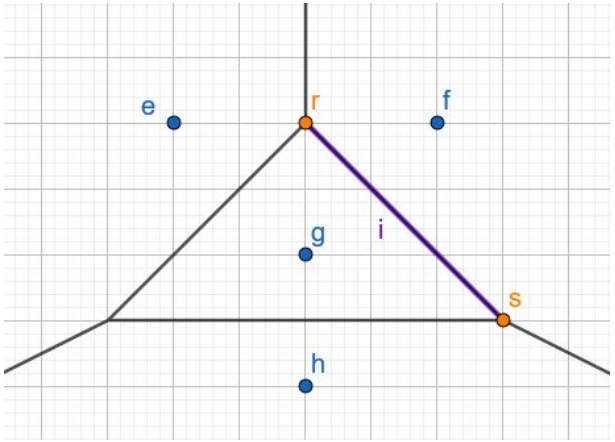
- So the Voronoi region V(a,S) contains all points closest to a in respect to the set S.



Constructed and captured by Christopher Oser using GeoGebra.

Some more definitions

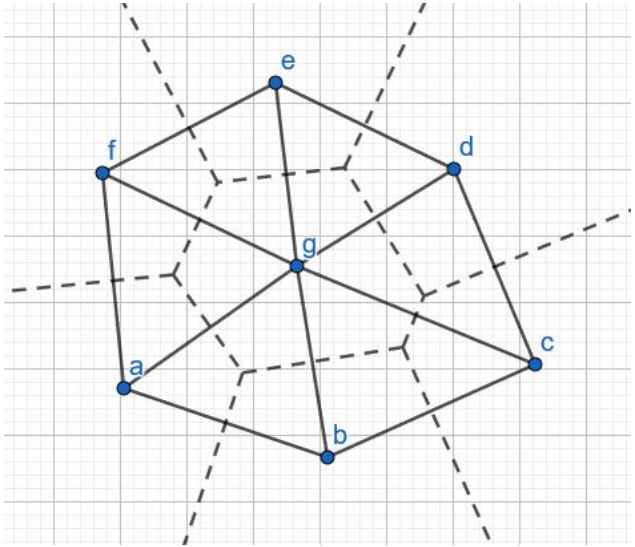
- The boundary between two Voronoi regions is a Voronoi edge.
- Endpoints of Voronoi edges are called Voronoi vertices.



Constructed and captured by Christopher Oser using GeoGebra.

Delaunay Triangulation

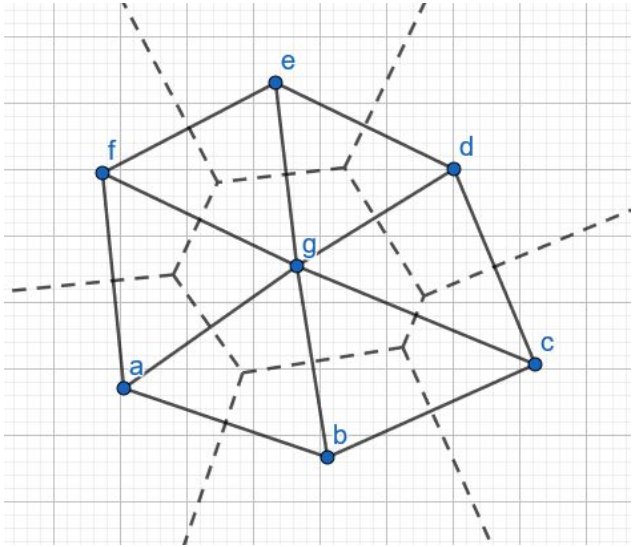
- Delaunay Tessellation is constructed by connecting any two points a,b of S, for which there is a circle that passes through them but contains no other points in S.
- If S in general position ⇒ tessellation turns into triangulation.
- Delaunay Tessellation/Triangulation is the dual of Voronoi diagram.



Constructed and captured by Christopher Oser using GeoGebra.

Delaunay Triangulation

- Delaunay edge ⇔ Voronoi edge
- Delaunay face ⇔ Voronoi vertex
- Delaunay vertex ⇔ Voronoi region
- Duality useful in many algorithms.
- If you have one, you can get the other.



Constructed and captured by Christopher Oser using GeoGebra.

