

Tessellating Urban Space based on Street Intersections and Barriers to Movement

GI_Forum 2017, Issue 1
Page: 114 - 125
Full Paper

Corresponding Author:
anitagraser@gmx.at

DOI: 10.1553/giscience2017_01_s112

Anita Graser

AIT Austrian Institute of Technology, Vienna, Austria

Abstract

This paper presents a novel method to partition urban space with respect to its mobility network rather than abstract administrative boundaries or simple grids. The main goal of our method is to support mobility infrastructure planning and optimization in an urban environment. To this end, our tessellation, which is based on constrained Voronoi tessellation, generates cells that are compact and centred around street intersections. Furthermore, it takes into account unsuitable areas and barriers that obstruct movement through urban space. The resulting cells are fine-grained planning units that reflect the characteristics of the underlying mobility network. Potential applications include demand modelling, location planning and optimization for bicycle- and car-sharing systems, or public transport stops.

Keywords:

Tessellation, street network, constrained Voronoi diagram, mobility infrastructure planning

1 Introduction

Spatial partitions come in a variety of forms. They can be based on coordinates alone, as in grids, based on manually drawn boundaries, or formed automatically by generators based on input seed locations, as in Voronoi diagrams (Gold, 2016). Tessellations subdivide space using one or more geometric shapes with no overlaps and no gaps. More specifically, common tessellations of urban space are city blocks, districts, postal areas, and regular square or hexagonal grids. These tessellations are widely used in fields such as urban planning, spatial marketing (Bradlow et al., 2005) or location optimization (Asamer et al., 2016) to model customer distribution, workplace data, and various other variables that influence demand.

However, these common tessellations are not well suited for planning mobility infrastructure, such as stations for bicycle- and car-sharing systems, or public transport stops, due to various shortcomings. City blocks or districts focus on the area between streets rather than the street space itself. In mobility infrastructure planning, when a location recommendation refers to a city block, it remains unclear for a planner which streets

enclosing the block should be preferred. In bicycle-sharing system planning, for example, a particularly suitable station location at an intersection could be right at a corner where multiple city blocks meet. On the other hand, regular square or hexagonal grids are sensitive to the selected cell size. If cell sizes are small, it is likely that the resulting tessellation will contain cells that have no connection to the street network. Large cells, on the other hand, may contain areas that are actually disconnected due to barriers, such as rivers or rail infrastructure without crossings. Furthermore, tessellations with large cells are of limited use for planning mobility systems that should be highly accessible to pedestrians, since most pedestrians will only walk a limited distance of few city blocks, up to 800m (0.5 mile) to rail transit stations (Weinstein et al., 2008), and less for local bus stops or bicycle-sharing stations.

We therefore propose a tessellation where cells are centred around street intersections in order to support mobility infrastructure demand modelling (Krykewycz et al., 2010, Rudloff & Lackner, 2014), planning and optimization (Asamer et al., 2016) in an urban environment. This approach makes it possible to perform location optimization without the need for manually predefined location candidates. Using groups of nodes or grid cells as potential locations rather than all available network nodes reduces the location optimization problem, which is NP-complete and therefore computationally intractable for realistic network sizes of hundreds to thousands of nodes (Owen & Daskin, 1998). The tessellation can be used in demand forecasting to predict demand per cell (based on number of residents, workplaces, access to public transport stations, education and other facilities), as well as subsequent location recommendations where certain cells are recommended as locations for new infrastructure. Our tessellation method supports modelling of unsuitable areas, for example parks or pedestrian-only areas where no new mobility infrastructure (such as stations of bicycle- and car-sharing systems) should be added, and barriers, such as rivers or rail infrastructure without crossings.

The remainder of this paper is structured as follows: Section 2 introduces our proposed method for determining tessellation seeds and performing the tessellation; Section 3 presents results of applying these methods to a use case of planning optimal locations for a bicycle-sharing system in the city of Vienna, Austria; we discuss the results and draw conclusions in Section 4.

