Building Footprint Aggregation with Preservation of Edge Orientations

Alexander Naumann[1,*], Samuel Bergé[2], Jonas Sauer[2], Jan-Henrik Haunert[1]

[1] Institute of Geodesy and Geoinformation, University of Bonn, Meckenheimer Allee 172, Bonn, 53115, Germany
[2] Institute of Computer Science, University of Bonn, Friedrich-Hirzebruch-Allee 8, Bonn, 53115, Germany

Keywords: building footprints, aggregation, map generalization, optimization, computational geometry

Abstract

The aggregation of building footprints is a key task of cartographic generalization, which is an important topic in geoinformation science. It has been approached from various angles, ranging from heuristics and optimization algorithms to machine learning. Given a set of input polygons that represent the building footprints, the task is to generate a set of polygons that provide a coarser representation of the input. The problem has applications in the visualization of settlement areas in small-scale maps, as well as settlement classification and analysis. A popular solution approach is to construct a subdivision of the plane and then build a solution by selecting faces from the subdivision. Often, a triangulation is used for the subdivision. However, this can cause the orientations of the boundary edges in the solution to differ drastically from the input polygons, which leads to a loss of information about the underlying settlement structure. We explore an alternative method that constructs the subdivision by extending the input building edges, thereby automatically preserving their orientations. To make the approach scalable to large instances without substantially decreasing the solution quality, we propose different methods of reducing the complexity of the subdivision. Our experimental evaluation on real-world data shows that our method is able to aggregate towns containing up to $\approx 10\,000$ building footprints while preserving input edge orientations much better than state-of-the-art methods.

1. Introduction

In cartography, generating smaller-scale maps from detailed geometric data is a core task. To keep visualizations legible for the human eye, objects need to be generalized. In this work, we focus on the processing of polygons, especially building footprints. Their generalization involves multiple operators such as simplification, selection, displacement, and, importantly, aggregation (Shea and McMaster, 1989). In small-scale maps, the buildings cannot be visualized individually. To keep the map readable, they must be aggregated into polygons that represent entire settlement areas. For the computation of these representative aggregated polygons, various approaches have been explored (Wang and Feng, 2024; Shen et al., 2019; Rottmann et al., 2025). However, many of the existing methods do not consider the preservation of edge orientations in the aggregation process. When displaying aggregated building footprints in a map, it is important that the aggregated shapes represent the orientations of the buildings near the drawn boundary. In this way, the viewer can gain an intuitive understanding of the underlying settlement structure.

In this work, we present a method for building footprint aggregation that respects edge orientations, based on the optimization-based approach by Rottmann et al. (2025). Given an input set of building footprints B and a subdivision F of the plane (Grünbaum, 1972), this approach constructs a solution S with $B \subseteq S$ by selecting faces from the subdivision. An optimization algorithm is used to find a solution that minimizes the objective value

$$g_\alpha(S) = A(S) + \alpha \cdot P(S), \tag{1}$$

where $A(S)$ is the area and $P(S)$ is the perimeter of S. The parameter α can be chosen arbitrarily in $\mathbb{R}_{\geq 0}$. Small values of α prioritize faithfulness to the input by heavily penalizing area, whereas larger values lead to a coarser aggregation in which more polygons are grouped together and the representative shapes are simplified. Given a fixed subdivision F, it has been shown that the aggregations are *nested* with respect to α: given an optimal solution S_1 for some α_1, there is always a solution S_2 for any $\alpha_2 > \alpha_1$ that fully includes S_1. Furthermore, the set of solutions over all $\alpha \in \mathbb{R}_{\geq 0}$ has linear size with respect to the number of input polygons and can be computed in polynomial time. This means that for every input polygon, it is possible to trace how, as α increases, it is merged with other (possibly already aggregated) polygons. Therefore, the solutions implicitly define a hierarchical clustering of the input polygons. As a consequence, the approach is applicable not only for settlement delineation, but also for urban analysis of settlement structures (Arribas-Bel et al., 2021; De Bellefon et al., 2021; Harig et al., 2021) and typification (Anders and Sester, 2000).

Rottmann et al. (2025) primarily used a Delaunay triangulation as the subdivision F, which does not preserve edge orientations. In a small experimental remark, they discussed an alternative method of constructing the subdivision by extending the building edges. This guarantees that every boundary edge of the solution follows the orientation of a building edge. In this work, we explore this idea further and engineer it to be applicable to real-world instances in a scalable manner. We identify two drawbacks of a naive extension approach: The complexity of the subdivision is quadratic in the input size, which limits the scalability. Furthermore, the boundary edges may represent the orientations of far-away buildings, rather than those close to the boundary, which leads to a poor representative quality. To address these issues, we introduce and evaluate methods for limiting the edge extension and simplifying groups of similar edges to simplify the line arrangement. See Figure 1 for an exemplary aggregation computed by our approach.

Our Contributions.

- We analyze the method for aggregating building footprints

* Corresponding author

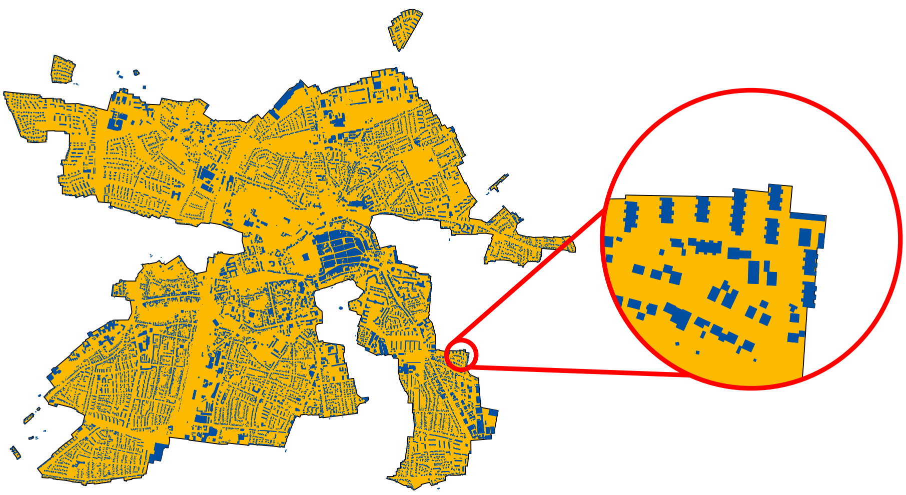


Figure 1. Building footprints of the town of Celle, Germany (blue) and an aggregation for $\alpha = 1\,000$ computed by our approach (yellow).

while preserving edge orientations proposed by Rottmann et al. (2025) and identify drawbacks regarding the representative quality of the solutions and the complexity of the produced subdivision.