Construction Algorithms

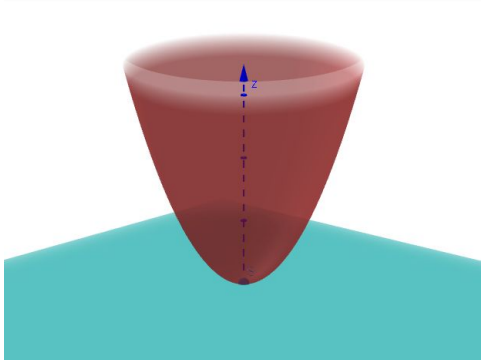
- Incremental:
  - Points are added one at a time.
  - In every step conflicts are analyzed & diagram is adapted.
  - Time: $O(n^2)$ - worst case
  - BUT randomized/expected: O(n log n)

- Divide & Conquer:
  - S is split in two equal sets ⇒ L and R.
  - Recursively split sub-sets until only three or two points left.
  - Merge sub-diagrams to the final Voronoi diagram.
  - Time: O(n log n)

Construction Algorithms

- Plane Sweep (Fortune's Algorithm):
  - Vertical line sweeps over plane.
  - Points are stored in y-order.
  - Voronoi regions are updated accordingly.
  - Time: O(n log n)

- Lift to 3-space
  - Project plane to 3rd dimension. ($z = x^2 + y^2$)
  - Use existing algorithms to compute convex hulls of n points.
  - Translate results back to 2 dimensional space.
  - Time: O(n log n)



Constructed and captured by Christopher Oser using GeoGebra.

Voronoi Diagram Implementations

Incremental:

- LEDA [C++]
  https://www8.cs.umu.se/kurser/TDBAfl/VT06/algorithms/WEBSITE/IMPLEMEN/LEDA/IMPLEMEN.HTM
- Khuyen Tran [Python]
  https://github.com/khuyentran1401/Voronoi-diagram

Divide & Conquer:

- Alex Shavlovsky [Java]
  https://github.com/alexshavlovsky/VoronoiDiagramJavaRecursive

Sweep-line:

- Mathias Westerdahl [Java]
  https://github.com/JCash/voronoi
- Raymond Hill [JS]
  https://github.com/gorhill/Javascript-Voronoi
- D3-voronoi [JS library]
  https://github.com/d3/d3-delaunay

…

Lift to 3-space:

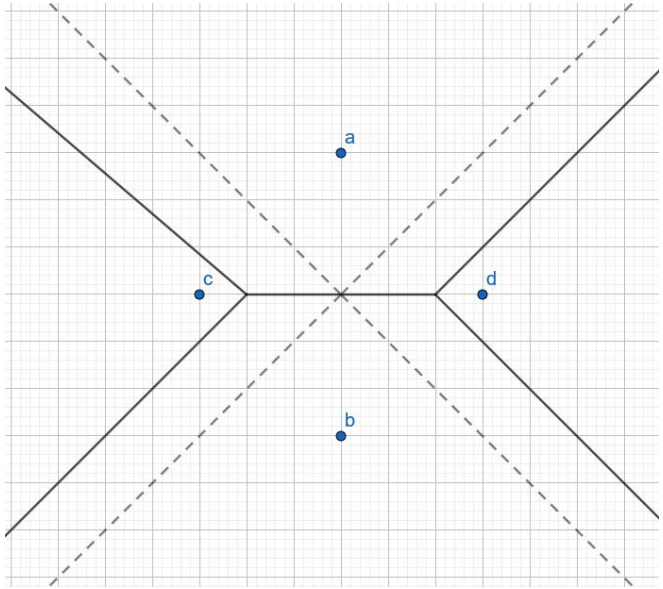
- Qhull [C]
  http://www.qhull.org/

Applications

- Computer Science:
  - Computing Textures
  - Classification
  - Clustering
  - Localization (closest point)
- Biology: Modelling structures.
- Chemistry: Computing atomic charges.
- Geometry: Nearest neighbor queries.
- Meteorology: Analyzing spatially distributed data.

