A Method for Topologically Consistent Polygon-to-Point Smooth Transitions in Vario-Scale Modelling

Amin Gholami[1], Pawel Boguslawski[1], Martijn Meijers[2]

[1]Wroclaw University of Environmental and Life Sciences, Institute of Geodesy and Geoinformatics, Grunwaldzka 53, 50-357 Wroclaw, Poland – amin.gholami@upwr.edu.pl, pawel.boguslawski@upwr.edu.pl.
[2]GIS Technology, Faculty of Architecture and the Built Environment, Delft University of Technology, Julianalaan 134, Delft, The Netherlands- b.m.meijers@tudelft.nl.

Keywords: vario-scale representation, straight skeleton, level of detail, weighted tree, LoD Transition Space

Abstract

Continuous vario-scale representations provide an effective framework for modelling geographic objects across multiple Levels of Detail (LoDs). In polygon-to-point transitions, straight-skeleton-based generalization may produce spatially disconnected polygon fragments when the resulting LoD Transition Space (LTS) is sliced at intermediate LoDs, leading to topological inconsistencies. This paper proposes a geometry-refinement method based on longest-path LoD normalization. The straight-skeleton network is transformed into a weighted tree in which transition vertices form the nodes and horizontal Euclidean distances define the edge weights. The longest root-to-leaf path is then used as a reference for reassigning the LoD coordinates of the transition vertices. The method is evaluated on three synthetic geometries, a real-world park polygon, and 50 building footprints extracted from OpenStreetMap in Stalowa Wola, Poland. For all evaluated post-normalization intermediate states, no disconnected polygon was observed. In the 50-building experiment, the connectivity rate increased from 93.6% to 100%, while execution time and memory consumption increased by 15.3% and 2.8%, respectively. Re-triangulation increased the numbers of edges and faces by 53.9% and 135.2%. These results provide empirical evidence that longest-path-based LoD normalization can substantially improve topological consistency, although a general connectivity guarantee and the geometric effects of LoD modification require further investigation. Although the pre-normalization connectivity rate varies with input geometry and LoD sampling, all evaluated post-normalization intermediate states remained connected.

1. Introduction

Representing geographic features consistently across multiple scales remains a central challenge in Geographic Information Systems (GIS) and cartography. Traditional multi-scale representations rely on discrete Levels of Detail (LoDs), where each scale is precomputed and stored independently (Li, 2006; Van Oosterom et al., 2014). Vario-scale representation has emerged as a powerful paradigm for modeling spatial data across continuous LoD, enabling smooth transitions and improved cross-scale analysis (Meijers, 2011; van Oosterom and Meijers, 2014). In the Space-Scale Cube (SSC), spatial objects are embedded in a 2D+1D structure where the third dimension represents scale (Meijers and van Oosterom, 2011).

Previous studies on generalization, where polygon-to-point and polygon-to-polyline collapsing have integrated straight skeletons, the Dual Half-Edge structure, and SSC for vario-scale representation (Gholami et al., 2025). The use of straight skeletons to model polygon shrinking through wavefront propagation was first justified for “roof model” in (Aichholzer et al., 1996), then the straight-skeleton “roof model” was reinterpreted as an LoD Transition Space (LTS) in (Gholami et al., 2025).

When the straight skeleton is used to drive area collapsing rather than to drive a continuous LoD transition space, the literature treats its per-vertex LoD value generically. (Gholami et al., 2025) integrate the straight skeleton with the Dual Half-Edge structure and the Space-Scale Cube to perform 2D-to-1D and 2D-to-0D generalization; in their framework all straight-skeleton vertices are assigned the same global LoD level, so that a single intermediate slice can collapse the polygon into a polyline (2D-to-1D) or directly into the target point (2D-to-0D). Haunert and Sester (2008) collapse polygonal areas by merging each straight-skeleton face with the neighbouring area that shares the originating edge, using a weighted straight skeleton only to adjust the size of each face; their topological constraints ensure that—a river joining a lake remains connected to the lake—but all skeleton vertices are treated on the same LoD footing implied by the underlying face-collapse sequence. Martinez et al. (2011) argue that, for the orthogonal polyhedra on which collapse algorithms frequently operate, the straight skeleton is preferable to the medial axis because it consists of planar faces rather than quadric surfaces and so enjoys strictly lower combinatorial complexity; Das et al. (2010) compute the straight skeleton itself in $O(n \log n)$ time and $O(n)$ space for monotone polygons, demonstrating that its vertex LoDs can be obtained efficiently but are still propagated uniformly across the wavefront; Furthermore, Eftekharian's extensive research regarding distance functions and skeletal representations emphasises that any skeletal reduction of a 2D or 3D domain encodes a single global distance function on which every skeleton vertex shares the same propagation rule (Eftekharian and Ilieş, 2009).

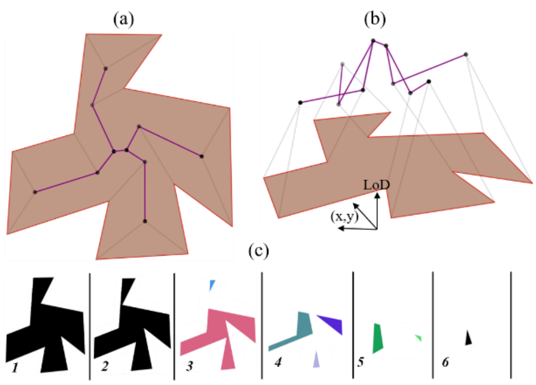

Figure 1. LTS construction for the polygon-to-point collapse: (a) the Straight Skeleton Network — skeleton line (purple lines) and skeleton vertices (black circles); (b) the resulting 3D LTS volume for the collapse; (c) six horizontal slicing surfaces.

