Extracting topological features from room polygons based on a 2-dimensional space partitioning approach

Felix Gabler[1], Amir Ziaee[2], Wolfgang Huhnt[1], Georg Suter[2]
[1]Technische Universität Berlin, Germany
[2]Technische Universität Wien, Austria
f.gabler@tu-berlin.de

Abstract. This paper presents results from the ReconTOP research project focusing on a 2D space partitioning approach. It shows the space partitioning approach, the extraction of topological graph data and an evaluation process using given room polygons from 33 space layouts of apartment buildings. The objectives are threefold. First, we test the robustness of a 2D space partitioning algorithm. Second, we show how graphs describing the topology can be extracted from such a partition without requiring extensive geometric calculations. Third, we analyze the structure of space connectivity graphs and identify space properties that may be used as features in developing machine learning-based prediction models, e.g., for automated space classification.

Keywords: topology reconstruction, space partitioning, feature engineering

1. Introduction

The research presented in this paper starts with room polygons. Room polygons are either created by users in BIM authoring systems or may result from point cloud processing. They are used for various tasks in architecture, engineering, and construction. Examples are floor area measurement, evacuation path analysis, energy analysis, or space classification. However, room polygons often cannot be used directly to execute these tasks. Instead, they need to be processed beforehand. Accessibility graphs are, for instance, required in evacuation analysis. Similarly, neighboring relations are used in energy analysis. For some applications, a decomposition of rooms is advantageous or even necessary. All these graphs and decompositions are topological information.

The objective of the collaborative research ReconTOP conducted by the authors is to create a data processing pipeline that can classify room functions using room polyhedra derived from point clouds. The final step in the pipeline involves utilizing machine learning models to identify room functions for floor area measurement and evacuation path analysis. However, a significant challenge that needs to be addressed is the extraction of relevant features from room polygons to train the machine learning models. This paper focuses on extracting topological features from room polygons based on a space partitioning approach. The research is restricted to residential buildings. The boundary of all room volumes and polygons are assumed to be planar and linear, respectively.

2. Related Research

Space partitioning is an alternative approach compared to traditional geometric modeling. It is an existing approach (Mäntylä, 1988), but it has not been investigated in detail for applications in the AEC sector. Space partitioning is a decomposition of space into disjoint subsets. The union of all subsets is the entire space. It can be implemented in such a way that neighboring relations are stored explicitly so that navigation does not require extensive geometric

calculations. The topology is stored explicitly and can be extracted without geometric calculations.

The computation of the sign of a determinant is a crucial primitive operation in geometric algorithms for location tests. Although Yap (1994) states that rounding errors in arithmetic operations may generally not be problematic, the build-up of errors may be concerning. However, for topological statements of geometric solids, a round-off error may lead to false topological statements.

Romanschek et al. (2020) address this problem with the usage of integer coordinates. Their approach addresses the identification of space boundaries from point clouds and the topological evaluation. Gabler and Huhnt (2022) make use of this two-dimensional approach in a modified version for the identification of possible building elements from point cloud data. The modification is necessary to transfer the algorithm to the three-dimensional space as shown by Hu et al. (2020) and Vetter and Huhnt (2021).

In the indoor environment, route planning is a common use case. In respect of movable and immovable objects, space subdivision may be purposeful. Zlatanova et al. (2014) present an overview of 2D, 2.5D, and 3D approaches. Approaches are classified in no, partial or complete space subdivision to support path-finding algorithms. Lamarche and Donikian (2004) present a series of algorithms for the detection of bottlenecks, such as narrow passages, for path planning.

Other application areas making use of a space subdivision may be the identification of space utilization. For example, large, multi-functional residential spaces may need to be subdivided into multiple functional subspaces for living, eating, sleeping, or circulation for accurate floor area calculations (Suter, 2022). This is the proposed approach presented in this paper.

3. Input Data

For the 2-dimensional approach presented in this paper, a set of room polygons is expected as input. In this section general requirements for the input data are discussed. Second, the evaluated dataset is presented.