Weighted Voronoi Diagrams

- In DataVis we want to weight visualizations according to data.
- Weights are assigned to the points in S.
- Weights influence size of Voronoi region.
- Makes it possible to customize size:
  - Bigger weight ⇒ larger region
  - Smaller weight ⇒ smaller region
  - All other properties remain the same
- Usually computed using power diagrams. But also:
  - Additively weighted
  - Multiplicatively weighted



Constructed and captured by Christopher Oser using GeoGebra.

Power Diagram

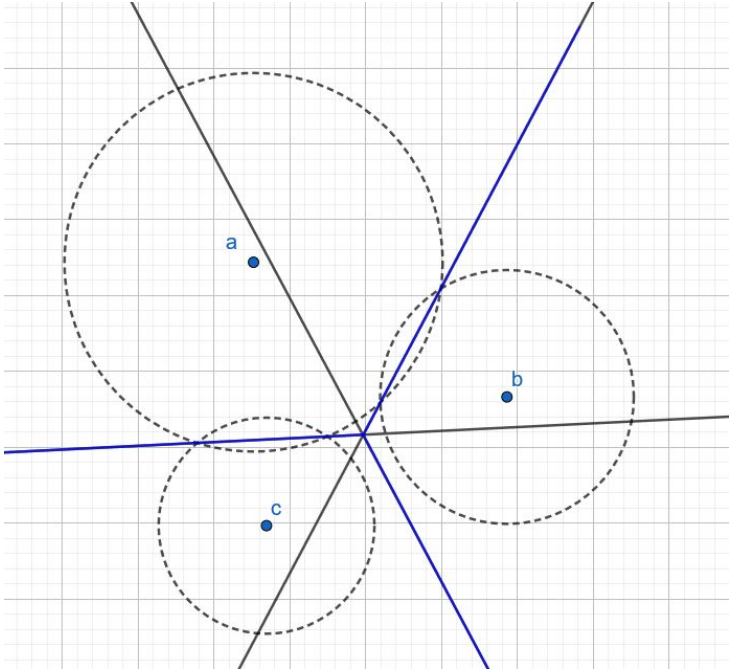
- Voronoi-like
- Uses Circles/Spheres to define area of influence:

  $Radius\ r = \sqrt{w(p)}$

- Power Function:

  $pow(x,p) = (x - p)^T\ (x - p) - w(p)$

- Intersection of areas result in diagram.



Constructed and captured by Christopher Oser using GeoGebra.