2 Methodology

To support mobility infrastructure planning and optimization in an urban environment, our tessellation cells should be compact and centred around street intersections. Locations near intersections are preferable for new mobility infrastructure (such as stations for bicycle- and car-sharing systems), since they are, by definition, accessible from multiple directions. This makes them more accessible for both users and maintenance (such as reallocation of vehicles). This preference is confirmed by an analysis of current bicycle-sharing stations in Vienna, which are mostly located near intersections.¹ Furthermore, the tessellation should

¹ <https://www.citybikewien.at/en/stations/stations-map>

Voronoi tessellation is a way to naturally partition space into sub-regions for modelling spatial structures and locational optimization (Okabe et al., 2009). However, ordinary Voronoi tessellation does not respect constraints posed by unsuitable areas and barriers. Simply removing these areas from Voronoi cells leads to inconsistencies, because cells can be split into multiple parts, particularly by narrow barriers (as illustrated in Figure 2). We therefore need a way to construct a constrained Voronoi tessellation.

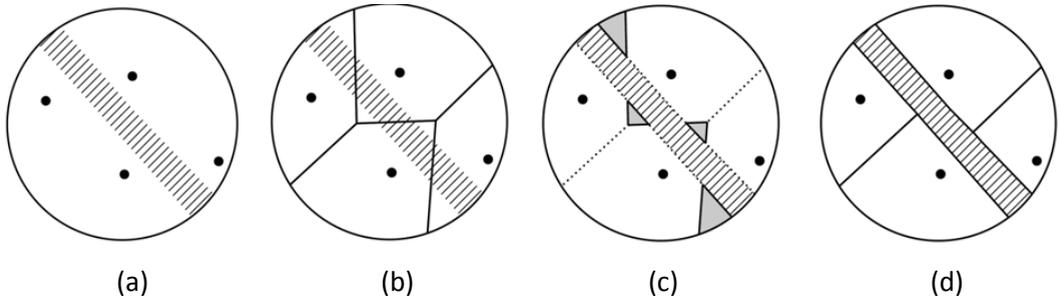


Figure 2: (a) Four tessellation seeds (points) and a barrier (hatched); (b) ordinary Voronoi tessellation; (c) Artefacts (grey areas) from splitting Voronoi cells along barrier; (d) constrained Voronoi tessellation

Ordinary Voronoi tessellations can be constructed using different algorithms, including Fortune's algorithm (Fortune, 1987) and the Bowyer-Watson algorithm (Bowyer, 1981; Watson, 1981). Fortune's algorithm is based on a sweep line approach. The Bowyer-Watson algorithm constructs a Delaunay triangulation from which the dual representation Voronoi tessellation can be derived. Similarly, constrained Delaunay triangulation (CDT), as described for example by Chew (1989), has a dual representation that is a type of Voronoi tessellation. However, cells in this dual graph can overlap each other (Chew, 1989; Aurenhammer & Klein, 2000). For a thorough discussion of algorithms for ordinary and constrained Voronoi tessellation and Delaunay triangulation, we refer the reader to Aurenhammer & Klein (2000). Research in this area is still ongoing; improved algorithms to compute network Voronoi diagrams (Okabe et al., 2008) or Voronoi tessellations with constraints (Tournois et al., 2010) have been proposed.

To the best of our knowledge, constrained Voronoi tessellation is not available in current GIS applications. While ArcGIS supports constrained Delaunay triangulation, Voronoi tessellation can only be performed on unconstrained Delaunay triangulations (ESRI, 2016a). An alternative, which is available in both ArcGIS (ESRI, 2016b) and GRASS GIS (GRASS Development Team, 2016), is to compute a cost allocation raster – that is, for each cell, its nearest source (in our case a tessellation seed) based on the least accumulative cost over a cost surface. We do not pursue network Voronoi diagram approaches (Okabe et al., 2008), since network-based approaches are more suitable to generate Voronoi cells with a greater spatial extent than are required by our approach. The strength of network-based Voronoi diagrams is that network connectivity is considered during cell generation. Since the cells in our approach, which is focused on use cases such as bicycle-sharing system planning, only encompass a small area around individual intersections, network connectivity does not come into play.

