

## QUANTIFYING THE QUALITY OF INDOOR MAPS

Moawiah Assali<sup>1</sup>, Georgios Pipelidis<sup>2\*</sup>, Vladimir Podolskiy<sup>1</sup>, Dorota Iwaszczuk<sup>3</sup>, Lukas Heinen<sup>4</sup>, Michael Gerndt<sup>1</sup>

<sup>1</sup> Chair of Computer Architecture and Parallel Systems, Dept. of Informatics, Technical University of Munich, Munich, Germany - muawiya.asali@gmail.com, v.podolskiy@tum.de, gerndt@in.tum.de

<sup>2</sup> Software and Systems Engineering Research Group, Dept. of Informatics, Technical University of Munich, Munich, Germany - pipelidi@in.tum.de

<sup>3</sup> Remote Sensing and Image Analysis Group, Dept. of Civil and Environmental Engineering Science, Technische Universität Darmstadt, Germany - iwaszczuk@geod.tu-darmstadt.de

<sup>4</sup> BMW Group IT, Munich, Germany - lukas.heinen@bmw.de

### Commission I, WG I/7

**KEY WORDS:** Quality Metrics, Indoor Map, Robot Map, Dynamic Quality Assessment

#### ABSTRACT:

Indoor maps are required for multiple applications, such as, navigation, building maintenance and robotics. One of common methods for map generation is laser scanning. In such maps, not only geometry of the map is of interest, but also its quality. This study aims at developing methods for real-time generation of indoor maps using features extracted from pointclouds obtained by a robot with their simultaneous quality assessment. We investigate, how this quality can be quantified for feature-based maps. First, we introduce a method for modeling 2D maps into 3D models that enable their usage for localization. Second, we review and evaluate a number of algorithms that can enable us to address features in a map. Hence, we enable the generation of objects from a pointcloud that has been sensed. Finally, we study several aspects of the map quality and we formalize them into metrics that can be applied to quantify their quality.

### 1. INTRODUCTION

Indoor maps are used in various applications, such as, navigation, building maintenance and robotics. One of the popular techniques to obtain indoor maps is laser scanning. In order to seamlessly sense larger indoor spaces including multiple rooms and corridors, mobile platforms are used. In such case, the indoor environment can be mapped using an approach called Simultaneous Localization And Mapping (SLAM) introduced by Durrant-Whyte et al. (1996). Pointclouds obtained in that way can be compared with already existing maps, which can be used for positioning or map update. For this purposes, not only the geometry of the existing an new map is needed, but also the quality of this map is of high interest, as it is one of the attributes which should be considered during the map usage as well as it should be considered during the matching and co-registration. Sester and Förstner (1989); Iwaszczuk and Stilla (2017).

Even though a lot of research focuses on map construction methods Karagiorgou and Pfoser (2012), there is a lack of methods that can enable the quantification of the quality of a robot generated map. Additionally, publications that focus on map evaluation (Frank and Ester, 2006; Podolskaya et al., 2009) do not provide a detailed analysis and comparison of criteria to be used for map quality assessment. Tran et al. (2019) focus on geometric comparison of a 3D indoor model with a reference. However, we need to consider the fact that maps may be used for a number of different applications and require different methods of comparison.

In order to determine the quality of a map, the quality of the 3D pointcloud can be calculated (Karam et al., 2018). 3D pointcloud, however, is an unorganized form of the 3D map and therefore

difficult for storage and useless in some analysis where more abstract or vector data as well as semantics are required. Therefore feature based representation and their quality are more suitable representation. In order to consider the quality of the map during the mapping process, both, mapping and quality assessment should be performed in real time. Frank and Ester (2006) propose a method to quantify the quality of a map after generalization takes place. They analyze many deformations that may result after generalization and propose an approach to specify map quality based on inaccuracies that can be detected. They provide a list of possible map operations that may result in increasing the uncertainty. According to them, those operations are: (1) *Elimination*, when all objects are not displayed, (2) *Exaggeration*, when objects are enlarged, (3) *Aggregation/Division*, when multiple objects are combined or larger objects are decomposed to smaller ones, (4) *Displacement*, when objects have been shifted, (5) *Reduction* when object have been eliminated, (6) *Typification*, when objects are described via typical patterns.

According to Pipino et al. (2002), quality metrics can be categorized into groups based on objective or subjective approaches. They call the metrics subjective, when they focus on the user needs and objective when they are task-dependent or independent. They apply this analysis on maps by assigning weights to each metric based on its importance for a specific user segment. Schwertfeger (2012) compares robot maps to ground truth maps. He makes use of map quality attributes such as: (a) Global Accuracy that is defined by the difference in distance between the actual map feature and its corresponding estimated distance, (b) Relative Accuracy that describes the amount of transformation (i.e. shifting, rotating, etc) that has been introduced to the generated map, (c) Local Consistencies that studies relative positions changes between objects, (d) Coverage, that describes the per-

\*Corresponding author

centage of features that have been mapped, (e) Resolution Quality that describes the number of map features that could be identified based on the originals.