- We present multiple strategies to address these drawbacks by limiting the extension of edges and simplifying the subdivision.
- We implement the approaches in C++ and provide an experimental comparison on real-world data. Our method is able to preserve the building orientations much better than existing approaches while being fast enough to handle instances with up to $\approx 10\,000$ buildings in 20 minutes.

2. Related Work

In this section, we first locate our work in the field of building footprint aggregation as an operator of cartographic map generalization. We then give an overview of prior work on the specific problem formulation of polygon aggregation that this work is based on, from the perspective of computational geometry. Lastly, we outline related work that places an emphasis on the preservation of edge orientations of buildings as an important and desirable property throughout simplification and aggregation processes.

Building Footprint Aggregation. Different fundamental approaches have been proposed for the aggregation of building footprints. These range from heuristics to optimization algorithms and to machine learning strategies. Among the heuristic approaches, Damen et al. (2008) proposed a sequence of morphological closing and opening operators for aggregating building footprints. Shen et al. (2019) instead used superpixel segmentation and selection, specifically fit to aggregate buildings given in the form of raster data. In the field of optimization algorithms, Rottmann et al. (2025) introduced the concept stating building footprint aggregation as an optimization problem with a mathematical objective at its core (see Equation 1). Within the field of machine learning techniques, most existing work primarily deals with the simplification of single buildings rather than the aggregation (Zhou et al., 2023). Recent approaches aim to simultaneously apply multiple generalization operators, such as simplification and aggregation. One can argue that this is also the case for the other approaches, as most aggregation techniques implicitly lead to a simpler boundary than the set of input shapes. This has been done via deep learning (Sester et al., 2018; Feng et al., 2019) and is being explored with graph convolutional networks as well as graph attention networks (Wang and Feng, 2024).

Polygon Aggregation. The framework in which the plane is triangulated and polygon aggregation becomes a selection problem on the set of triangles was first introduced by Jones et al. (1995). This idea was later experimentally evaluated (Li and Ai, 2010) and improved (Li et al., 2018). However, the original work did not define objectives that should guide the selection of triangles. This was later done by Rottmann et al. (2025), who introduced the objective function g_α (Equation 1) and also generalized the problem by allowing for any subdivision of the plane instead of a triangulation. They showed that an optimal solution according to this objective can be computed in polynomial time by constructing a weighted graph based on the triangulation and the input polygons and computing a minimum s-t-cut of it. Blank et al. (2025) studied generalized problem variants in which no subdivision of the plane is prescribed, but the objective is still to minimize g_α. For the most general variant, in which the solution may consist of arbitrary (not necessarily polygonal) regions, they gave an exact polynomial-time algorithm and proved that the boundaries of optimal solutions consist exclusively of pieces of input polygon edges and circular arcs of radius α. They also explored variants in which the regions must be polygons whose vertices lie on the input polygon edges, for which they gave polynomial-time approximation algorithms. Similarly, Funke and Storandt (2024) utilized the Delaunay triangulation to decompose the input into sub-instances, which they then aggregated individually using α-shapes.

Preserving Edge Orientations. The idea of considering the edge orientations in polygon processing has been utilized in various works, predominantly in the domain of polygon simplification: Kada (2007) used the dominant facade orientations in their framework for simplifying 3D building data. Buchin et al. (2011) introduced edge-moves for the simplification of polygons, where the manner in which edges are shifted and manipulated ensures that the edge orientation of the input is preserved. Goebbels and Rethmann (2024) proposed a mixed integer linear programming formulation for the simplification of polygons under consideration of the edge orientations and evaluated it on building rooftop data. Conceptually similar to our approach, Haunert and Wolff (2010) extended the edges of building footprints to compute optimal simplifications. Although we consider polygon aggregation rather than simplification, we note that aggregation includes an implicit simplification step because it can be seen as a combination of two subtasks: computing a clustering of the polygons and computing a simplified representative shape for each cluster.

3. Methodology

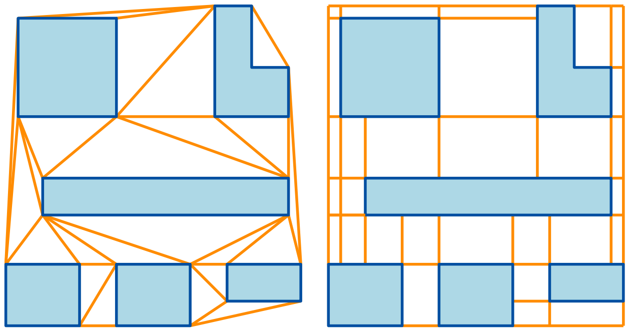
Our approach of aggregating polygons while respecting the edge orientations follows the method proposed by Rottmann et al. (2025): We are given an input set of polygons (i.e., building footprints) B and a subdivision of the plane. We denote by F the set of faces in the subdivision, i.e., the set of cells that form the subdivision. We require that each polygon in B is a face in F. A solution S to our problem is a set of faces in F that includes all input polygons, i.e., $B \subseteq S \subseteq F$. Let $\mathcal{S}$ denote the set of all possible solutions. We compute the solution $S \in \mathcal{S}$ that minimizes the objective $g_\alpha(S)$ stated in Equation 1. The parameter $\alpha \in \mathbb{R}_{\geq 0}$ allows us to control the penalty tradeoff between area and perimeter in our minimization problem. Intuitively, small choices of α place a high penalty on area. Because each included face increases the area, this makes it less likely that input polygons are grouped into a single polygon. The more we increase α, the more penalty we put on the perimeter instead. This can make it beneficial to add more faces to the solutions, provided that they reduce the total perimeter enough to outweigh the added area.