Figure 3 shows the tessellation process implemented as a QGIS Processing model (Graser & Olaya, 2015) using GRASS GIS algorithms. The model input parameters include the planning area, tessellation seeds and raster cell size (provided as the length of the side of a square raster cell). First, the cost raster is derived from the input planning area. Barriers are modelled as holes in the planning area and result in cost raster cells with NULL values. The cost allocation raster is then computed using the previously generated tessellation seeds as starting points. The resulting cost allocation is then vectorized to obtain a preliminary tessellation. It is worth noting that it is necessary to clean the vectorized cost allocation raster from areas smaller than or equal to the input raster cell size squared. These single raster cells are artefacts of the cost allocation raster (particularly at allocation area corners with acute angles), or artefacts from seeds that were located inside barriers. Finally, the geometries are generalized to smooth the jagged edges of the vectorized areas.

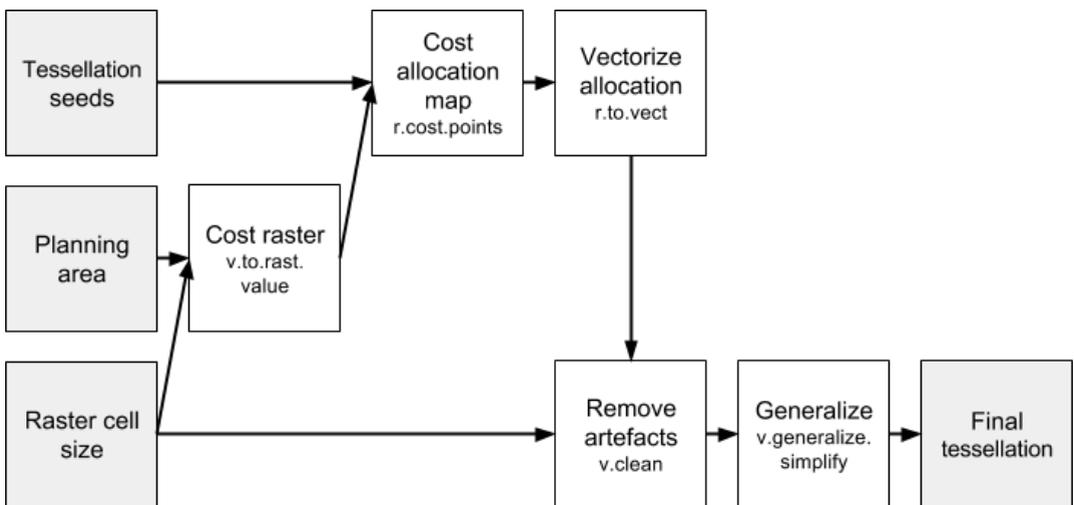


Figure 3: Flow chart of the tessellation implementation in QGIS Processing using GRASS GIS algorithms.

3 Results

In this section we apply our tessellation to a use case of planning optimal locations for a bicycle-sharing system in the city of Vienna, Austria. Since distances between stations in successful bicycle-sharing systems are as low as 300m (García-Palomares et al., 2012; O'Brien et al., 2014), it is particularly important to plan at the level of intersections, and the high number of stations required calls for an objective automated planning process. We compare the results of different clustering methods on tessellation seeds and the resulting tessellations. The test area is depicted in Figure 4. It covers the inner city, with a bicycle network of 23,364 nodes. The bicycle network was derived from raw OpenStreetMap data by extracting ways that can be used by cyclists and making them routable.