Birk (2009) focuses on grid maps quality analysis. He introduces a metric that he calls ‘brokeness’, responsible to control the number of regions in a map as well as their spatial relations. He argues that a common error in map generation is rotated regions or particular regions that have been rotated due to technical problems such as slipping wheels in a robot. When applying cross entropy between cells at such situations will result in very low quality values despite the fact local contents of the wrong rotated room are correct. Brokeness is introduced to quantify such errors, based on a ground truth map. Zhang et al. (2011) propose a metric called Feature Similarity Index for Image Quality Assessment - FSIM for assessment of low level detail of an image, similar to what the human visual system perceives. Their approach consist of phase congruency, which specifies the local significance of a feature and image gradient magnitude. that involves three gradient operators, such as Sobel, Prewitt, and Sharr operators as for the estimation of the gradient magnitude. Cakmakov and Celakoska (2004) discuss shape similarity of detected objects in an image or a map. They distinguish two curve types, closed and free, where Closed are those that describe an objects, while free which are not closed, i.e. have no closing point. In such cases, a part of the curve should be imagined as to be able to compare with others. The authors define four perspectives possibilities for the matching process: (a) Transformational - where representations such as turning functions are used, (b) Geometrical - where polygon representations are used. (c) Structural - where graph matching techniques are used. (d) Quantitative - where closed curves are identified for shape description.

## 1.1 Contribution

In this study we aim at developing a set of methods for real-time generation of indoor maps using features extracted from pointclouds acquired by a robot with their simultaneous quality assessment. Then we investigate, how this quality can be quantified for feature-based maps. First, we introduce a method for modeling 2D maps into 3D models that enable their usage for localization. Second, we review and evaluate a number of algorithms that can enable us to address features in a map. Hence, we enable the generation of objects from a pointcloud that has been sensed. Finally, we study several aspects of the map quality, and review existing methods for quality quantification. Based on analysis of existing metrics, we identify a need to introduce a new quality metric - Characteristic Similarity, in order to quantify the rotation mismatch, which is not covered by existing metrics.

## 2. 3D MAP GENERATION

We propose an approach to generate a dataset for our investigation based on Open Street Maps. We use them, to generate both: the reference the 3D map and the LiDAR pointcloud. The required data is being generated in four steps, as depicted in Figure 1.

1. Blender model of the environment (OSM2World, n.d.) (Figure 1a).
2. Model introduced to the Gazebo simulation tool (Gazebo, n.d.) (Figure 1b).

3. Scanning the environment via LiDAR sensors (Figure 1c).
4. reasoning on the LiDAR sensed values and extraction of the point cloud (Figure 1d).

Using this approach, we can ensure that the generated 3D map and 3D LiDAR poincloud are in the same coordinate system.

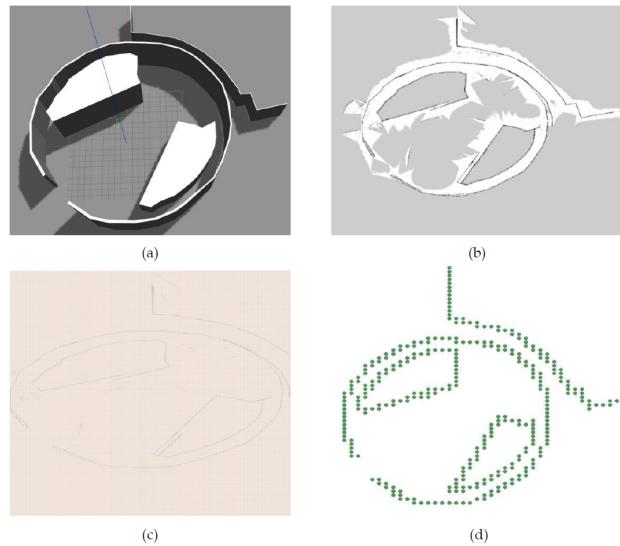


Figure 1. Pointcloud building: a) blender model of the environment; b) model in the Gazebo simulation tool; c) scanning the indoor environment via robot’s LiDAR sensors that Turtlebot3 robot is equipped with. The fourth step, in 1.(d), is to reason on the LiDAR sensed values and extract a point cloud

After the pointcloud is acquired, points are first clustered into lines, which are later clustered into polylines or polygons. Clustering points into objects is achieved via buffering, where each point’s boundaries are extended until they overlap with boundaries of another point or reach a maximum threshold. After points are structured into groups according to their nearest neighbors, a second level of clustering is applied that enables us to separate points that belong to different polygons .