Other weighting methods

- Additively:
  - *add_distance(x, p) = distance(x, p) - weight*

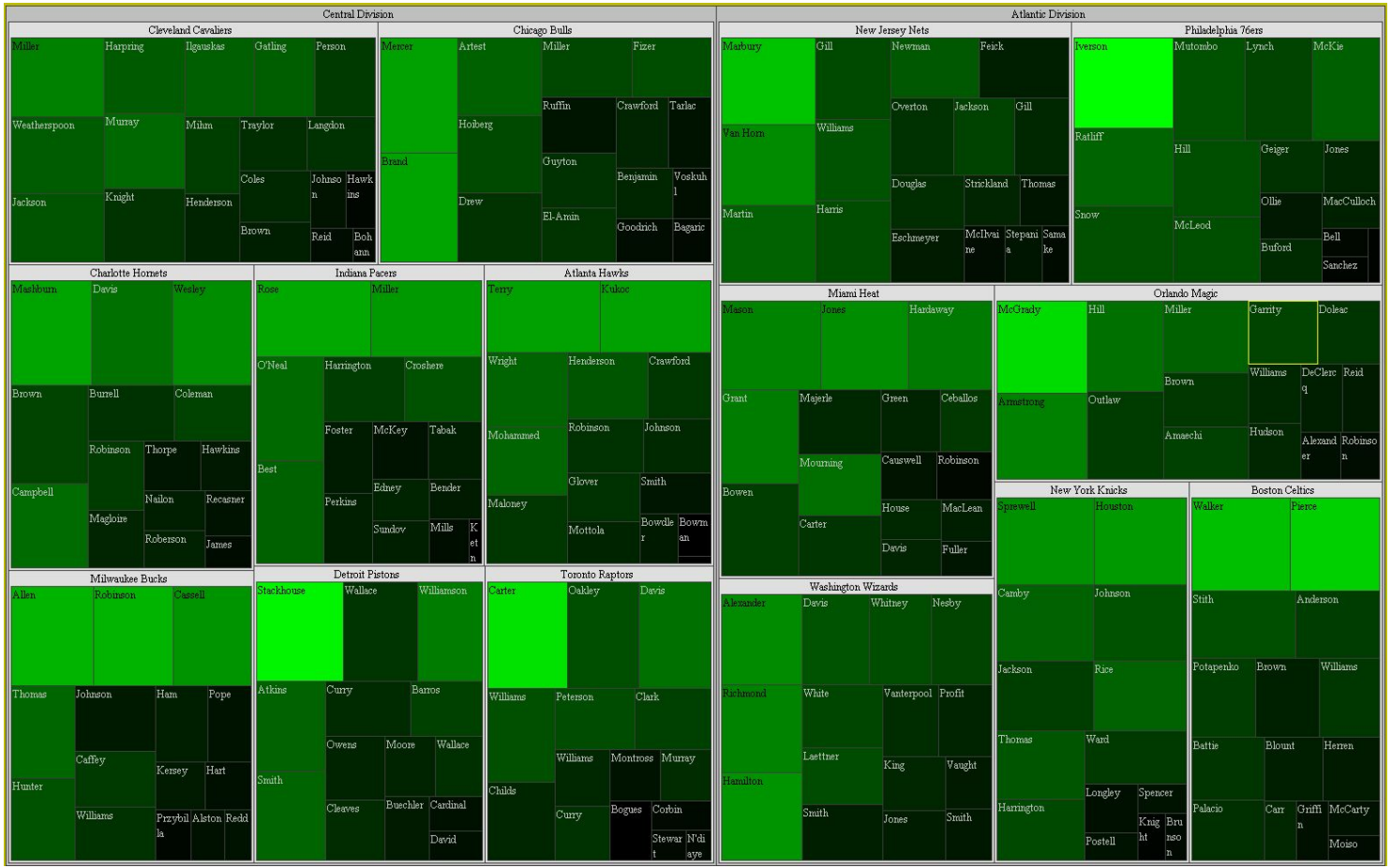
- Multiplicatively:
  - *mul_distance(x, p) = distance(x, p) / weight*

- Both yield curved diagrams, which is why we focus on power diagrams.

Treemaps

- Method does not originate in Voronoi diagrams.
- Used to display hierarchical data.
- Nested polygons (usually rectangles) visualize the hierarchical nature of the data.
- Can be weighted by additional attributed data.
- Treemap Software developed by Ben Shneiderman's team at University of Maryland.

Constructed and captured by Christopher Oser using [TreeMap 4.1.1](http://www.cs.umd.edu/hcil/treemap/).