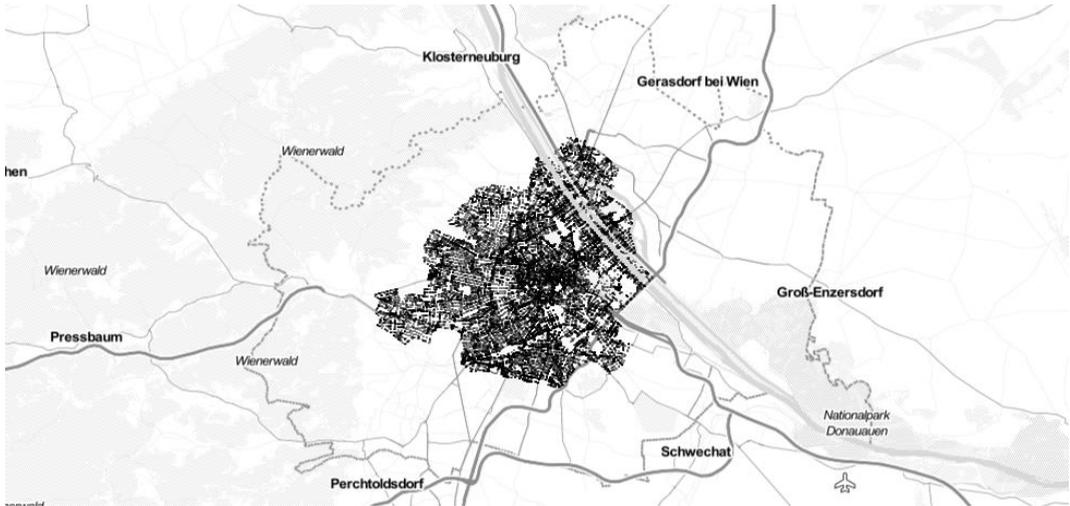
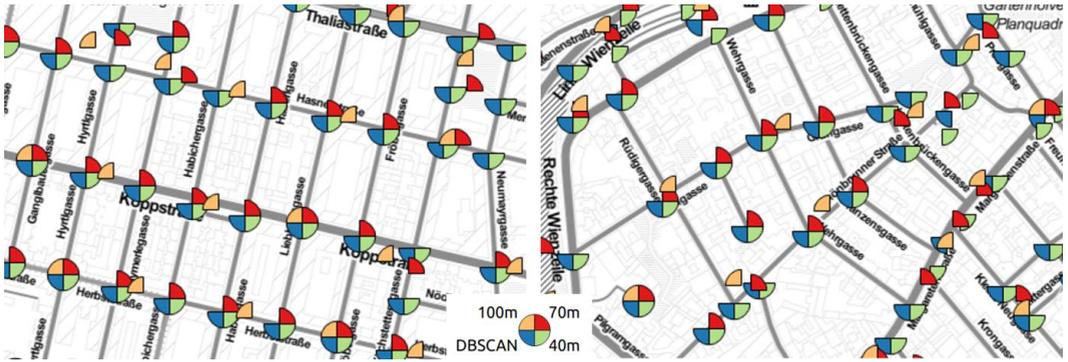


Figure 4: Map with the border of Vienna and the bicycle network nodes. This and the following maps use background map tiles cc-by Stamen Design, data © OpenStreetMap contributors.

First, we compute tessellation seeds using convex clustering and DBSCAN clustering with a neighbourhood size of 40m, which was determined empirically to be large enough to contain all nodes of an intersection while being small enough, in most cases, to avoid the merging of neighbouring intersections. Figure 5a shows the results of convex clustering for three target cell sizes (40m, 70m and 100m), as well as DBSCAN clustering in a part of the city where the block size is roughly 75m by 150m. For comparative purposes, Figure 5b shows the resulting tessellation seeds for an area with irregular street layout. Convex clustering with 40m target size creates the highest number of seeds ($n=9,563$), followed by DBSCAN clustering ($n=7,249$); seeds are located at almost every intersection, with only a few cases of merged neighbouring intersections. By contrast, convex clustering with 70m ($n=5,998$) or 100m ($n=3,822$) merges considerably more intersections. This illustrates the greatest challenge of convex clustering – that is, to select a suitable target cell size, particularly in cities where block size varies considerably. DBSCAN clustering does not require a target cell size and adapts to variations in block size. This also means that we cannot affect the number of resulting tessellation seeds from DBSCAN clustering, while we can influence the number of seeds from convex clustering by varying the target cell size. Overall, the reduction of potential locations from the initial network nodes ($n=23,364$, shown in Figure 4) to the clustering results (n between 3,822 and 9,563) significantly reduces the size of the location optimization problem, which is NP-complete.