Based on the clustering results, polygons are recognized. We devised three different strategies for polygon specification. Those approaches are based on a number of existing algorithms, such as the Concave Hull, Convex Hull, and Dijkstra Algorithm. We used and tested all the three algorithms and a detailed evaluation is available in the evaluation section. Examples of the concave hull and convex hull algorithms are presesnted in Figure 2.

### 2.1 Concave Hull

This algorithm aims to describe the region, which is occupied by a set of points. The algorithm operates based on a smoothness parameter that controls the corners of a polygon and their allowed irregularity. For example, a polygon may be assembled from points that their interior angles are less than or equal to 180 degrees. Hence, a concave polygon is the polygon that surrounds a set of points by the smallest possible area. An example of the concave hull algorithm is available in Figure 2a.

### 2.2 Convex Hull

The concave hull based algorithm operates based on the same principles with concave hull. Its main difference from Concave

Hull is that instead of the level of smoothness, it tunes the tightness of a polygon. As result, a polygon obtained through the concave approach tends to obtain tighter areas around the same set of points, when compared to a convex approach. An example of the convex hull algorithm is available in Figure 2b.

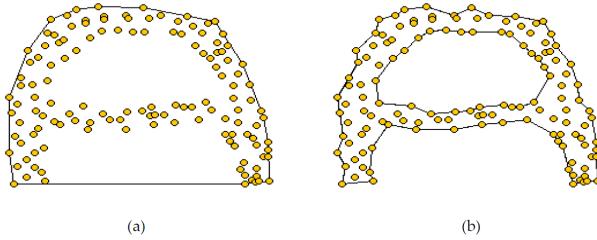


Figure 2. Example of polygons obtained via the concave hull (a) and convex hull (b). Franois Blair (n.d.).

### 2.3 Clustering Based on the Dijkstra Algorithm

Dijkstra is commonly used to calculate the shortest path between two points or locations. However, in our scenario, it is used to determine the sequence among the set of points, necessary for forming a polygon. More specifically, this step is executed following three different sub-steps. The first step is to generate the sequence of points, by marking each feature with a value that represents its location in reference to the previous nearest point. The second step is to generate lines between all points, based on the sequencing occurred above. The last step, is to fill enclosed areas via the connected lines, which eventually enables us to reveal the polygons.

## 3. QUALITY ASSESSMENT

In this section, we describe the set of quality metrics that we applied to the dynamically generated maps to quantify their quality.

### 3.1 Location Similarity - LS

This metric, introduced by Frank and Ester (2006), aims at quantifying the difference between locations of the same feature in the generated map, when compared against the reference map (displacement error). The metric makes use of Voronoi cells, aiming to reveal offsets between objects, by estimating the shifting that has occurred to the center of gravity between each object. The main idea behind Voronoi cells is to divide the map area into segments of equal surface, while the center of each cell remains to be the object's center.

The component's execution begins by applying the Voronoi algorithm on the map and obtain a list of neighbors for each feature. The next step is to calculate distances between each feature and its direct neighbors. Finally, it computes the differences of those distances as

$$LS_i(A, B) = 1 - \frac{[\sum_{m=1}^p |\frac{dist_A(i,m) - dist_B(i,m)}{\max(dist(i))}|]}{p}, \quad (1)$$

where,  $LS_i(A, B)$  is a value that describes the location similarity of the  $i^{th}$  object of map A and map B. The distance between two objects  $i$  and  $m$  of the map A and map B is described by  $dist_A(i, m)$  and  $dist_B(i, m)$  respectively. The maximum distance between two objects is expressed by  $\max(dist(i))$  and finally, the number of neighbors of an object is described by  $p$ .

The offset between those distances reflects the location similarity for each object, from which, it eventually computes a global Location Similarity value for the entire map as

$$LS(A, B) = 1 - \frac{[\sum_{m=1}^p LS_i(A, B)]}{n}, \quad (2)$$

where  $LS_i(A, B)$  is location similarity value for the  $i^{th}$  object of maps A and B and  $n$  describes the number of objects in the map.

### 3.2 Semantic Content Similarity - SCS

This metric, introduced by Frank and Ester (2006), aims to quantify neglected objects that have been either merged with others, have not being included or their only partial represented in the map. This metric operates by estimating the Voronoi entropy of the identified objects as

$$VE_i(A) = \sum([P_i * \ln(P_i) * \%V]), \quad (3)$$

where  $VE_i(A)$  indicates the entropy of the  $i^{th}$  object category in the map A,  $\%V$  is the percentage of Voronoi area of one object class with respect to the entire Voronoi area, and  $P$  is the number of objects of the category  $i$ . More specifically, this metric operates by first categorizing objects according to the their occupied area, usage and shape and then it estimates the entropy for each of the categories on the reference, as well as the generated map. Finally, it estimates the rate of change of Entropy between those two maps.