Once the subdivision F is given, one can construct a weighted graph based on the underlying line arrangement and retrieve an optimal solution for a given α by solving a minimum s-t-cut problem. We refer to the work of Rottmann et al. (2025) for further details regarding this step. In the following, we will treat it as given.

Our contribution focuses on the way the subdivision is built. In the original approach, the subdivision was a constrained (or conforming) Delaunay triangulation of all vertices of the input polygons B. Each edge in the triangulation connects two polygon vertices and is therefore adjacent to four polygon edges. If the two polygons are perfectly lined up, the triangulation edge has the same orientation as two of the adjacent edges. This can be seen in the lower left corner of Figure 2 (a), for example. However, in all other cases, the orientation of the triangulation edge will differ from those of the adjacent edges, which means that information about the building orientation is lost during the aggregation process. As originally proposed by Rottmann et al. (2025), we form the subdivision F by extending the edges of the input polygons B instead; see Figure 2 (b). This ensures that every boundary edge in the computed solution represents the orientation of some input polygon edge, although it may not necessarily be located close to the boundary. In the following, we propose different approaches for constructing F based on the concept of edge extension.

3.1 Edge Extension Strategies

Unlimited Extension. The most immediate strategy for building the subdivision based on edge extensions is to extend every edge of the input polygon set B until it either hits another polygon or the axis-aligned minimum bounding box of B. We refer



(a) triangulation (b) edge extension (ours)

Figure 2. A side-by-side comparison of the arrangement resulting from (a) a triangulation and (b) our edge extension approach. The input polygons B are drawn in blue, the added edges for the arrangement in orange.

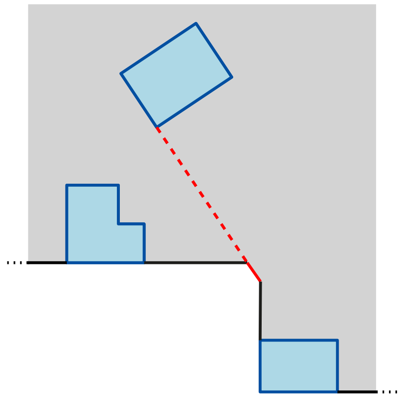


Figure 3. An example of how the unlimited edge extension might lead to edges in the solution (marked red) that stem from polygons not in the immediate vicinity. The interior of the aggregated solution polygon is drawn in gray.

to this strategy as *unlimited edge extension*. To see that the bounding box can be used as a limit, consider any solution that includes some area outside of it. Intersecting this solution with the bounding box reduces both the area and the perimeter, and thus improves the objective value $g_\alpha(S)$. This naive strategy has two drawbacks: Firstly, because the extended edges may be much longer than the building edges they originate from, they may appear on the boundary of a solution in places that are not in the vicinity of the original edge; see Figure 3. Since the goal is to represent the orientation of nearby building edges, this is undesirable. Second, the complexity of the resulting subdivision F may be too high: In a triangulation, the number of edges and faces is linear with respect to the number n of input vertices in B (i.e., its complexity is in $O(n)$). By contrast, in our unlimited edge extension strategy, every edge could intersect every other edge, leading to a quadratic complexity in the worst case, i.e., $O(n^2)$. This implies a higher running time for all arrangement operations, the graph construction, and the minimum cut computation. Therefore, we consider approaches with fewer potential edge intersections and a more locally constrained impact of an edge in the arrangement.

Limited Extension. Let E denote the set of all edges of the input polygons B. To reduce the number of edge intersections and thereby the complexity of our computed arrangement, we introduce a function $t\colon E \to \mathbb{R}$ that assigns to each edge e in E an *extension length* $t(e)$. Intuitively, the extension length

https://doi.org/10.5194/isprs-annals-XI-4-2026-153-2026

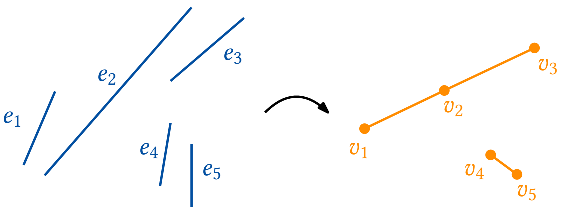


Figure 4. An exemplary configuration of edges of different length and orientation (drawn in blue) and a possible resulting similarity graph (drawn in orange). Corresponding edges e_i and vertices v_i have the same index i.

should scale with the length of the original edge e. This is because short building edges become visually indistinguishable at small scales, so it would be counterintuitive for them to substantially impact the aggregated boundary. On the other hand, if the extension is too short, then the extended edges of neighboring buildings do not intersect and no faces are formed in the subdivision. This is especially problematic in settlement areas with small but widely spaced buildings, which then become impossible to aggregate. We therefore propose to choose the extension length as

$$t(e) = c_{\text{ext}} \cdot \sqrt{\text{length}(e)}, \tag{2}$$

where the *extension factor* $c_{\text{ext}} \in \mathbb{R}_{\geq 0}$ can be chosen arbitrarily. This represents a good compromise by allowing shorter edges to be extended by a relatively higher proportion. After applying the limited extension to an edge, we discard the segment after the last intersection with another (extended) edge, as it does not form any new faces.

3.2 Edge Filtering

Although the limited extension methods already reduce the complexity of the arrangement, this might not suffice to make the algorithm scale well to large datasets. Especially in our use case of building footprints, neighboring input polygons are often similarly oriented, for example, if they are aligned along a street. Therefore, many extended edges represent very similar orientations. In smaller-scale maps, these edges are indistinguishable by the human eye. However, on a geometric level, they add many intersections and thereby tiny faces to the arrangement, making it more complex and the computations more expensive. We exploit the visual ambiguity of such sets by proposing an approach to group similar edges and then filter the edges within each group.