In the LTS, the third dimension encodes the Level of Detail of each transition vertex rather than the original wavefront propagation time. The LTS thus represents the collapse of the source polygon into the target point (see Figure 1b).

For an LTS generated from the straight skeleton, the graph is equivalent to the straight skeleton network and forms the basis for constructing the proposed weighted tree (see Figure 1a). In the LoD Transition Space for polygon-to-point, the brown source face represents the original polygon at the highest level of detail, while the highest vertex, serves as the target point. Every face other than the source face is defined as a transition face, while every vertex that does not belong to the source face is defined as a transition vertex (or skeleton vertex), illustrated by the black dots, in Figure 1b. By treating each transition vertex as a node and the edges connecting adjacent transition vertices through transition faces as graph edges, a graph is constructed (shown in purple in Figure 1).

Intermediate LoDs in a vario-scale representation are achieved by slicing the 3D feature model. However, as shown in Figure 1c, the slicing process can sometimes generate multiple features from a single slice (Gholami et al., 2026), which, in certain applications, is inconsistent with topological integrity requirements. Following the slicing process, a single feature is depicted in black, while multiple features are shown in other colours. For instance, in Figure 1c, slices 3, 4, and 5 generate more than one feature. In this paper, the slicing process is applied to generate six intermediate LoDs between the source and target LoDs. This inconsistency occurs when an LTS has more than one valley in its 3D model; consequently, slicing can reveal disconnected components that contradict the intended single object representation. To mitigate these discontinuities, integrating the Dual Half-Edge data structure allows for a consistent topological relationship during scale transitions (Gholami et al., in press). We employ this structure to construct the 3D representation, ensuring that topological connections remain valid throughout the construction process.

We address multiple feature limitations by proposing a geometry refinement strategy that eliminates valleys in the 3D vario-scale representation. The method is based on analyzing the straight skeleton structure and reassigning the LoD values of its vertices using a normalization strategy derived from the longest path in a proposed weighted tree structure.
In a standard polygon-to-point LTS (see Figure 1), the vertical coordinate corresponds to a normalized LoD parameter derived from straight skeleton propagation. While the propagation is monotonic in time propogation, the resulting 3D geometry may contain local depressions due to varying edge configurations and event sequences (Aichholzer et al., 1996). Therefore, the problem can be defined as follows: given a polygon-to-point LTS, adjust the LoD embedding such that the resulting 3D surface is free of valleys and produces consistent single-polygon slices for all intermediate LoD values. However, the issue of multiple features is not limited to the LTS of polygon-to-point collapsing based on the straight skeleton; it can occur with other methods as well. Although this paper focuses on the LTS for polygon-to-point collapsing using the straight skeleton as a case study, the proposed approach can be extended to other models.
In the aforementioned approaches, straight-skeleton vertices inherit LoD information from the wavefront propagation or collapse process, which is globally monotonic in propagation time but does not necessarily provide a monotonic ordering along the resulting skeleton network. By contrast, this paper operates on the polygon-to-point LTS of Gholami et al. (2026) and assigns an LoD value to each transition vertex individually using the weighted-tree structure described in Sections 2.2–2.4. The longest root-to-leaf path provides the reference for normalizing the LoD coordinates, with the aim of reducing local valleys responsible for disconnected intermediate slices. Rather than assuming a general connectivity guarantee, the effectiveness of the proposed normalization is evaluated experimentally. The evaluation includes three synthetic geometries, a real-world irregular polygon, and a collection of 50 real building footprints, and considers slice connectivity, computational cost, memory consumption, and the increase in geometric complexity caused by re-triangulation.

2. Methodology

We present a strategy of eliminating valleys in straight-skeleton-based LoD transition spaces by manipulating the third dimension of transition vertices. This allows avoiding multi-polygon representation of a single feature in the process of smooth polygon-to-point transition.

2.1 Overview

Figure 2 summarizes the proposed LTS refinement workflow. Starting from a polygon-to-point LTS containing multiple valleys, the straight skeleton network is transformed into a weighted tree. The longest root-to-leaf path is then identified and used to define a normalization function. Finally, the LoD values of transition vertices are reassigned according to the normalized path lengths, yielding a valley-free LTS suitable for continuous vario-scale representation.

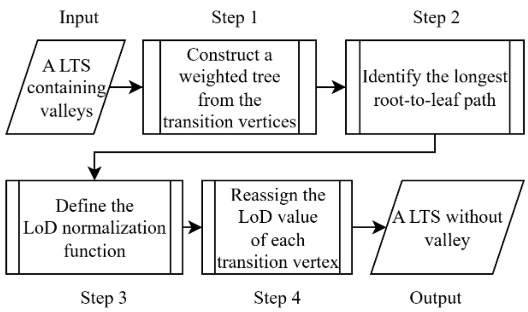

Figure 2. Overview of the proposed LTS refinement workflow.