(a) Seeds in an area with grid-based layout (b) Seeds in an area with irregular street layout

Figure 5: Local comparison of different tessellation seed-creation options. Circle quarters represent seeds generated using convex clustering for cell size 100m (orange, $n=3,822$), 70m (red, $n=5,998$), and 40m (green, $n=9,563$), as well as DBSCAN clustering (blue, $n=7,249$). With 40m convex clustering and DBSCAN, seeds are located at almost every intersection. By contrast, convex clustering with 70m or 100m merges considerably more intersections.

For the cost raster, we opted for a raster resolution of 5m per cell, which is small enough to preserve necessary details (such as narrow barriers) but large enough to keep the subsequent computation time of the cost allocation raster low. Each passable cell has a value of 1 and barriers are modelled as cells with NULL values. The cost raster dimensions are 2,632 by 2,420 cells. Both cost raster and tessellation seeds are then used to compute the cost allocation map, which in turn is vectorized to generate the desired vector tessellation cells.

Figure 6 provides a comparison of the results of tessellation using convex clustering with a target cell size of 70m and DBSCAN clustering. In many areas both seeds create a similar tessellation, but in regions with smaller blocks convex clustering starts to merge areas around multiple neighbouring intersections. The difference in the number of seeds and the number of resulting tessellation cells is caused by some seeds being located in unsuitable areas and therefore being discarded during the tessellation process. Figure 6 also illustrates some advantages of constrained tessellation cells over regular square grid cells. In a regular grid with a similar number of grid cells (6,319 compared to 5,830 for convex clustering or 7,082 for DBSCAN clustering), many cells contain areas that are divided by barriers. Splitting these cells along the barriers would result in artefacts of which some would be very small and not connected to the mobility network. Furthermore, the location of cell borders in a regular grid is dependent on an arbitrarily defined grid origin location (such as a corner of the bounding box around the planning area) rather than on any characteristics of the mobility network.