3.1 Requirements for the Input Data

The expected input data for our approach are room boundaries. The footprint of a room is modeled as a planar polygon. The boundary is a closed polygonal chain in the Euclidean plane. Besides the requirements of being closed and planar, polygonal chains must neither be self-intersecting nor self-touching. Polygonal chains oriented counter-clockwise are noted as positively oriented. The area left of the oriented edge is the corresponding face. Edges separate faces. However, faces may have holes. The polygonal chain of a hole-polygon is oriented clockwise and noted as negatively oriented. Therefore, a face with a hole consists of two polygonal chains.

3.2 Input Data from the SFS-A68 Data Set

We evaluated our approach using room polygons extracted from the SFS-A68 dataset, which was published by Ziaee and Suter (2022). This dataset contains space layouts for apartment buildings, ranging from one to more than five rooms per apartment. Rooms are modeled as BREP volumes. The footprint of each room is a closed, planar polygon. Each apartment consists of one or multiple polygons based on its number of rooms and the presence of holes.

The dataset also includes information about doors and openings, which are assigned to edges of polygons that represent building components such as walls. Figure 1 shows the input polygonal chains for a chosen example of a floor of a residential building designed by Kaden Klingbeil Architects. The example consists of 12 polygonal chains describing 11 spaces. Two polygons describe a shaft inside a room with different orientations, one positive describing the shaft, and one negative describing the hole in the room. Figure 1 shows the input polygons.

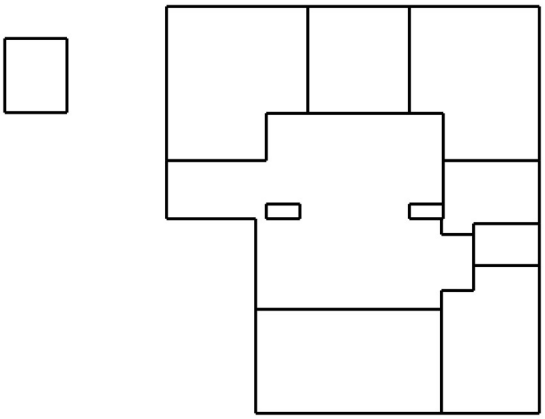

Figure 1: An example floorplan from the SFS-A68 dataset by Ziaee and Suter (2022). The top left isolated space is an elevator. All other ten spaces belong to an apartment unit.

4. Space Partitioning

In the first step, the boundaries of rooms are inserted in a space partition. The algorithm is based on the consideration to determine the location of points exactly. For this purpose, the coordinates of all points are mapped onto integer values. This part of the algorithm is equivalent to the approach presented by Romanschek et al. (2020).

2-simplices, triangles, are the basis for this space partitioning. An initial mesh of two triangles always forms the basis for partitioning. The two triangles share a common edge. The other four edges, two of each triangle, form a bounding box. All vertices of all polygons are known to be inside the two triangles or at the shared edge. No vertex can be at the edges of the bounding box.

The union of the two initial triangles represents the space of interest. The exterior of the space of interest is unbounded. The space of interest is decomposed into triangles. During the entire process, the following concept is applied: the space of interest is the union of all triangles, and triangles do not overlap.

For the insertion of a point in the mesh, exactly two conditions may apply: the point is inside a triangle or the point is on an edge. For the first case, the triangle will be split into three new triangles. The union of the three new triangles is the old triangle. For the other case, the two adjacent triangles of that edge will be split into four.

Topological relations between triangles are stored explicitly throughout the whole process. A triangle always has exactly three neighbors, one at each edge. Exceptions are only the four triangles at the boundary of the space of interest which represent the unbounded exterior.

After the insertion of all transformed vertices, the boundary edges of the polygons will be reconstructed. Contrary to Romanschek et al. (2020), coordinates of intersection points of boundary edges are not stored as rational numbers. The current algorithm makes use of local refinement. The coordinates of all vertices of the space of interest are stored as positive integer values of the data type int (32-bit). The refinement uses the class BigInteger as defined in (Java^{TM}).