Grouping the Edges. Our goal is to group nearby edges with a similar orientation. This similarity needs to be quantified geometrically. We consider two edges *similar* if (1) the absolute difference in their orientation is below a threshold $\delta_\angle$ and (2) the distance between the two edges is below a threshold δ_{dist}. We define the distance between two edges as the minimum distance between any pair of points such that one point lies on each edge. Based on this definition, we define a similarity graph $G_{\text{sim}} = (V_{\text{sim}}, E_{\text{sim}})$, which has one vertex $v_e \in V_{\text{sim}}$ for each edge e in the arrangement. Two vertices are connected by an edge if and only if the corresponding edges in the arrangement are similar according to our definition; see Figure 4. On this similarity graph, we compute a *dominating set* (Allan and Laskar, 1978), i.e., a set of vertices $V_{\text{dom}} \subseteq V_{\text{sim}}$ such that every vertex is either in V_{dom} itself or has a neighbor in V_{dom}. We then form a group for each vertex in V_{dom} and assign each vertex that is not in V_{dom} to the group of one of its neighbors. This ensures that every group of edges in the arrangement has a “dominating” edge that is similar to all the other edges in the group. Finding a dominating set of minimum size is an NP-hard problem (Karp, 1972) and therefore unlikely to be solvable in polynomial time. We thus apply a greedy approximation algorithm; see Algorithm 1. We iteratively find the vertex $v_{\max}$ with the highest degree, form a new group out of $v_{\max}$ and its neighbors, and remove them from the graph.

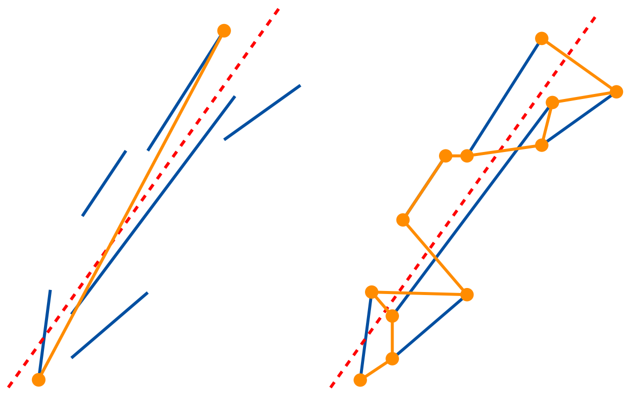
Algorithm 1 Greedy Similarity Grouping

Input: Similarity graph $G_{\text{sim}} = \{V_{\text{sim}}, E_{\text{sim}}\}$
Output: Partition P of V_{sim} into groups

$P \leftarrow \varnothing$
while $V_{\text{sim}} \neq \varnothing$ **do**
  $v_{\max} \leftarrow \text{argmax}_{v \in V_{\text{sim}}} \text{degree}(v)$
  $P \leftarrow P \cup v_{\max}$
  $V_{\text{sim}} \leftarrow V_{\text{sim}} \setminus \{v_{\max} \cup \text{neighbors}(v_{\max})\}$
end while
return P

This is known to yield an $O(\Delta)$-approximation, where Δ is the maximum degree of any vertex in the graph (Young, 2008).

Processing the Groups. To reduce the complexity of the arrangement without changing its structure too drastically, we replace the edges in each group with a single representative or a set of representatives. For this, we propose two strategies: *OuterConnect* and *EdgeRelink*. For both of these approaches, we first compute an *average direction line* for each group, as depicted in Figure 5. Let p denote the centroid of all edge endpoints in the group, and let o denote the weighted sum of the edge orientations in the group, where the weight of an edge is its length. Then the average direction line is the line with orientation o that goes through p. We then project all edge endpoints onto the average direction line and sort them according to the order in which the projected points appear on the line. OuterConnect replaces all edges in the group with a single new edge, which connects the first and last endpoint in this sequence. This leads to a substantial simplification of the arrangement. However, a side effect is that many input polygon vertices are no longer adjacent to any edges in the arrangement. Hence, it



(a) OuterConnect (b) EdgeRelink

Figure 5. The two methods we propose to process groups of similar edges. The edges are drawn in blue, the average direction line in dashed red, and the output of the processing in orange. Angles and spatial differences are exaggerated for legibility.

https://doi.org/10.5194/isprs-annals-XI-4-2026-153-2026 | © Author(s) 2026. CC BY 4.0 License.

may become impossible to generate solutions that separate certain buildings from each other because the arrangement may not contain any edges that separate them. The second method, EdgeRelink, attempts to circumvent this issue by connecting each consecutive pair of endpoints in the sequence with an edge, as shown in Figure 5 (b). In this way, all vertices are retained in the arrangement, at the price of a smaller reduction in the arrangement complexity compared to OuterConnect. In fact, the number of edges in the arrangement is not reduced, but edges from different groups are less likely to intersect each other. Naturally, both methods require reasonably small choices of $\delta_\angle$ and δ_{dist} to ensure that the arrangement is not changed too drastically.

3.3 Quantifying the Preservation of Edge Orientations

Given a solution S whose boundary edges are extensions of edges from our input polygon set B, we want to quantify how well it preserves the edge orientations of polygon edges that are close to the boundary. For this, we introduce a function $q\colon \mathcal{S} \to [0,1]$ that assigns an *edge preservation score* to each solution. Every edge e on the boundary of S is assigned an individual score $q(e) \in [0,1]$. Let $d_o(e)$ be the minimum distance of e to the polygon vertex from which it was extended. We normalize this distance with the diameter of the bounding box of B and subtract the result from 1. This yields

$$q(e) = 1 - \frac{d_o(e)}{\text{diameter\_bbox}}. \qquad (3)$$

If $q(e) = 1$, the edge e is directly incident to its originating building and therefore perfectly represents the orientation of the building in the solution. The further away the edge is from its originating building, the lower the score. Finally, we define $q(S)$ as the arithmetic mean of the scores of all edges on the boundary of S.