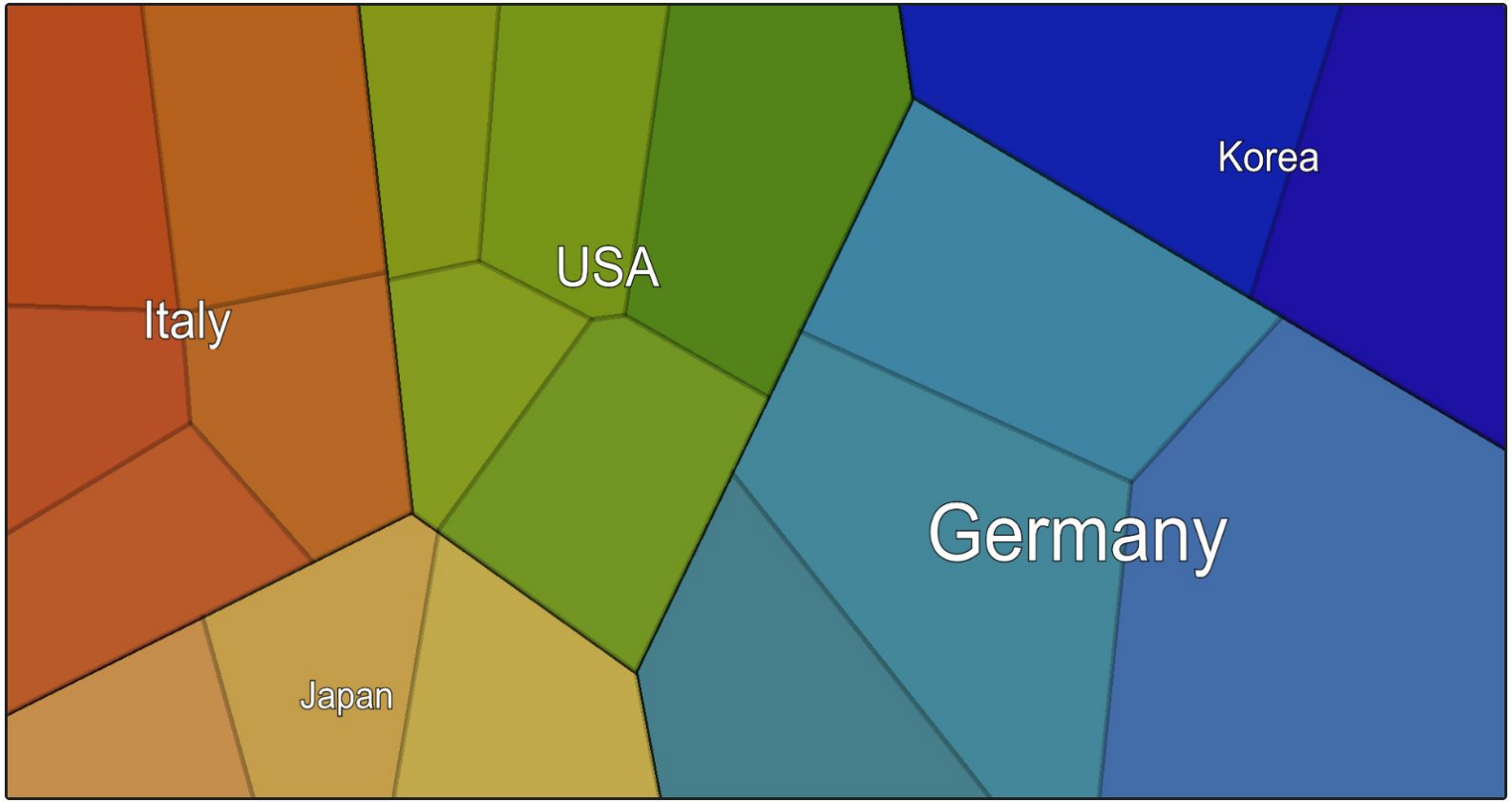


Constructed and captured by Christopher Oser using TreeMap 4.1.1.
http://www.cs.umd.edu/hcil/treemap/