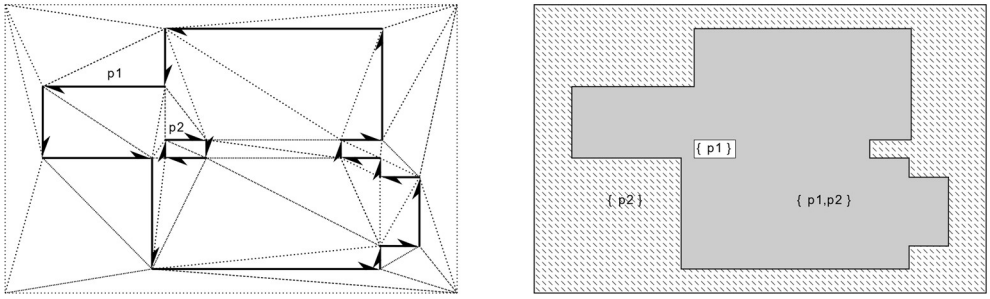

Figure 2: The left figure shows two input polygonal chains p1 and p2 and the resulting triangulation. P1 is oriented positively, p2 negatively. The right figure shows the set-theoretical evaluation and the configurations of this example: a room is assigned to p1 and p2, the exterior to p2, and the hole to p1.

After the reconstruction of all input polygons, a breadth-first search is performed to identify contiguous areas. A contiguous area, a 2D cell, is bounded by the edges of inserted polygons. The result of this step is a set of 2D cells. All 2D cells are disjoint, and the union of all 2D cells is the partitioned space of interest.

Besides the possibility of detecting overlapping input polygons a huge advantage of this approach is the detection of voids. For each 2D cell, the configuration of input polygons is stored. Figure 2 shows an example with a positive and a negative-oriented polygonal chain as input, three configurations are present:

1. Exterior: The 2D cell which touches the unbounded exterior is part of the exterior. All negative-oriented polygons are referenced in this 2D cell. All not referenced polygons are positive-oriented.

2. Face with a hole: The grey-shaded 2D cell from Figure 2 shows a configuration of two polygons. One of them, p2, has already been identified as a negative-oriented polygon. The resulting face with that configuration is a face with a hole.

3. Hole: A single polygon is referenced in the third configuration. The configuration does not refer to the negative polygon. Compare the white 2D cell in Figure 2.

Multiple further configurations exist. The interpretation of these configurations differs in models with positive and negative polygons. Models with only positive polygons are simple to interpret:

1. Exterior and void: no reference to a polygon
2. Room: a single reference to a polygon
3. Overlap: references to multiple polygons.

5. Output Data and its Computation

The output, different graphs describing the topology, is explained in this section. An adjacency and an accessibility graph are derived after an evaluation of contiguous areas in the space of interest, each represented by a node in the resulting graph. Each node is identified as a space of the building. Performing a convex decomposition splits non-convex spaces of the building into convex subspaces. A subspace-adjacency graph is derived.

5.1 Finding Contiguous Areas

After the insertion of all room polygons, closed 2D cells are found using a breadth-first search. Boundary edges of the inserted polygons are considered boundaries of an area. Self-intersecting polygons are not supported as input. However, polygons may intersect each other. The presented approach supports the detection of overlaps and additionally the identification of voids. For all identified 2D cells the configuration of the initial polygons is stored. In case an identified 2D cell is not linked to an input polygon, the 2D cell is identified and marked as void. Always present is the exterior, the unbounded space in which all polygons are inserted.

5.2 Space Adjacency and Space Access Graph

An adjacency graph is generated as the first result, storing the topological relations of the determined cells from the evaluation process. Besides the exterior, detected voids and overlaps, usually, each 2D cell represents a modeled space of the building. Each 2D cell is represented as a node in the adjacency graph. Two cells are adjacent if they share a common boundary edge. Adjacencies between cells are represented by undirected edges between corresponding nodes in the graph.