2.2 Construction of the Weighted Tree

To identify the geometric dependencies within the LoD Transition Space (LTS), the skeleton vertices are transformed into a weighted tree (see Figure 3a, b). Each node of the tree represents a transition vertex, while each edge corresponds to a straight skeleton edge. The weight assigned to an edge is the horizontal (2D) Euclidean distance between its two incident skeleton vertices. The root of the tree is the target (peak) vertex of the polygon-to-point collapse, and the tree is recursively expanded by traversing all connected skeleton edges while avoiding previously visited vertices. The resulting weighted tree preserves the connectivity of the straight skeleton and provides the basis for computing path lengths used during LoD normalization. Algorithm 1 describes the construction procedure. It should be noted that the e.V and e.S navigational pointers used we refer to the start vertex and the symmetric half-edge of a half-edge (e) in the straight skeleton network, respectively.

The recursive construction initiates from the skeleton edge e_t, which contains the target vertex (vertex 1 in Figure 3a). To prevent loops in the tree, we maintain a list of visited vertices (VV); once a vertex is added to the tree, it is marked as visited, ensuring each transition vertex is represented exactly once as a node within the hierarchy. The recursive construction of the weighted tree begins with the call CreateWeightedTree (e_t, NULL, 0, VV, SV) and concludes once all skeleton vertices (SV) have been traversed and integrated, ensuring every transition vertex in the straight skeleton network is mapped as a node within the final tree structure. For example, in Figure 3, the skeleton vertices consist of vertices 1, 2, 3, ..., 10.

Algorithm 1. Construction of the weighted tree from the straight skeleton network.

```
Data structure TreeNode
   Attributes
      value: SkeletonVertex
      parent: TreeNode ∪ {NULL}
      children: List<TreeNode>
      length: Real
   Operations
      TreeNode(value, parent, length)
      AddChild(child: TreeNode)
Function CreateWeightedTree(e, parent, length, VV, SV)
   Input:
      e: current skeleton edge
      parent: parent tree node
      length: horizontal edge length
      VV: set of visited skeleton vertices
      SV: set of all skeleton vertices
   Output:
      node: root of the constructed subtree
v ← e.V            (1)
node ← TreeNode(v, parent, length)          (2)
VV ← VV ∪ {v}    ▷ mark current vertex BEFORE recursion (3)
if parent ≠ NULL then                (4)
parent.AddChild(node)                (5)
for each edge e' incident to v do    (6)
if e'.S.V ∉ VV and e'.S.V ∈ SV then          (7)
CreateWeightedTree(e', node, HorizontalLength(e'), VV, SV) (8)
return node        (9)
```

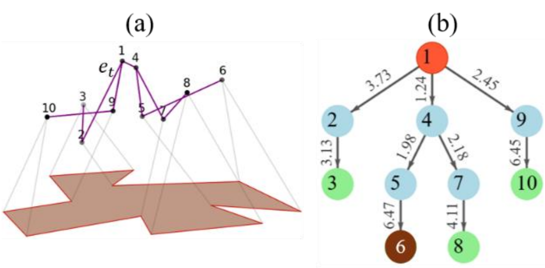

Figure 3. Step1 (a) LTS construction for the polygon-to-point collapse and straight skeleton network (purple lines and black vertices); (b) the weighted tree derived from the same 10 skeleton vertices; the longest leaf highlighted in brown.

2.3 Longest Path Identification

After the weighted tree has been constructed, the cumulative horizontal distance from the root to every leaf is computed. The root-to-leaf path with the maximum accumulated length is selected as the reference path. This path represents the longest geometric transition within the LTS and serves as the basis for subsequent normalization. The leaf at the end of this path is designated as the "longest leaf", which is highlighted in brown in Figure 3b.

2.4 LoD Normalization

The longest-path length is used to define a normalization function that maps the accumulated distance of every transition vertex to a normalized LoD value. Each transition vertex is processed independently, and its LoD value is updated according to its relative position along the weighted tree. Before normalization, we determine the slope (α) of the longest path:

$\alpha = \frac{H}{L}$, where H represents the total elevation change (differential LoD values) between root and longest leaf and L denotes the total cumulative horizontal distance of the longest path.
Consequently, the normalized LoD value for each vertex v is determined by multiplying its accumulated path distance d_v from the root by the slope α. The new LoD value for each node in the weighted tree is given by $node.value.LoD = root.value.LoD - d_v \cdot \alpha$.

The normalization assigns LoD values according to the accumulated horizontal distance from the root along the weighted tree. Consequently, vertices located farther along their respective paths receive proportionally lower LoD values, while the longest root-to-leaf path defines the reference range. This produces a monotonic LoD ordering along the selected reference path and reduces the local LoD depressions observed in the original LTS. The approach is therefore designed to eliminate the valleys responsible for the disconnected slices observed in the tested examples. However, this procedure does not, by itself, constitute a general theoretical proof that every possible polygon-to-point LTS will produce a single connected polygon at every intermediate LoD. The connectivity property is therefore evaluated experimentally in Section 3, and its general applicability remains subject to the assumptions and limitations discussed in Section 4. In the evaluated cases, the resulting monotonic LoD assignment produces a single connected polygon at each intermediate LoD, as demonstrated by the synthetic and real-world experiments in Section 3.