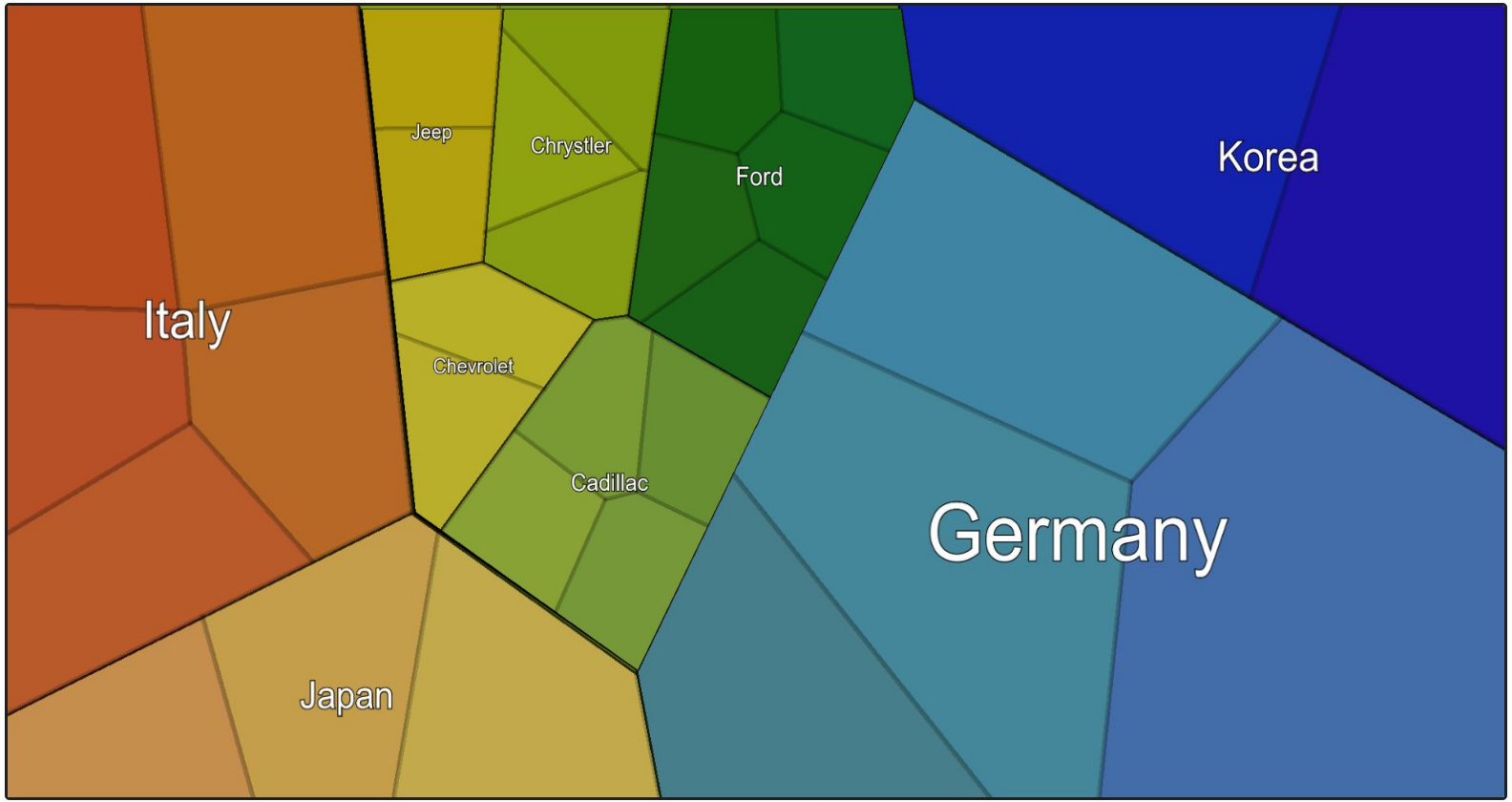
Constructed and captured by Christopher Oser using TreeMap 4.1.1.
http://www.cs.umd.edu/hcil/treemap/