4. Experimental Evaluation

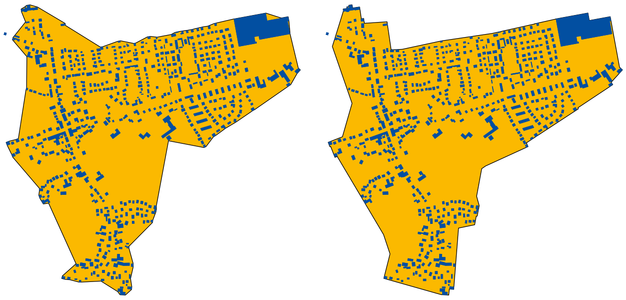
We implemented all variants of the algorithm proposed in Section 3 in C++. For geometric operations, we used the CGAL library (The CGAL Project, 2023). Graph computations utilized the Boost library (Boost-Contributors, 2024). All computations were performed on a machine using an AMD Ryzen Threadripper 3950X CPU and 128 GB of DDR4 RAM.

For the experiments, we used a set of real-world building footprint instances from 22 villages and 8 towns in Germany, which was also used by Rottmann et al. (2025). The data was taken from OpenStreetMap[1]. For the villages, the data sizes range from 337 polygons with 2 281 vertices to 1 722 polygons with 24 983 vertices. The towns vary from including 1 811 polygons with 17 187 vertices to 8 489 polygons with 84 704 vertices. The largest towns have populations of up to 60 000 inhabitants.

For all of our runs, we chose $\alpha \in \{0, 2.5, 5.0, 10.0, 20.0, 27.5, 40.0, 50.0, 100.0, 200.0, 1000.0, 4000.0, 100000.0, \infty\}$. These values proved to have a reasonable step size regarding the change in the solution shapes and groupings.

4.1 Comparison to Delaunay Triangulation

We begin by evaluating our basic approach with unlimited edge extension and without edge filtering, i.e., all edges are extended until they either hit a building or the bounding box of the input dataset. We compare the computed solutions to those given by the existing approach of Rottmann et al. (2025), which uses a Delaunay triangulation as the subdivision. Figure 6 shows a side-by-side visualization of the aggregations computed with both methods. Here we see that the aggregation computed by



(a) Delaunay triangulation (b) edge extension

Figure 6. A side-by-side comparison of the optimal aggregations (orange) of the village of Bokeloh (blue) with $\alpha = 200$ when using (a) the Delaunay triangulation and (b) the unlimited edge extension as an underlying arrangement for the minimization.

our approach better represents the orientation and structure of the village. Its overall edge preservation score $q(S)$ is 0.982, and $\approx 70\%$ of the boundary length is made up of edges with a score of 1. Additionally, it has a simpler boundary, making it computationally easier to visualize and easier to interpret for the human eye. For reference, Figure 7 shows the optimal solution for the problem variant studied by Blank et al. (2025), which drops all subdivision-based constraints. This solution achieves the smallest possible g_α value, but because this requires circular arcs, the building orientations are not represented at all.

We observe that in all three approaches, the computed regions represent the same sets of polygons. The boundaries follow roughly similar outlines, but the orientations are at times drastically different. In the upper area of the shown example, the buildings near the boundary are roughly but not perfectly aligned. Here, both the triangulation-based solution and the theoretically optimal solution produce an uneven boundary, whereas the boundary for the edge extension is much smoother. The differences are also notable in the parts of the boundary that are far away from any buildings, e.g., in the lower right area. Here, the theoretically optimal solution uses a large circular arc. The triangulation-based solution approximates this with the best available edges that connect far-away buildings, which in case leads to a prominent corner that does not reflect the outline of the village well. The edge extension approach approximates the cir-

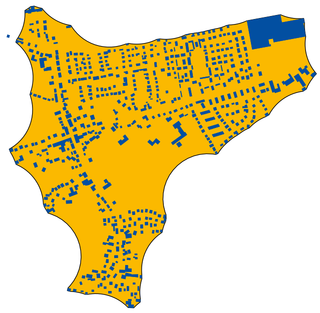


Figure 7. The theoretically optimal solution minimizing $g_{200}(S)$ for the village of Bokeloh, Germany, without any constraints regarding the output shape.

[1] https://www.openstreetmap.org

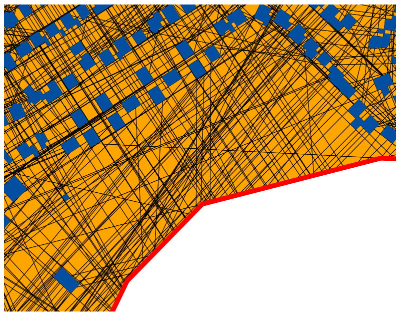


Figure 8. An excerpt of the aggregated solution (yellow) of the village of Altenrath (blue), Germany, with $\alpha = 200$. The edges of the outer boundary marked in red do not represent the edge orientations of any nearby buildings but result from extended edges of buildings located further away.

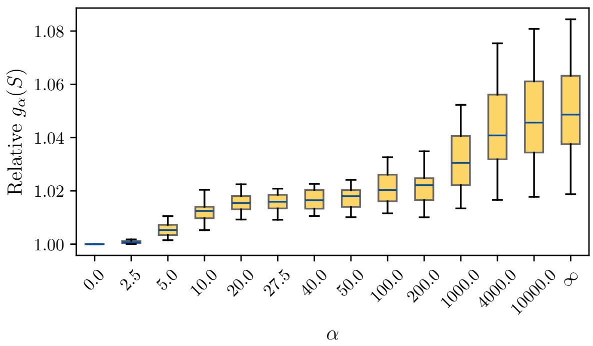


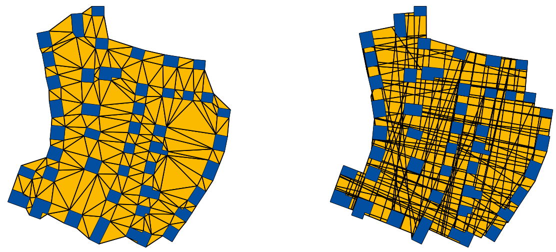
Figure 9. Relative deviation in the objective value $g_\alpha(S)$ of unlimited edge extension compared to Delaunay triangulation, averaged over all villages per α value.

cular arc with a sequence of extended edges that originate from several far-away buildings. These sequences, which do not represent the orientations of any nearby buildings, are undesirable artifacts of the unlimited extension strategy. A close-up view of such an artifact is shown in Figure 8.