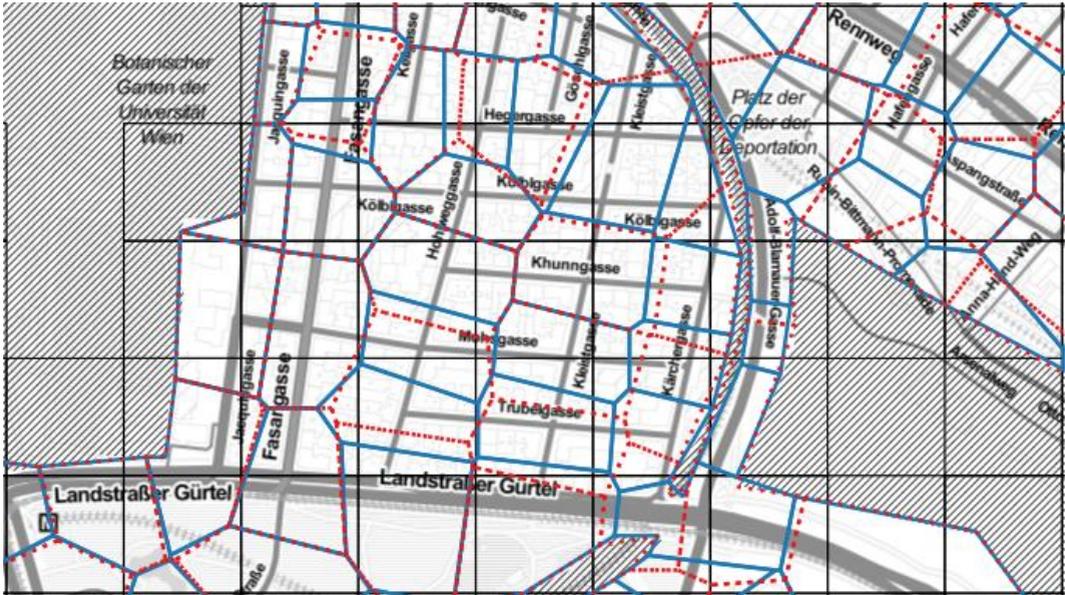


Figure 6: Constrained tessellations using convex clustering seeds (dashed red, $n=5,830$ for a target cell size of 70m) and DBSCAN clustering seeds (solid blue, $n=7,082$). A regular square grid with 180m cell size is provided for comparison (solid black, $n=6,319$). Hatched areas are tessellation barriers.

The error introduced by computing Voronoi-like cells from a cost allocation raster rather than using vector Voronoi algorithms is negligible for our use case. Figure 7 shows a regular Voronoi diagram overlaid with a cost allocation raster tessellation in an area which is not affected by barriers and is therefore suited for comparison. As expected, the greatest observed distances between cell borders correspond to the chosen cost raster cell size of 5m. The distances can be decreased by computing a more fine-grained cost raster.

Smaller raster cells increase computation time. Table 1 gives the runtime of the cost allocation raster tessellation for 5,830 seeds (from convex clustering with 70m) on a single core of an Intel® Core™ i7-4600U CPU @ 2.10GHz × 4 CPU. By default, the most expensive computation – that is, the `r.cost` function – is configured to use up to 300MB of memory. This value was increased to 1GB for 2m rasters and 4GB for 1m rasters. For comparison, ordinary GRASS Voronoi tessellation takes 4.3s to construct a tessellation for the same 5,830 seeds.

Table 1: Runtime of the cost allocation raster tessellation of 5,830 seeds (from convex clustering with 70m) on a single core of an Intel® Core™ i7-4600U CPU @ 2.10GHz × 4 CPU (runtime values based on 7 independent runs).

Raster size [m]	5	4	3	2	1
Runtime [s]	34.3 – 34.7	39.5 – 40.4	50.2 – 51.0	80.3 – 81.6	237.7 – 250.7



Figure 7: A comparison between vectorized cells from a cost allocation raster (dashed blue) and ordinary Voronoi cells (solid pink) in an area without barriers shows that both approaches achieve similar results.

To illustrate the benefits of our constrained tessellation approach for planning mobility infrastructure, cells from DBSCAN clustering were used to compute different scenarios for bicycle-sharing system planning in Vienna. (The exercise was hypothetical, Vienna already having a bicycle-sharing system.) As inputs for the planning process, relevant values, such as number of inhabitants, work places and other attractors, were determined for each cell. Additionally, a distance matrix describing travel times between all cells was computed. Since our cells are centred on network intersections, computing network travel times between cells is straightforward using cell centres as start and end locations. In comparison, the definition of distances between cells in a regular grid would be much less clear, since cell centres might be further away from network nodes and some cells might not have any network connection at all. Figure 8 shows preliminary bicycle-sharing demand forecasts for each cell, which will be used to determine optimal locations. Both the demand forecasting and location optimization model are still under development and therefore further results are not yet available.