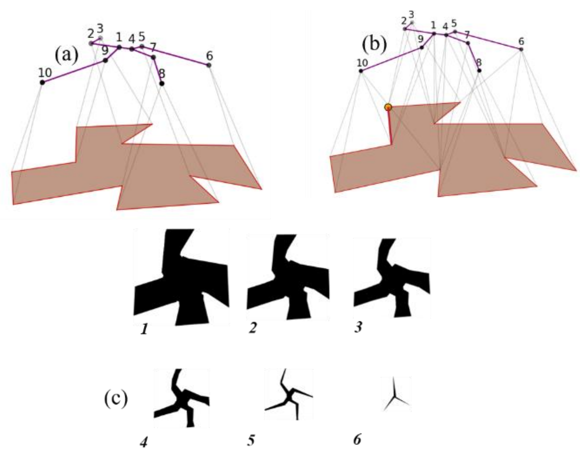

Figure 4. Geometry manipulation (a) After LoD normalization, (b) Re-triangulation of the affected transition faces using a constrained triangulation, (c) Slicing illustration, showing one feature for all slices.

The resulting LTS after LoD normalization is shown in Figure 4a. However, because the normalization process compromises the planarity of the transition faces, they must be re-triangulated. We employ a face triangulation approach, where each new edge must connect one vertex from the transition vertices and one from the source vertices (see Figure 4b). In the evaluated cases, re-triangulation of the normalized LTS enables the intermediate slices to remain single-feature polygonal representations at all six sampled LoD levels, as illustrated in Figure 4c. The broader applicability of this property is assessed empirically in Section 3 rather than assumed as a general theoretical guarantee.

3. Results

The proposed geometry refinement method is evaluated on three synthetic polygonal datasets and two real-world case studies, presented in that order.

3.1 Synthetic polygonal datasets

In this section, we evaluate our approach using three example polygons, as illustrated in Figure 5. The first synthetic dataset consists of a snail-shaped polygon. Figure 5a (top-right) displays the results prior to LoD normalization, where non-monotonic propagation rates create several disjointed fragments. In contrast, after applying the proposed LoD normalization, the slicing generates a coherent, single-part polygon, demonstrating the preservation of structural integrity during the multi-scale generalization process. Additionally, the proposed LoD transition improves the shrinking process, which proceeds from the longest-path node zone to the root zone.

The second synthetic dataset (see Figure 5b) represents two polygons connected by a narrow corridor, which before using the proposed LoD normalization leads to topological fragmentation of the polygon cluster during the shrinking process. Following the normalization process, the narrow corridor is preserved until the final stages of generalization, ensuring the cluster remains a singular, topologically consistent entity throughout the reduction. Finally, the third synthetic dataset (see Figure 5c) involves polygons with holes, where the normalized transition space successfully manages the collapse of interior rings without inducing premature geometry type changes.

The results demonstrate that the proposed LoD normalization facilitates smooth polygon shrinking around the transition vertices. When the shrunk polygon touches a transition vertex, the process triggers a topological change, effectively eliminating that vertex from the polygon. This procedure is applied iteratively to subsequent vertices until the original polygon structure collapses into a simplified point. The elimination sequence is governed by the weighted tree, proceeding from the longest leaf to the root node. For instance, in Figure 5b, before LoD normalization, the corridor collapses prematurely, causing a disjointed transition. In contrast, the normalized approach maintains the connection until the final generalization phase because the corridor's transition vertices have higher LoD values than those of the small polygon. Consequently, as illustrated in Figure 5b, during slices 2–4 after LoD normalization, the small polygon collapses first, followed by the corridor.

To complement the visual assessment, the proposed method was quantitatively evaluated using the three synthetic cases shown in Figure 5. For each case, six intermediate LoD slices were analysed before and after LoD normalization. The evaluation considered topological connectivity (Table 1), computational cost, memory consumption, and the geometric complexity of the resulting representation (Table 2).

A slice was considered topologically connected when it contained a single polygonal face. Before normalization, multiple polygonal faces were observed in all three test cases. For the snail polygon (Figure 5a), three of the six intermediate slices contained multiple faces. For the corridor case (Figure 5b), six of the six slices contained two faces, indicating premature separation of the connected polygonal configuration. For the polygon with holes (Figure 5c), four of the six slices contained multiple faces. After normalization, all 18 intermediate slices across the three synthetic cases contained a single face. Thus, the connectivity rate increased from 50.0%, 0%, and 33.3% before normalization to 100% after normalization for Figures 5a, 5b, and 5c, respectively. Overall, 18 out of 18 post-normalization slices remained single-face representations, corresponding to a 100% connectivity rate for the tested synthetic cases.

Case	**Connected slices before**	**Connectivity before**	**Connected slices after**	**Connectivity after**
Figure 5a	3/6	50.0%	6/6	100%
Figure 5b	0/6	0%	6/6	100%
Figure 5c	2/6	33.3%	6/6	100%
Overall	5/18	27.7%	18/18	100%

Table 1. Connectivity of intermediate slices before and after LoD normalization.

Using the presence of multiple polygonal faces within a slice as the operational definition of a connectivity-related topology violation, 13 of the 18 intermediate slices (72.2%) exhibited such violations before normalization, whereas none of the 18 slices exhibited this condition after normalization. This result provides quantitative evidence that the proposed normalization effectively removes the fragmentation observed in the tested transition spaces. For the polygon-with-holes case, the multiple-face counts should be interpreted together with the geometry shown in Figure 5c, since holes and their associated transition structures can produce complex face configurations.