For a quantitative comparison between the triangulation-based and edge extension approaches, we compared the average difference in the objective value $g_\alpha(S)$ over all α values in our experiments on the village datasets. Figure 9 shows that the mean difference is below 5% for all α. Hence, the unlimited edge extension approach does not deviate substantially in $g_\alpha(S)$, while improving the edge orientation information maintained throughout the aggregation. In terms of the running time, however, this approach is not yet feasible for the larger town datasets. This is due to the quadratic complexity of our arrangement; see Figure 10 for a visual example.

4.2 Limited Edge Extension

When choosing a limit for the edge extension lengths in the form of the extension factor c_{ext}, there are three metrics to consider: the arrangement complexity, the objective value $g_\alpha(S)$ and the edge preservation score $q(S)$. Strict limits simplify the arrangement and ensure that the boundary edges represent nearby buildings, but the price for this is a substantial loss of available aggregation options, which may cause the objective



(a) Delaunay triangulation (b) edge extension

Figure 10. The resulting arrangements when using (a) a Delaunay triangulation and (b) unlimited edge extension on the example of an excerpt of Gerolstein, Germany.

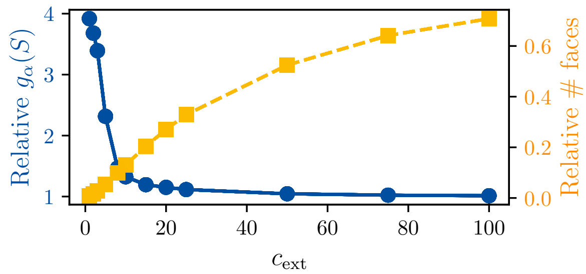


Figure 11. Comparison of the relative deviation in objective $g_\alpha(S)$ (blue) and the relative number of faces in the arrangement (orange) between the unlimited extension and limited extension with the function t. On the x-axis, the extension factor c_{ext} is varied. For each c_{ext} value, the g_α values are averaged over all villages and choices of α.

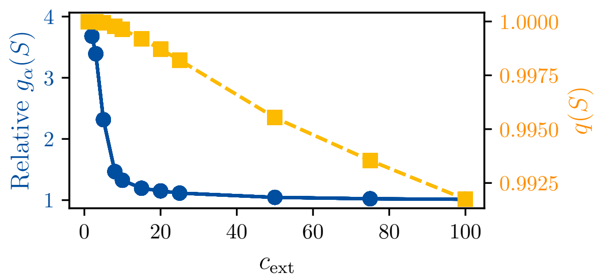


Figure 12. Edge preservation score $q(S)$ (orange) of the limited extension with the function t and relative deviation in the objective value $g_\alpha(S)$ (blue) compared to unlimited extension, averaged over all villages and choices of α.

value to degrade. Figure 11 shows that the loss in the objective value $g_\alpha(S)$ remains below 10% for $c_{\text{ext}} > 15$ but escalates quickly for lower values. The reduction of the number of faces in the arrangement is more gradual, reaching around 25% of the original size at $c_{\text{ext}} = 15$. Figure 12 shows that the loss in the edge preservation score is roughly linear in c_{ext}. Note that $c_{\text{ext}} = 0$ forces the solution to use input edges exclusively and thus achieves an optimal score $q(S) = 1.0$. Overall, we observe that values $c_{\text{ext}} \in [15, 30]$ strike a good balance between objective value, edge preservation score, and reduction of the arrangement complexity. Hence, we used values in this range for the following experiments. Figure 13 shows an exemplary comparison of the solutions with unlimited and limited edge extension. The red circle marks a spot where the unlimited solution uses a non-local edge, but the limited extension version

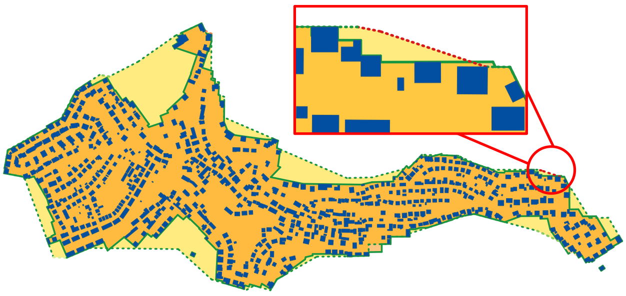
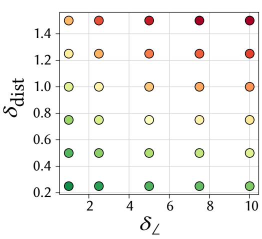


Figure 13. An overlay of the aggregations of the village of Erlenbach, Germany, with $\alpha = 200$. In yellow and with dashed outlines is the solution with unlimited edge extension, in orange and with solid outline the solution with limited extension length ($c_{\text{ext}} = 15$). The outlines are drawn in graduated style depending on $q(S)$ (< 0.5 in red, gradually more green towards 1.0).

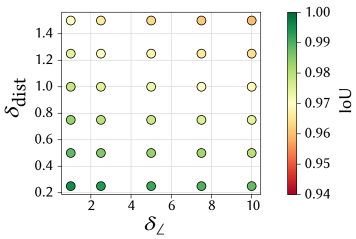
does not. Generally, we see that the limited extension leads to aggregated boundaries that better represent the structure of the polygons.

4.3 Edge Filtering

When evaluating the edge filtering methods, it is again important to find a tradeoff between the reduction in the arrangement complexity and the deviation in the computed shapes. Recall that OuterConnect replaces each group of similar edges with a single edge connecting the outermost points, whereas EdgeRelink uses a sequence of new edges that cover all vertices of the replaced ones. In both cases, the newly inserted edges cannot be clearly traced back to the original input building edges. Hence, we cannot apply the edge preservation score $q(S)$ here. Instead, we use the intersection over union (IoU) as an indicator of the difference in the solutions when applying the edge filtering approaches. Figure 14 shows that both filtering approaches achieve high IoU scores (> 0.97) for reasonably small choices of the parameters $\delta_\angle$ and δ_{dist}, with slightly better scores for EdgeRelink. Regarding the effectiveness in reducing the arrangement complexity (Figure 15), we observe that OuterConnect leads to fewer faces in the arrangement. This is natural, as it replaces entire groups with one edge, whereas EdgeRelink keeps the number of edges unchanged. In Figure 16, we show a visual comparison of the arrangements when using the different edge filtering techniques.