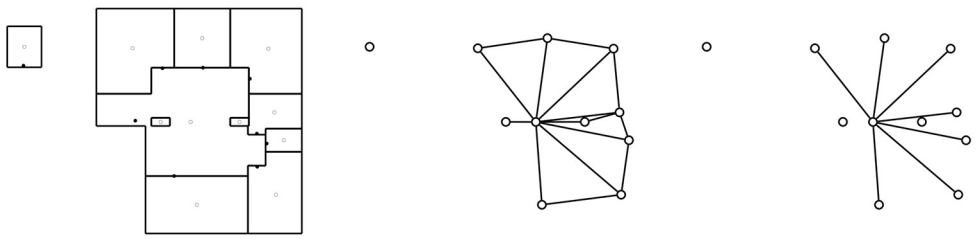

Figure 3: The example floorplan with positions of space labels (circles) and positions of doors and openings (filled circles) on the left. The corresponding space adjacency graph is in the middle, and the resulting space access graph is on the right.

Although the adjacency graph stores information about adjacent spaces of the building, no knowledge is contained on whether those are accessible. In the following step, the accessibility graph is generated.

Doors and openings are possibilities to allow accessibility of two adjacent spaces. In this 2-dimensional approach, only a single (mid)-point of a door or an opening is needed. The location at or close to an existing boundary edge is assumed.

The mid-point of each door and each opening is transformed and inserted into the space of interest. Four cases are distinguished:

1. The mid-point is located on a boundary point of two adjacent 2D cells: The adjacent 2D cells are connected.

2. The mid-point is located on a boundary point of more than two 2D cells: This case cannot occur because openings and doors are not located at corners where several spaces of a building meet.
3. The mid-point is located on a boundary edge of two adjacent 2D cells: The adjacent 2D cells are connected.
4. The mid-point is located in the interior of a 2D cell: The closest boundary edge is determined. The two 2D cells sharing this edge are connected.

The generation of the accessibility graph differs from the adjacency graph only in one detail: a common edge between two cells must contain a door or opening for the creation of a relation. Each cell is represented by a node. An undirected graph edge is created for each adjacent cell if the edge they have in common contains a door or an opening.

5.3 The Convex Decomposition

For non-convex 2D cells except the exterior, a convex decomposition algorithm is applied. The results are convex sub-cells. The algorithm, as described in Gabler and Huhnt (2022) traverses the boundary edges of a cell in a counter-clockwise direction. At each reflex vertex, the cell is split in two directions, in the positive direction of the previous edge and the negative direction of the next edge. Depending on the input data this algorithm may lead to a huge number of sub-cells. However, all sub-cells are ensured to be convex. The resulting decomposition is unique.

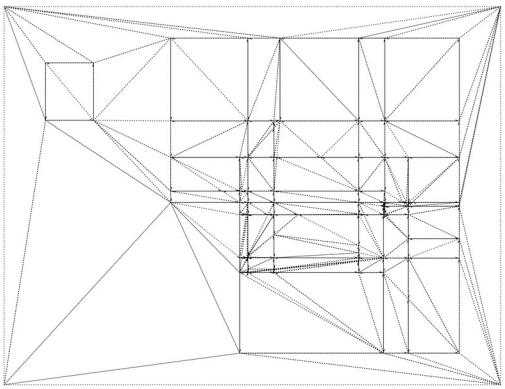

Figure 4: The mesh with the example floorplan after the convex decomposition, compare to Figure 1.

5.4 Decomposed Adjacency Graph

For the investigation of a room function, an adjacency graph of subspaces is derived. Rooms with a convex boundary will not be decomposed and are represented as a single node in the graph. Non-convex rooms are decomposed into multiple subspaces. Each subspace is a node in the corresponding graph. Adjacent subspaces are connected with edges.

Figure 5 shows the decomposed room from Figure 2 on the left. The derived subspace adjacency graph is shown on the right.