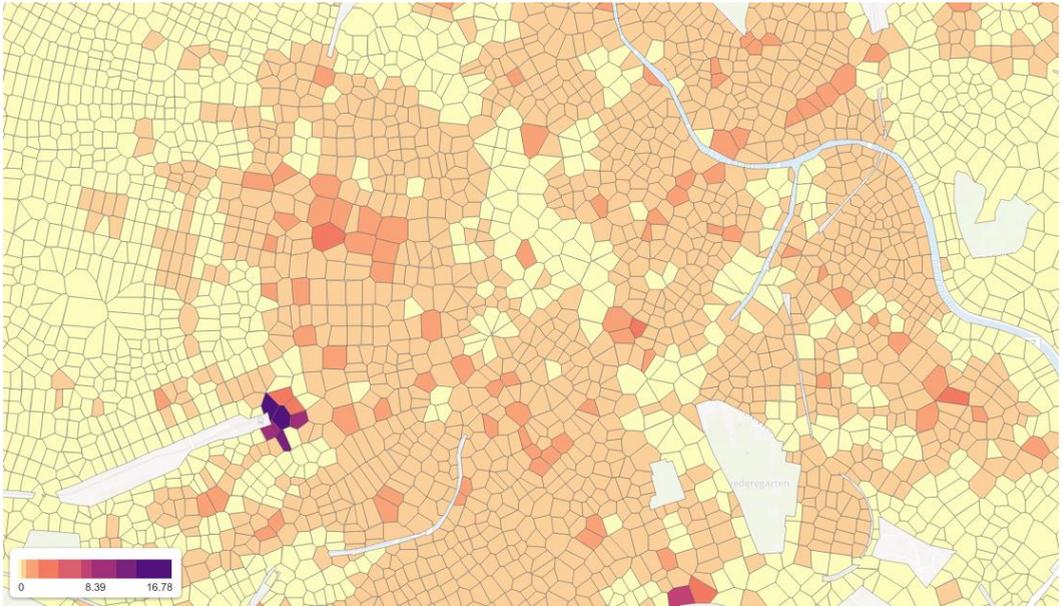


Figure 8: Forecast bicycle-sharing demand (in number of trips during the morning peak hours 07:00–11:00 on an average work day) originating in the cell for Vienna city centre with the highest values, at Westbahnhof in the south-western corner. Grey areas represent tessellation barriers.

4 Discussion and Conclusion

Tessellating urban space based on street intersections and barriers to movement presents two distinct challenges: extracting suitable tessellation seeds from the underlying mobility network, and constructing a tessellation that accounts for unsuitable areas and barriers that obstruct movement through urban space. We presented a novel tessellation method which tackles both challenges and which is specifically designed to enable the demand modelling, location planning and optimization of mobility systems.

We presented two methods for extracting tessellation seeds. Convex clustering requires the specification of a target cell size, while DBSCAN clustering adapts to the city block size, which can vary throughout the planning area. This means that, using DBSCAN clustering, we cannot affect the number of resulting tessellation seeds, while we can influence the number of seeds from convex clustering by varying the target cell size. If an application’s computation time is sensitive to the number of input cells, it might be necessary to resort to convex clustering and increase the target cell size. In many cases, for other applications which can accommodate a higher number of input cells, the adaptive cell sizes of DBSCAN clustering will be preferable.

Since current GISs lack support for constrained Voronoi diagrams, we opted for a solution based on computing cost allocation rasters to account for unsuitable areas and barriers. Our results show that the error introduced by computing Voronoi-like cells from a cost allocation

raster rather than using vector Voronoi algorithms is negligible for use cases in mobility infrastructure planning.

Potential applications include demand modelling, location planning and optimization for systems that should be highly accessible to pedestrians, since most pedestrians will only walk a limited distance of a few city blocks. Examples of such systems include but are not limited to bicycle- and car-sharing systems, and public transport stops. A demand forecasting model for bicycle-sharing systems based on this tessellation method is currently under development.

Acknowledgements

This work was supported by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) within the programme ‘Mobilität der Zukunft’ under Grant 849028 (project PlanBISS).