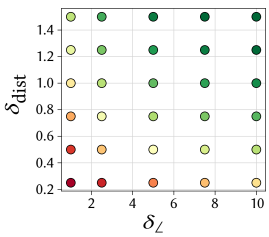


(a) OuterConnect

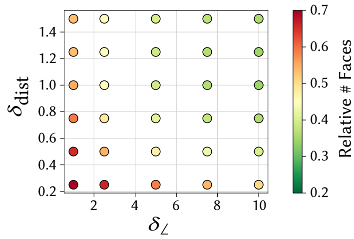


(b) EdgeRelink

Figure 14. The intersection over union (IoU) of the solutions computed with the edge filtering methods (a) EdgeRelink and (b) OuterConnect, compared to the solution without any filtering, averaged over all villages and choices of α and with $c_{\text{ext}} = 15$. The parameter $\delta_\angle$ is measured in degrees; δ_{dist} is multiplied by the average edge length in the respective input dataset.

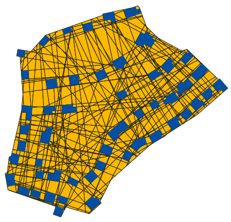


(a) OuterConnect

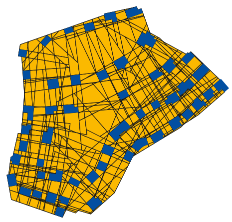


(b) EdgeRelink

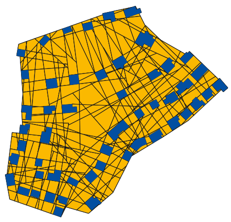
Figure 15. The relative number of faces in the resulting arrangements computed with the edge filtering methods (a) EdgeRelink and (b) OuterConnect, compared to the unfiltered arrangement, averaged over all villages and choices of α and with $c_{\text{ext}} = 15$.



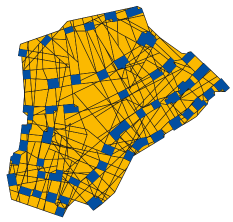
(a) unlimited extension



(b) limited extension



(c) limited + OuterConnect



(d) limited + EdgeRelink

Figure 16. The solutions and underlying arrangements with (a) unlimited extension, (b) limited extension ($c_{\text{ext}} = 15$), (c) additional OuterConnect filtering, and (d) additional EdgeRelink filtering on an excerpt of the village of Gerolstein, Germany.

Combining these findings, we conclude that both filtering methods are capable of reducing the arrangement while only leading to small changes in the output shapes. OuterConnect is more effective at simplifying the arrangement while leading to only slightly worse deviations from the unfiltered solution than EdgeRelink. Additionally, it leads to simpler shapes because it induces longer, consistent edges, making it the preferred strategy on our experimental data.

4.4 At Scale

After evaluating and fine-tuning the parameters of the limited edge extension and the edge filtering on the smaller village instances, we are now able to apply the algorithm to the instances of towns with up to $\approx 10\,000$ polygons. Runtime measurements are shown in Figure 17. The limitation of the edge extension and the edge filtering makes the algorithm scale substantially better for the larger instances, allowing us to aggregate over 8 000 polygons in roughly 20 minutes. For an exemplary result, see Figure 1, where we chose the parameters as described in Figure 17.

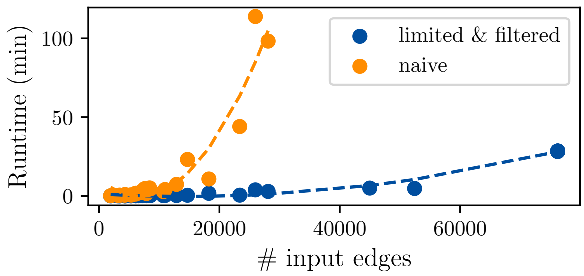


Figure 17. The running time per number of input edges over all instances (villages and towns). The unlimited edge extension method is shown in orange and the t-limited version ($c_{\text{ext}} = 27$) with OuterConnect edge filtering ($\delta_{\angle} = 5°, \delta_{\text{dist}} = 0.75 \cdot \text{avg\_input\_edge\_length}$) in blue.

4.5 Importance of Parameter Settings

Our techniques to improve the efficiency of the algorithm, while providing no theoretical guarantees, proved effective in our empirical evaluation on real-world data. We remark, however, that it remains important to strike a balance between the reduction of the arrangement size and thereby the running time on the one hand and the quality of the computed solutions on the other hand. Most importantly, overly restrictive limits to the edge extension and/or heavy edge filtering might rule out too many possible solutions by discarding edges that connect buildings to aggregated shapes, leading the algorithm to produce cluttered results. Moreover, the solution shapes produced by such extreme settings might not meaningfully represent the input orientations anymore. Therefore, it remains important to use the techniques cautiously and strike a balance between scalability and solution quality.

5. Conclusion and Future Work

We investigated strategies for preserving edge orientations in the optimization-based polygon aggregation method of Rottmann et al. (2025). The most basic approach extends the edges of the input polygons in an unlimited manner to build a subdivision of the plane from which the solution is selected. We observed that this preserves edge orientations much better than existing methods. In our use case of building footprints, this allows for better readability of small-scale maps and, additionally, a better understanding of the orientation of buildings in aggregated areas. However, we observed two drawbacks: The arrangement becomes quite complex, leading to high running times. Moreover, extending edges in an unlimited fashion can cause them to have a non-local impact on the solution boundary, which leads to unrepresentative shapes. Hence, we introduced methods to limit the extension of the edges. These substantially reduce the arrangement complexity, and thereby the running time, with only a marginal loss in the objective value. Additionally, we proposed edge grouping and filtering techniques to reduce the arrangement further by replacing very similar edges in the arrangement with fewer representative ones. The combination of both techniques enabled the application of the algorithm to town-sized datasets with up to $\approx 10\,000$ buildings in 20 minutes.