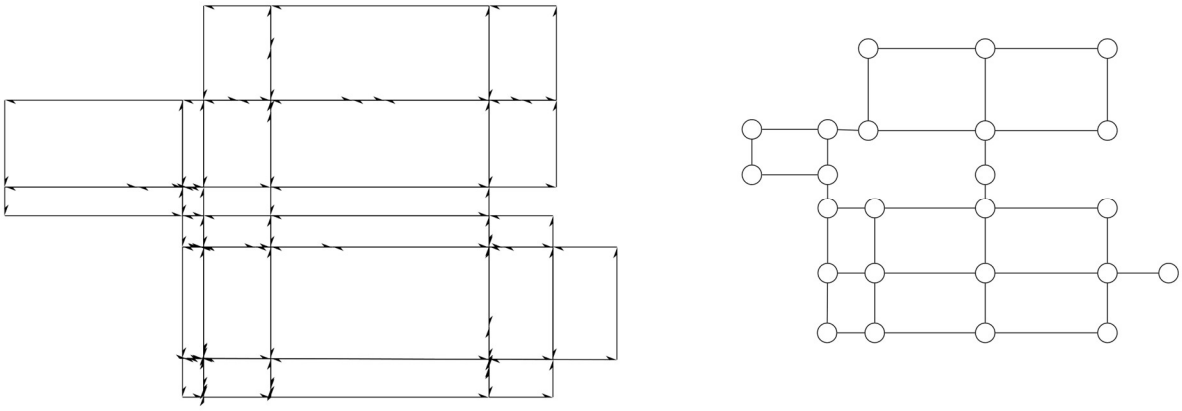

Figure 5: The two initial example polygons (see Figure 2) after the convex decomposition on the left. The corresponding adjacency graph for the subspaces on the right.

For the chosen example from the SFSA68 dataset, the resulting graph is shown in Figure 6. Of the 11 input spaces, six are convex and therefore not decomposed. They are isolated nodes in the corresponding graph. Three spaces have one reflex vertex each, leading to a decomposition into three subspaces each. Two non-convex spaces with multiple reflex vertices are contained. The convex decomposition leads to 7 and 24 subspaces for them.

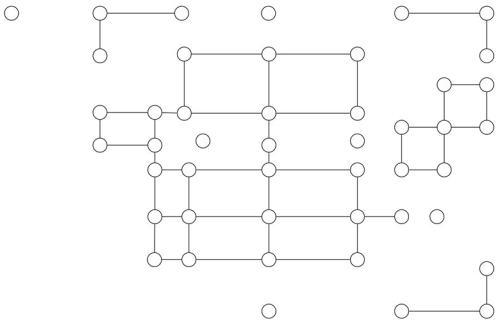

Figure 6: The resulting subspace adjacency graph for the presented example. The corresponding decomposition is shown in Figure 4.

5.5 Discussion

The presented space partitioning algorithm is capable of handling various types of models. It is not restricted to the Manhattan assumption. Exterior, rooms, voids, and any kind of overlap are identified based on the set-theoretical evaluation. By applying a convex decomposition, the algorithm ensures a unique partitioning and provides additional partitions for identified rooms. Adjacency and accessibility graphs can be efficiently extracted by comparing references.

The results demonstrate the effectiveness of space partitioning in this field, as subsequent evaluations do not rely on geometric algorithms, eliminating the possibility of round-off or truncation errors commonly associated with floating point numbers used in computers.

6. Data Analysis

We process space and subspace adjacency graphs with graph algorithms to extract graph-based space features. In follow-up work, we plan to use these features to develop space function

classifiers using machine learning methods. Graph-based features may be complementing existing space geometry and quantity features. Examples of the latter include floor area, space volume, or the number of doors, windows, or openings in a space (Bloch and Sacks, 2018; Buruz et al., 2022). We estimate the relevance of a graph-based feature by analyzing correlations between the space floor area property and the other graph-based features.

6.1 Measures

Node and graph measures that are used in the analysis are defined in Table 1. The measures are widely used in graph and network analysis (Newman 2010). They are computed using the NetworkX library (Hagberg, Schult, and Swart, 2011). Node measures include degree and selected centrality measures. Cluster measures are beyond the scope of this work. Similarly, a set of basic graph measures are selected.

Node measures are computed for each node in the space adjacency graph. A subspace adjacency graph typically consists of multiple components, where each component corresponds to a space. Thus, graph measures are computed separately for each such component.