References

- Andrienko, N., & Andrienko, G. (2011). Spatial generalization and aggregation of massive movement data. *IEEE Transactions on visualization and computer graphics*, 17(2), 205-219.
- Asamer, J., Reinthaler, M., Ruthmair, M., Straub, M., & Puchinger, J. (2016). Optimizing charging station locations for urban taxi providers. *Transportation Research Part A: Policy and Practice*, 85, 233-246.
- Aurenhammer, F., & Klein, R. (2000). Voronoi diagrams. *Handbook of computational geometry*, 5, 201-290.
- Bowyer, A. (1981). Computing dirichlet tessellations. *The Computer Journal*, 24(2), 162-166.
- Bradlow, E. T., Bronnenberg, B., Russell, G. J., Arora, N., Bell, D. R., Duvvuri, S. D., Hofstede, F. T., Sismeiro, C., Thomadsen, R., & Yang, S. (2005). Spatial models in marketing. *Marketing Letters*, 16(3-4), 267-278.
- Chew, L. P. (1989). Constrained delaunay triangulations. *Algorithmica*, 4(1-4), 97-108.
- ESRI (2016a). Fundamentals of TIN triangulation in ArcGIS. Retrieved from <http://desktop.arcgis.com/en/arcmap/10.3/manage-data/tin/fundamentals-of-tin-triangulation.htm>.
- ESRI (2016b). Cost Allocation. Retrieved from <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/cost-allocation.htm>.
- Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231).
- Fortune, S. (1987). A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2(1-4), 153-174.
- García-Palomares, J. C., Gutiérrez, J., & Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: a GIS approach. *Applied Geography*, 35(1), 235-246.
- Gold, C. (2016). Tessellations in GIS: Part I—putting it all together. *Geo-spatial Information Science*, 19(1), 9-25.
- Graser, A., & Olaya, V. (2015). Processing: A Python Framework for the Seamless Integration of Geoprocessing Tools in QGIS. *ISPRS International Journal of Geo-Information*, 4(4), 2219-2245.
- GRASS Development Team (2016). GRASS GIS Manual: r.cost. Retrieved from <https://grass.osgeo.org/grass70/manuals/r.cost.html#cost-allocation>.

- Krykewycz, G., Puchalsky, C., Rocks, J., Bonnette, B., & Jaskiewicz, F. (2010). Defining a primary market and estimating demand for major bicycle-sharing program in Philadelphia, Pennsylvania. *Transportation Research Record: Journal of the Transportation Research Board*, (2143), 117-124.
- O'Brien, O., Cheshire, J., & Batty, M. (2014). Mining bicycle sharing data for generating insights into sustainable transport systems. *Journal of Transport Geography*, 34, 262-273.
- Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2009) *Spatial tessellations: concepts and applications of Voronoi diagrams*. Vol. 501. John Wiley & Sons.
- Okabe, A., Satoh, T., Furuta, T., Suzuki, A., & Okano, K. (2008). Generalized network Voronoi diagrams: Concepts, computational methods, and applications. *International Journal of Geographical Information Science*, 22(9), 965-994.
- Owen, S.H., & Daskin, M.S. (1998). Strategic facility location: A review. *European Journal of Operational Research* 111, 423–447.
- Rudloff, C., & Lackner, B. (2014). Modeling demand for bikesharing systems: neighboring stations as source for demand and reason for structural breaks. *Transportation Research Record: Journal of the Transportation Research Board*, (2430), 1-11.
- Tournois, J., Alliez, P., & Devillers, O. (2010). 2D centroidal Voronoi tessellations with constraints. *Numerical Mathematics: Theory, Methods and Applications*, 3(2), 212-222.
- Watson, D. F. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The computer journal*, 24(2), 167-172.
- Weinstein Agrawal, A., Schlossberg, M., & Irvin, K. (2008). How far, by which route and why? A spatial analysis of pedestrian preference. *Journal of urban design*, 13(1), 81-98.
- Yang, B., Luan, X., & Li, Q. (2011). Generating hierarchical strokes from urban street networks based on spatial pattern recognition. *International Journal of Geographical Information Science*, 25(12), 2025-2050.