The computational measurements indicate that the proposed normalization introduces only a modest increase in processing time and memory. The measured execution times and memory consumption are summarized in Table 2.

Case	**Time increase (%)**	**Memory increase (%)**	**Edge increase (%)**	**Face increase (%)**
Figure 5a	1.6	2.4	64.1	185.2
Figure 5b	7.0	1.33	56.4	146.7
Figure 5c	4.6	3.85	73.7	210.0

Table 2. Computational cost and geometric complexity before and after LoD normalization.

For Figure 5a, the execution time increased from 1.3274 s to 1.3490 s (1.6%), while for Figures 5b and 5c it increased from 0.3472 s to 0.3714 s (7.0%) and from 1.0077 s to 1.0542 s (4.6%), respectively. Across the three cases, the mean execution time increased from 0.8941 s before normalization to 0.9249 s after normalization, corresponding to an average increase of approximately 3.4%. Memory consumption increased by 1.02 MB, 0.52 MB, and 1.58 MB for Figures 5a, 5b, and 5c, respectively. The mean memory consumption increased from approximately 40.80 MB to 41.84 MB, corresponding to an average increase of approximately 1.04 MB. These results indicate that the proposed refinement has a relatively small computational overhead for the tested geometries. The main computational trade-off is the increase in geometric complexity resulting from re-triangulation, as summarized in Table 2.

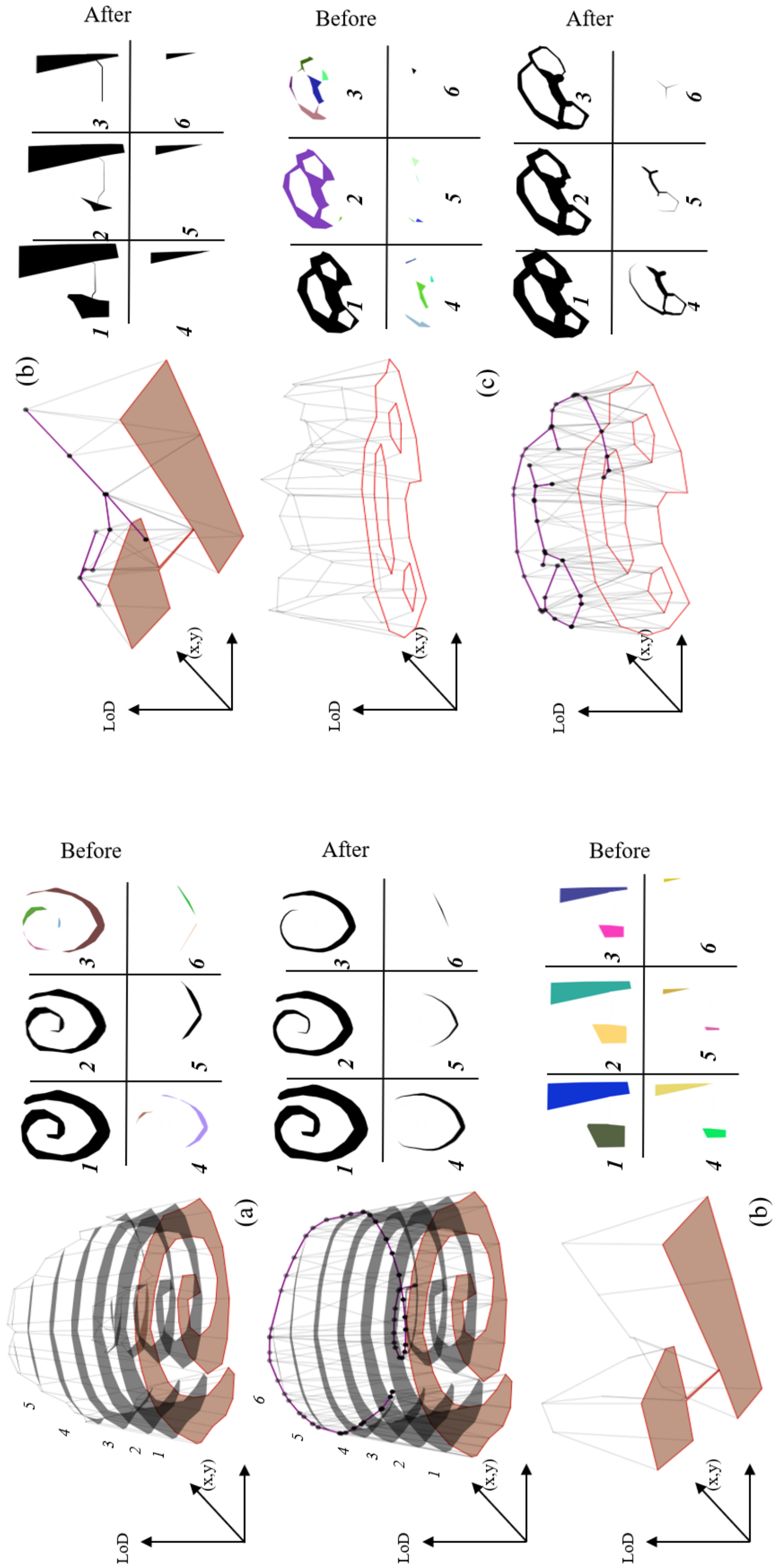

Figure 5. Before and after using proposed LoD normalization with their corresponding six intermediate LoDs by slicing of some mock data sets, using (a) a snail polygon, (b) two connected polygons with a corridor and (c) a polygon with three holes.