Table 1. Graph measures (Hagberg, Schult, and Swart, 2011).

Measure	**Node-level measures**
Degree	Number of edges incident on a node.
Degree centrality	Fraction of nodes in a graph that are connected to a node.
Closeness	Reciprocal of the average shortest path distance from a node to all reachable nodes over the number of all reachable nodes.
Betweenness	Number of shortest paths between other node pairs that pass through a node.
Eigenvector	The centrality of a node depends on the centrality of its adjacent nodes.
Pagerank	The centrality of a node depends on the structure of incoming edges.
Measure	**Graph-level measures**
Cycle count	Number of basis cycles.
Diameter	Length of the shortest path between the most distanced nodes.
Avg. shortest path length	The average length of the shortest paths between all node pairs.

6.2 Results

Figure 8 shows squares of Pearson correlation coefficients (r^2) for space floor 'Area' property and graph measures in space adjacency and subspace adjacency graphs, respectively. The space count is higher in Figure 8 (a) than in Figure 8 (b) because the former includes loggias and access balconies.

(a) Space 'Area' property and node-level measures in the space adjacency graph (n=729 space nodes)

	Area	Degree	Degree centrality	Closeness	Betweenness	Eigenvector	Pagerank
Area	1.00	0.35	0.18	0.08	0.38	0.14	0.17
Degree	0.35	1.00	0.34	0.17	0.63	0.43	0.23
Degree centrality	0.18	0.34	1.00	0.69	0.50	0.77	0.95
Closeness	0.08	0.17	0.69	1.00	0.26	0.68	0.67
Betweenness	0.38	0.63	0.50	0.26	1.00	0.51	0.46
Eigenvector	0.14	0.43	0.77	0.68	0.51	1.00	0.71
Pagerank	0.17	0.23	0.95	0.67	0.46	0.71	1.00

(b) Space 'Area' property and graph-level measures in subspace adjacency graph components (n=689 space components)

	Area	Node count	Edge count	Cycle count	Diameter	Avg shortest path length
Area	1.00	0.43	0.39	0.28	0.50	0.49
Node count	0.43	1.00	0.99	0.86	0.85	0.89
Edge count	0.39	0.99	1.00	0.99	0.86	0.85
Cycle count	0.28	0.86	0.99	1.00	0.93	0.77
Diameter	0.50	0.85	0.86	0.93	1.00	0.55
Avg shortest path length	0.49	0.89	0.85	0.77	0.55	1.00

Figure 7: Squares of Pearson correlation coefficients (r2) between the space 'Area' property and graph measures in the (a) space adjacency graph and (b) subspace adjacency graph.

6.3 Discussion

Correlations between the 'Area' property and graph measures are generally weak for space adjacency as well as subspace adjacency graphs (max r^2=0.5). On the other hand, correlations between graph measures vary significantly in both graphs. For example, 'Closeness' and 'Betweenness' measures are weakly correlated (r^2=0.26) in the space adjacency graph. By contrast, the lowest correlation between graph measures exists between 'Cycle count' and 'Average path length' measures in the subspace adjacency graph, but it is rather strong (r^2=0.77). Alternatively, the correlation between 'Diameter' and 'Average shortest path' length is weaker (r^2=0.55) but the former correlates more strongly with 'Area' than 'Cycle count' (r^2=0.50 and r^2=0.28, respectively).

In summary, we conclude that graph measures generally appear to complement the 'Area' property well due to low correlation. More specifically, 'Closeness', 'Betweenness', and 'Cycle count' measures seem particularly well-suited for use as features in machine learning applications. Further studies are needed to assess if these observations also hold for other geometric space properties, such as the floor polygon perimeter property.

7. Conclusions and Outlook

The results show that the overall goal of our common research in establishing an automated chain from room polygons to functional space classification is reachable. The presented 2-dimensional approach focuses on floor plans modeled by users. Further investigations are required in almost every individual task in the addressed chain. We want to establish the chain starting with point clouds, computing polyhedrons from these point clouds, reconstructing the topology in 3D, extracting features from this topology, and using these features for space classification based on artificial intelligence.