For future work, an extensive experimental evaluation of our techniques and their applicability to different city structures and layouts would complement this work in a meaningful way. Such an evaluation could also examine the robustness of the parameters of the techniques on varying input data. In a more general sense, it would be interesting to dive deeper into algorithm engineering techniques to speed up the computation further. One promising approach would be to decompose the instances into smaller subproblems, which can be solved in parallel. Furthermore, an alternative approach that might be promising is a postprocessing method, where one takes the solutions computed with the Delaunay-triangulated method (Rottmann et al., 2025) and modifies the boundary afterwards. This could potentially lead to outcomes of similar quality, but with less computational effort due to the linear complexity of the triangulation, in contrast to the potentially quadratic complexity of the edge extensions.

Acknowledgments

This work was supported by the German Research Foundation (DFG) under grant number 459420781.

References

Allan, R. B., Laskar, R., 1978. On domination and independent domination numbers of a graph. *Discrete Mathematics*, 23(2), 73-76.

Anders, K.-H., Sester, M., 2000. Parameter-free cluster detection in spatial databases and its application to typification. *Proceedings of the ISPRS Congress*, XXXIII, ISPRS, International Society for Photogrammetry and Remote Sensing (ISPRS), Amsterdam, The Netherlands, 75–74.

Arribas-Bel, D., Garcia-López, M.-A., Viladecans-Marsal, E., 2021. Building(s and) cities: Delineating urban areas with a machine learning algorithm. *Journal of Urban Economics*, 125, 103217. Delineation of Urban Areas.

Blank, L., Eppstein, D., Haunert, J.-H., Haverkort, H., Kolbe, B., Mayer, P., Mutzel, P., Naumann, A., Sauer, J., 2025. Bicriteria Polygon Aggregation with Arbitrary Shapes. *arXiv:2507.11212*.

Boost-Contributors, 2024. Boost C++ Libraries. Boost.org. Version 1.74.

Buchin, K., Meulemans, W., Speckmann, B., 2011. A new method for subdivision simplification with applications to urban-area generalization. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, Association for Computing Machinery, 261–270.

Damen, J., van Kreveld, M., Spaan, B., 2008. High quality building generalization by extending the morphological operators. *Proceedings of the 11th ICA Workshop on Generalisation and Multiple Representation*, 1–12.

De Bellefon, M.-P., Combes, P.-P., Duranton, G., Gobillon, L., Gorin, C., 2021. Delineating urban areas using building density. *Journal of Urban Economics*, 125, 103226.

Feng, Y., Thiemann, F., Sester, M., 2019. Learning Cartographic Building Generalization with Deep Convolutional Neural Networks. *ISPRS International Journal of Geo-Information*, 8(6).

Funke, S., Storandt, S., 2024. Smooth building footprint aggregation with alpha shapes. *Web and Wireless Geographical Information Systems: 21st International Symposium, W2GIS 2024, Yverdon-Les-Bains, Switzerland, June 17–18, 2024, Proceedings*, Springer-Verlag, Berlin, Heidelberg, 117–129.

Goebbels, S., Rethmann, J., 2024. Polyline simplification with predefined edge directions by mixed integer linear programs. *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2024, Volume 1: GRAPP, HUCAPP and IVAPP, Rome, Italy, February 27-29, 2024*, Scitepress, 203–210.

Grünbaum, B., 1972. *Arrangements and Spreads*. American Mathematical Society, Providence, RI. Published for the Conference Board of the Mathematical Sciences.

Harig, O., Hecht, R., Burghardt, D., Meinel, G., 2021. Automatic Delineation of Urban Growth Boundaries Based on Topographic Data Using Germany as a Case Study. *ISPRS International Journal of Geo-Information*, 10(5).

Haunert, J.-H., Wolff, A., 2010. Optimal and topologically safe simplification of building footprints. *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS ’10, Association for Computing Machinery, New York, NY, USA, 192–201.

Jones, C. B., Bundy, G. L., Ware, M. J., 1995. Map Generalization with a Triangulated Data Structure. *Cartography and Geographic Information Systems*, 22(4), 317–331.

Kada, M., 2007. Scale-dependent simplification of 3D building models based on cell decomposition and primitive instancing. *Proceedings of the 8th International Conference on Spatial Information Theory*, COSIT’07, Springer-Verlag, Berlin, Heidelberg, 222–237.

Karp, R. M., 1972. *Reducibility among Combinatorial Problems*. Springer US, Boston, MA, 85–103.

Li, C., Yin, Y., Liu, X., Wu, P., 2018. An Automated Processing Method for Agglomeration Areas. *ISPRS International Journal of Geo-Information*, 7(6).

Li, J., Ai, T., 2010. A triangulated spatial model for detection of spatial characteristics of GIS data. *2010 IEEE International Conference on Progress in Informatics and Computing*, 1, 155–159.

Rottmann, P., Driemel, A., Haverkort, H., Röglin, H., Haunert, J.-H., 2025. Bicriteria Shapes: Hierarchical Grouping and Aggregation of Polygons with an Efficient Graph-Cut Approach. *ACM Transactions on Spatial Algorithms and Systems*, 11(1).

Sester, M., Feng, Y., Thiemann, F., 2018. Building Generalization using Deep Learning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4, 565–572.

Shea, K. S., McMaster, R. B., 1989. Cartographic generalization in a digital environment: When and how to generalize. *Proceedings of the Auto-Carto IX: International Symposium on Computer-Assisted Cartography*, American Congress on Surveying and Mapping, Baltimore, MD, USA, 56–67.

Shen, Y., Ai, T., Li, W., Yang, M., Feng, Y., 2019. A polygon aggregation method with global feature preservation using superpixel segmentation. *Computers, Environment and Urban Systems*, 75, 117-131.

The CGAL Project, 2023. CGAL User and Reference Manual. 5.6 edn.

Wang, Y., Feng, Y., 2024. Simplification and aggregation of vector building footprints using deep learning models. *Abstracts of the ICA*, 7, 184.

Young, N. E., 2008. *Greedy Set-Cover Algorithms*. Springer US, Boston, MA, 379–381.

Zhou, Z., Fu, C., Weibel, R., 2023. Move and remove: Multitask learning for building simplification in vector maps with a graph convolutional neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*, 202, 205-218.