For Figure 5a, the number of edges increased from 156 to 256 (64.1%), while the number of faces increased from 54 to 154 (185.2%). For Figure 5b, the corresponding increases were from 39 to 61 edges (56.4%) and from 15 to 37 faces (146.7%). For Figure 5c, the number of edges increased from 114 to 198 (73.7%), while the number of faces increased from 40 to 124 (210.0%). This increase is expected because the affected transition faces are re-triangulated after modifying the LoD coordinates. Therefore, although the normalization substantially improves slice connectivity, it also increases the geometric complexity of the intermediate LTS representation.

The quantitative results demonstrate that the proposed normalization eliminated the multiple-face intermediate slices observed in the three synthetic examples, while introducing a relatively small increase in execution time and memory consumption. However, the increase in the number of edges and faces indicates that the scalability of the approach may become an important consideration for larger and more geometrically complex datasets. The present evaluation is therefore an initial quantitative validation rather than a comprehensive scalability assessment.

3.2 Real-World Evaluation on Single and Multiple Polygons

To extend the synthetic experiments in Section 3.1, the method was first evaluated on a real-world park polygon at 156 Wildwood Road, Kings Point, New York, USA, extracted from OpenStreetMap (Figure 6a). Its concave regions and branching protrusions provide an irregular geometry for evaluating the polygon-to-point LoD Transition Space (LTS).

Before normalization (Figure 6b), the straight-skeleton-based LTS contains local valleys and several of the six sampled intermediate LoDs produce disconnected polygonal components. After longest-path-based LoD normalization and re-triangulation (Figure 6c), all six sampled LoDs are represented by a single connected polygon. For this case, the result provides real-world evidence of the connectivity improvement observed in the synthetic experiments.

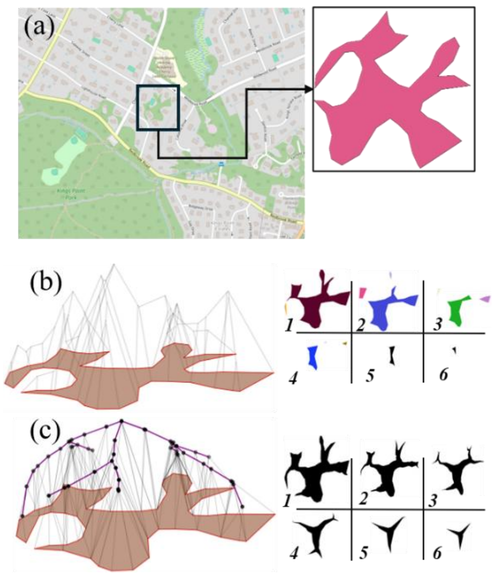

Figure 6. Real-world case study using a (a) park polygon [extracted from OpenStreetMap], (b) before and (c) after using the proposed LoD normalization with their corresponding intermediate LoDs between target and source by slicing.

A second experiment was conducted on 50 polygonal building footprints from a region of Stalowa Wola, Poland (see Figure 7a). The same polygon-to-point workflow was evaluated at six intermediate LoD levels. As the footprints progressively collapsed toward their target representations, the numbers of polygonal features present at Slices 1-6 were 38, 19, 12, 5, 3, and 1, respectively, giving 78 polygonal feature instances for connectivity assessment.

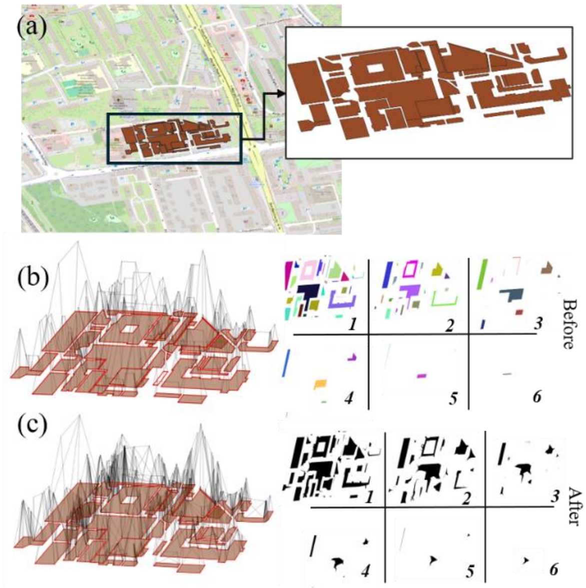

Figure 7. Expanded real-world evaluation on (a) 50 building footprints from Stalowa Wola, Poland [extracted from OpenStreetMap], (b) before and (c) after using the proposed LoD normalization with their corresponding intermediate LoDs between target and source by slicing.