The upcoming next step in the determination of topological relations is the transfer to the 3-dimensional space using a tetrahedral mesh. In a concurrent ongoing effort, we are including extracted node features from the space adjacency graph to develop machine learning-based space classification models for residential buildings. A challenge that needs to be addressed in future work concerns the improved exchange of geometry and topology data generated by the

space partitioning algorithm and client applications in the data processing pipeline, including space classifiers.

8. Acknowledgements

ReconTOP is a joint research project executed at TU Berlin, Germany, and TU Wien, Austria. The results presented in this paper have been worked out by a part of the ReconTOP team, funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 454008779 and by the Austrian Science Fund (FWF) - Project number I 5171-N. The authors thank Timo Hartmann and Abdullah Elsafty, TU Berlin, Germany, for the good cooperation in the ReconTOP research project.

References

Bloch, T. and Sacks, R. (2018). Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. Automation in Construction, 91, 256–272. DOI: https://doi.org/10.1016/j.autcon.2018.03.018

Buruzs, A., Šipetić, M., Blank-Landeshammer, B., & Zucker, G. (2022). IFC BIM Model Enrichment with Space Function Information Using Graph Neural Networks. Energies, 15(8), 2937. DOI: https//doi.org/10.3390/en15082937

Gabler, F. and Huhnt, W. (2022). Where is the end of the wall: decomposition of air and material into spaces and building components. In: Proceedings of the 19th International Conference on Computing in Civil and Building Engineering, ICCCBE 2022, Capetown, South Africa. [Accepted December 2022]

Java™ Platform, Standard Edition 7 API Specification, (2017). *BigInteger*. [online] Available at: https://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html [Accessed April 4, 2023]

Hagberg, A. A., D. A. Schult, and P. J. Swart (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. In: Proceedings of the 7th Python in Science Conference. Ed. by G. Varoquaux, T. Vaught, and J. Millman. Pasadena, CA USA, pp. 11–15.

Hu, Y., Schneider, T., Wang, B., Zorin, D. and Panozzo, D. (2020). Fast Tetrahedral Meshing in the Wild. ACM Transactions on Graphics, Vol. 39, No. 4. DOI: https://doi.org/10.1145/3386569.3392385

Lamarche, F., and Donikian, S. (2004). Crowd of Virtual Humans: a New Approach for Real Time Navigation in Complex and Structured Environments. Computer Graphics Forum, 23.

Romanschek, E., Clemen, C. and Huhnt, W. (2020). From Terrestrial Laser Scans to a Surface Model of a Building; Proof of Concept in 2D, In: 27th International Workshop on Intelligent Computing in Engineering, 2020, Online Workshop.

Mäntylä, M. (1988). An Introduction to Solid Modeling, Computer Science Press, 1988.

Newman, M. (2010). Networks: An Introduction. Oxford University Press. DOI: 10.1093/acprof:oso/9780199206650.001.0001.

Vetter, J., Huhnt, W. (2021). Accuracy Aspects when Transforming a Boundary Representation of Solids into a Tetrahedral Space Partition, In: 28th International Workshop on Intelligent Computing in Engineering, 2021, Berlin, Germany.

Yap, C. and Dubé, T. (1995). The Exact Computation Paradigm, In: Computing in Euclidean Geometry, pp. 452 – 492. DOI: https://doi.org/10.1142/9789812831699_0011

Ziaee, A. and Suter, G. (2022). SFS-A68: a dataset for the segmentation of space functions in apartment buildings. In: Proceedings of the 29th International Workshop on Intelligent Computing in Engineering, EG-ICE 2022, Aarhus, Denmark. DOI: https://doi.org/10.48550/arXiv.2209.09094

Zlatanova, S., Liu, L., Sithole, G., Mortari, F. (2014). Space subdivision for indoor applications. OTB Research Institute for the Built Environment, Delft University of Technology: Delft, The Netherlands, 2014. DOI: http://rgdoi.net/10.13140/2.1.2914.2081