Weighted Voronoi Treemaps

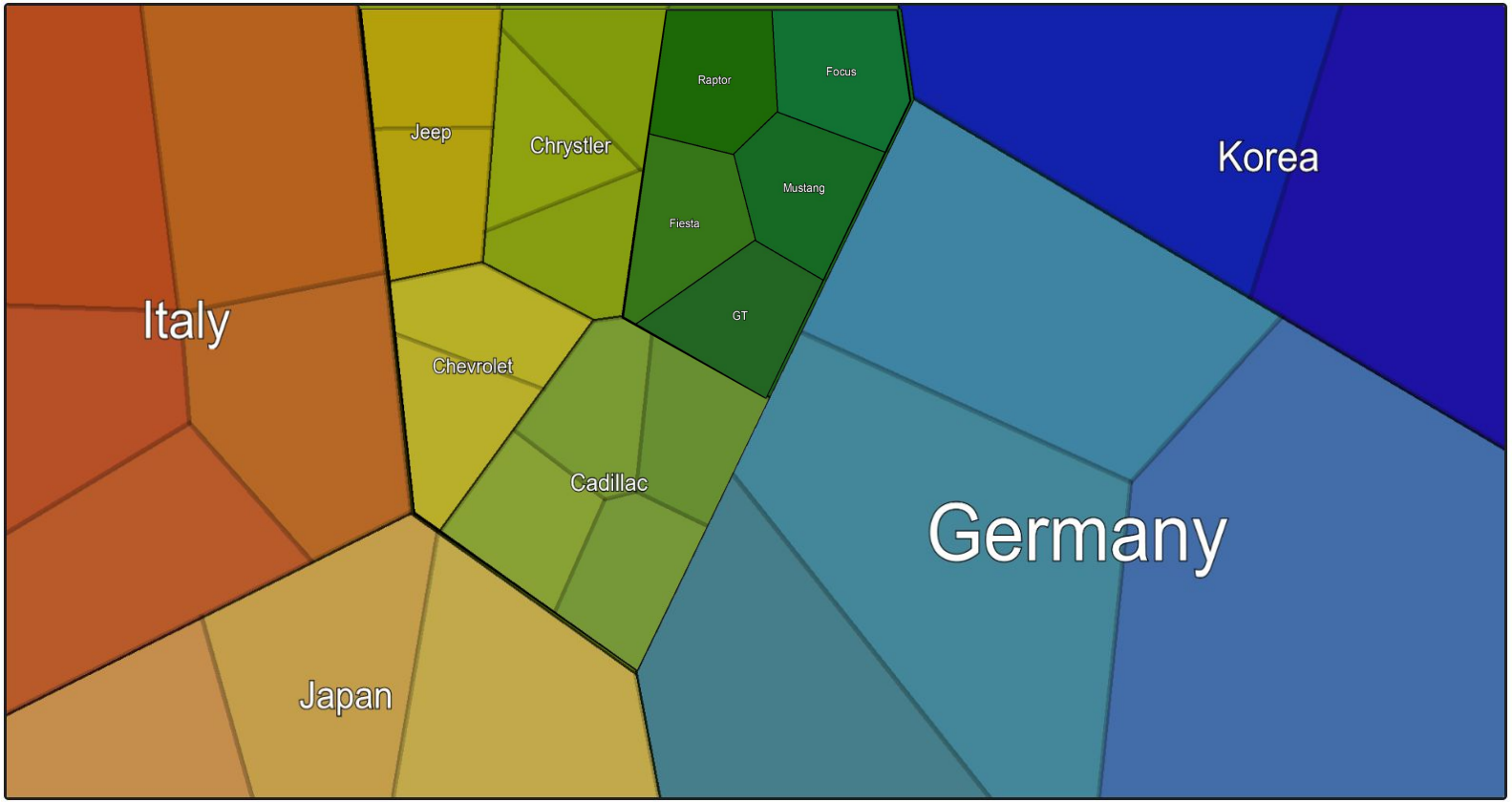
- Use tree mapping to visualize hierarchical data.
  - Voronoi diagram for each layer
  - Embed child diagrams in parent Voronoi region
- Instead of rectangles ⇒ Voronoi regions.
- Assign weights:
  - Arbitrarily, for customization
  - According to children



Constructed and captured by Christopher Oser using IVT.



Constructed and captured by Christopher Oser using IVT.



Constructed and captured by Christopher Oser using IVT.

History of Voronoi Treemaps

- First introduced: InfoSky by Andrews et al, 2002.
  https://doi.org/10.1057/PALGRAVE.IVS.9500023
  - Used in visual explorer.
  - Large hierarchical datasets visualized in voronoi diagram.

- Also thoroughly dealt with in “Voronoi Treemaps” by Balzer, 2005.
  https://doi.org/10.1109/INFVIS.2005.1532128
  - Establishes overview.
  - Addresses theory in detail.
  - Christened the technique

Voronoi Treemap Libraries

- D3-voronoi-treemap [JS]
  https://github.com/Kcnarf/d3-voronoi-treemap
- ArlindNocaj Voronoi-Treemap-Library [Java]
  https://github.com/ArlindNocaj/Voronoi-Treemap-Library
- Voronoi Treemaps in R - Paul Murrell [R]
  https://www.stat.auckland.ac.nz/~paul/Reports/VoronoiTreemap/voronoiTreeMap.html
- Tableau template by Tristan Guillevin [Tableau]
  https://ladataviz.com/2020/01/02/build-a-voronoi-treemap-in-tableau-in-two-steps/

Treemap Applications

- FoamTree
  - Commercial software
  - Not open source
  - Showcase video
    https://youtu.be/-8ZbKzipfTI

Constructed and captured by Christopher Oser using FoamTree.

- IVT
  - Developed at TU Graz
  - Oser, Ruplitsch, Weissl, Gruber
  - Open source
  - Uses d3-voronoi-treemap (for now)
  - Showcase video
    https://youtu.be/phWakUsk-7Y

Constructed and captured by Christopher Oser using IVT.

Conclusion

- Voronoi Diagram introduction.
- Focus on Voronoi Treemaps.
- Visualizing hierarchical data.
- Interactive visualizations are desirable.
- Further work on IVT to come (Master Thesis).