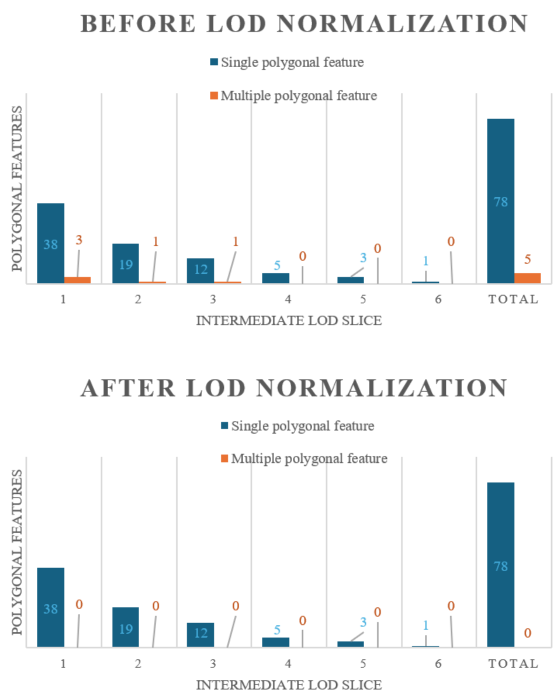

Figure 8. Connected and disconnected polygonal features at the six intermediate LoD slices before and after LoD normalization

Before normalization (Figures 7b and 8), disconnected features were observed at Slices 1-3 with counts of 3, 1, and 1, respectively; none occurred at Slices 4-6. Thus, 5 of 78 evaluated instances violated the single-connected-polygon criterion, giving a connectivity rate of 93.6%. After normalization (Figures 7c and 8), no disconnected feature was observed at any sampled LoD, increasing the connectivity rate to 100% for the 78 evaluated instances. No post-normalization connectivity failure was observed among the 50 input footprints at the sampled LoD levels.

For the same 50-footprint experiment, execution time increased from 8.9526 s to 10.3227 s (+15.3%), while memory use increased from 88.09 MB to 90.52 MB (+2.43 MB; +2.8%). The total number of edges in all cells increased from 1,533 to 2,359 (+53.9%) and the number of faces from 611 to 1,437 (+135.2%), reflecting the re-triangulation required after modifying the LoD coordinates. The connectivity improvement therefore has modest memory overhead but increases execution time and geometric complexity.

Additional sensitivity experiments indicate that the connectivity percentage before LoD normalization depends on both the number of sampled intermediate LoDs and the geometric characteristics of the input polygons. For example, when the real-world park case was evaluated using 20 intermediate slices, 14 of the 20 slices produced disconnected polygonal components, corresponding to a connectivity rate of 30.0% before normalization. For the Stalowa Wola dataset, evaluation using 20 intermediate LoDs resulted in 15 disconnected polygonal instances among 314 active polygonal instances, corresponding to a connectivity rate of 95.2%. The number of active polygonal instances decreases toward the target LoD because individual features complete their polygon-to-point transition at different stages. The substantial difference between the two pre-normalization connectivity rates is consistent with the different geometric characteristics of the datasets: the park polygon contains pronounced concavities and branching structures, whereas the building footprints are generally less irregular. Thus, the connectivity percentage before normalization should be interpreted as dataset- and sampling-dependent rather than as a fixed baseline performance measure. Importantly, after applying the proposed LoD normalization, no disconnected polygonal representation was observed in either of the 20-slice experiments, corresponding to 100% connectivity for all evaluated post-normalization states.

These results extend the empirical validation from isolated examples to a larger collection of real-world polygons. The additional 20-slice experiments further show that the absolute connectivity rate before normalization may vary with LoD sampling density and input geometry. Therefore, the principal observation is not a fixed pre-normalization connectivity percentage, but the consistent removal of the observed disconnected intermediate representations after normalization in all evaluated cases. Nevertheless, because the 50 footprints originate from one urban region, the experiment should be interpreted as an expanded robustness test rather than a comprehensive assessment across building typologies, geographic regions, or more complex hole and loop configurations.

Because the proposed normalization modifies the LoD coordinates of transition vertices, corresponding intermediate slices before and after normalization do not necessarily represent identical stages of the original straight-skeleton propagation. Consequently, the connectivity improvement reported above should not be interpreted as evidence that all geometric characteristics of the original propagation are preserved. The present evaluation focuses primarily on topological connectivity, computational cost, and representation complexity. A dedicated assessment using shape-fidelity measures such as relative area and perimeter change, compactness, centroid displacement, symmetric-difference area, or Hausdorff distance is required to quantify the geometric distortion introduced by the normalization.

4. Conclusion

This paper presents a geometry-refinement method for polygon-to-point Level of Detail Transition Spaces aimed at improving topological consistency during continuous vario-scale transitions. The method addresses local valleys in straight-skeleton-based LTS representations that may cause a horizontal intermediate-LoD slice to intersect the representation in multiple disconnected polygonal components. The proposed approach transforms the straight-skeleton network into a weighted tree, identifies the longest root-to-leaf path, and uses this path as a reference for normalizing the LoD coordinates of transition vertices. Because the resulting transition faces may no longer remain planar, the affected faces are subsequently re-triangulated.

The experimental results show a consistent improvement in connectivity for the evaluated geometries. Across the three synthetic cases, all 18 sampled intermediate slices were represented by a single polygon after normalization, compared with 5 of 18 before normalization in the current experiments. The synthetic experiments also showed that this improvement was obtained with a relatively small increase in execution time and memory consumption, although re-triangulation substantially increased the numbers of edges and faces.

The real-world experiments provide further empirical evidence of the method's robustness. For the irregular park polygon, all six sampled post-normalization LoDs remained connected. The expanded experiment using 50 building footprints from Stalowa Wola produced 78 polygonal intermediate states across the six sampled LoDs. Five disconnected states were observed before normalization and none after normalization, corresponding to an increase in connectivity from 93.6% to 100%. For this dataset, execution time increased from 8.9526 s to 10.3227 s (+15.3%), memory consumption increased from 88.09 MB to 90.52 MB (+2.8%), and the numbers of edges and faces increased by 53.9% and 135.2%, respectively. These measurements illustrate the principal trade-off of the method: improved topological connectivity at the cost of additional computational and geometric complexity.

The results should nevertheless be interpreted as empirical validation rather than a general proof of connectivity. The 50-building dataset originates from a single urban region, and the current implementation has not been comprehensively evaluated for geographically diverse building typologies or complex straight-skeleton configurations involving multiple holes and loops. In addition, modifying the LoD coordinates alters the progression of the original straight-skeleton propagation. Dedicated shape-distortion measures are therefore needed to determine how strongly the normalization affects cartographic and geometric fidelity. Comparison with alternative monotonic LoD-assignment and topology-preserving generalization methods would also provide a stronger assessment of the relative advantages of the proposed approach.

The additional sensitivity experiments indicate that pre-normalization connectivity varies with the sampled LoD positions and the geometric complexity of the input, whereas all evaluated cases reached 100% connectivity after normalization.

Future work will consequently focus on four directions. First, larger and more diverse real-world datasets should be evaluated to determine the robustness and scalability of the method across different polygon types and geographic contexts. Second, neighbouring polygons that share boundaries should be normalized jointly, since the present multi-building experiment processes individual polygon-to-point LTSs rather than enforcing coordinated behaviour between adjacent objects. Third, alternative monotonic mapping functions and optimization strategies should be investigated to reduce the geometric complexity introduced by re-triangulation while improving the geometric quality of intermediate representations. Finally, polygons with holes require further investigation because their straight-skeleton graphs may contain loops and multiple paths between vertices and the root; the current visited-vertex strategy resolves traversal ambiguity operationally but does not provide a general solution for simultaneous ring collapse.

Acknowledgments

This research work was funded in whole by the National Science Centre (NCN), Poland as part of the research program OPUS, project no.: 2021/41/B/ST10/03178.

References

Aichholzer, O., Aurenhammer, F., Alberts, D., Gärtner, B., 1996. A Novel Type of Skeleton for Polygons, in: Maurer, H., Calude, C., Salomaa, A. (Eds.), J.UCS The Journal of Universal Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 752–761. https://doi.org/10.1007/978-3-642-80350-5_65.

Das, G.K., Mukhopadhyay, A., Nandy, S.C., Patil, S., Rao, S.V., 2010. Computing the straight skeleton of a monotone polygon in O (n log n) time., in: CCCG. pp. 207–210.

Eftekharian, A.A., Ilieş, H.T., 2009. Distance functions and skeletal representations of rigid and non-rigid planar shapes. Comput.-Aided Des. 41, 865–876. https://doi.org/10.1016/j.cad.2009.05.006.

Gholami, A., Boguslawski, P., Meijers, M., 2026. Hierarchical Polygon-to-Point Collapsing for Multi-Scale Representation Based on the Straight Skeleton. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 11, 21–28. https://doi.org/10.5194/isprs-annals-xi-4-2026-21-2026.

Gholami, A., Boguslawski, P., Meijers, M., van Oosterom, P., 2025. 2D to 1D and 2D to 0D Geometry Generalization Based on the Straight Skeleton by the Dual-Half Edge Structure. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 48, 513–519. https://doi.org/10.5194/isprs-archives-xlviii-g-2025-513-2025.

Gholami, A., Boguslawski, P., Meijers, M., in press. Construction Method for Topological Integration of Multi-scale Datasets Supported by a Primal-Dual Data Structure. *Geo-spatial Information Science* .

Haunert, J.-H., Sester, M., 2008. Area Collapse and Road Centerlines based on Straight Skeletons. GeoInformatica 12, 169–191. https://doi.org/10.1007/s10707-007-0028-x

Li, Z., 2006. Algorithmic foundation of multi-scale spatial representation (1st ed.). CRC press. https://doi.org/10.1201/9781420008432.

Martinez, J., Vigo, M., Pla-Garcia, N., 2011. Skeleton computation of orthogonal polyhedra. Comput. Graph. Forum 30, 1573–1582. https://doi.org/10.1111/j.1467-8659.2011.02031.x

Meijers, B.M., 2011. Variable-scale Geo-information. NCG Delft. https://doi.org/10.54419/r2gjph.

Meijers, B.M., van Oosterom, P.J.M., 2011. THE SPACE-SCALE CUBE: AN INTEGRATED MODEL FOR 2D POLYGONAL AREAS AND SCALE. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-4-C21, 95–102. https://doi.org/10.5194/isprsarchives-XXXVIII-4-C21-95-2011

van Oosterom, P., Meijers, M., 2014. Vario-scale data structures supporting smooth zoom and progressive transfer of 2D and 3D data. Int. J. Geogr. Inf. Sci. 28, 455–478. https://doi.org/10.1080/13658816.2013.809724

Van Oosterom, P., Meijers, M., Stoter, J., Šuba, R., 2014. Data Structures for Continuous Generalisation: tGAP and SSC, in: Burghardt, D., Duchêne, C., Mackaness, W. (Eds.), Abstracting Geographic Information in a Data Rich World, Lecture Notes in Geoinformation and Cartography. Springer International Publishing, Cham, pp. 83–117. https://doi.org/10.1007/978-3-319-00203-3_4