The final SCS score, is computed as

$$SCS(A, B) = \frac{\min[VE_i(A), VE_i(B)]}{\max[VE_i(A), VE_i(B)]} \quad (4)$$

where  $VE_i(A)$  and  $VE_i(B)$  describes the Voronoi entropy of the maps A and B respectively. This formula allows to obtain the ratio of entropy change. When reference and new maps obtain the same entropy values, then the two maps are identical.

### 3.3 Characteristic Similarity - CS

This metric is introduced by us and aims at quantifying how each polygons attributes preserved after generalization.

The most important step here is to design a suitable list of characteristics to be used for comparison:

- Prepare and compute an attribute list which supports the area and perimeter of a polygon.
- Compare each polygon with its corresponding one in the reference map .
- Provide each attribute with a specific weight reflecting its importance.

After obtaining a numerical list of attributes for each object in the map, the metric

$$CS(A, B) = \frac{\min[C_i(A), C_i(B)]}{\max[C_i(A), C_i(B)]}, \quad (5)$$

where  $C_i$  is characteristic of  $i^{th}$  element in maps A and B. The value is normalized to be between 0 and 1.

### 3.4 Shape Similarity - SS

This metric was introduced by Frank and Ester (2006). Its goal is to quantify the difference of the shape of the objects that belong to the maps. For achieving this, it makes use of the turning function. The turning function is a step-function that describes a shape by plotting its perimeter against its slope, steps on the function reflect rotations that a perimeter performs to construct the shape.

More specifically, this subcomponent is implemented by first estimating the turning function of each object using the Formula 6. Later, it subtracts the turning function of each object from each of object's turning function on the reference map, since the area between the two curves describes the difference between two turning functions. Finally, it normalizes this value based on the maximum difference.

$$SS_i(N_i) = 1 - \frac{Area(N_i)}{\max([Area(TF_i, A), Area(TF_i, A)])} \quad (6)$$

where  $TF_i, A$  is the turning function that describes the shape of the  $i^{th}$  object in the map  $A$ , while  $N_i$  is the area between the two turning functions.

### 3.5 Polygon Assessment - PA

Polygon Assessment, introduced by Podolskaya et al. (2009), is an aggregation of the Shape Similarity and Characteristic Similarity, with an additional metric, the Vertices of each object. It expresses the trade-off between the reduced amount of data and the generalized characteristics of the map.

This metric is calculated by first computing the turning function for each object in order to compute the shape similarity, as well as computing the area for each available object for assessing the characteristic similarity. Finally, we compute the number of vertices for each available object. The final numerical metric value can be then computed by

$$PA = W_{SS} * SS + W_{CS} * CS + W_V * V, \quad (7)$$

where  $SS$  is the shape similarity,  $CS$  is the characteristic similarity and  $V$  describes the vertex characteristics, while  $W_{SS}$ ,  $W_{CS}$  and  $W_V$  are weights for each corresponding metric.

## 4. EVALUATION

Two different shaped areas were selected to evaluate the developed framework. The data was acquired by the Turtlebot3 using the Gazebo simulation tool Gazebo (n.d.). The two scenarios were composed based on the variety of polygons in these areas:

**Scenario 1** The first scenario is placed at Technical University of Munich, which is marked red square in Figure 3. The room is a lecture hall and it consists of three polygons, which are composed by a collection of arc, circular and straight vertices, the two smaller polygons are overlapping with the larger polygon, which influences the Voronoi diagrams generation. Additionally, the place consists of narrow corridors and cantilever points, which could cause a drop to the LiDAR accuracy.

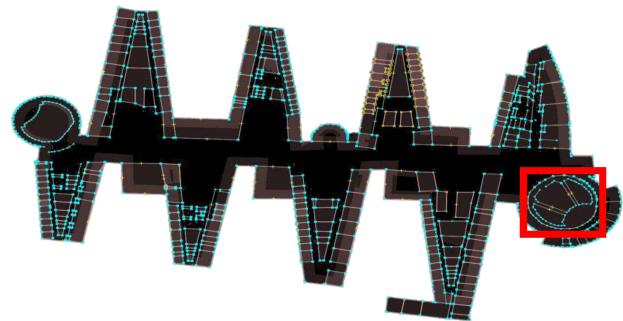


Figure 3. Test area

In Figure 4 it can be seen the extracted polygons. They were extracted via the concave hull algorithm, with tightness range between 0.36 and 0.51. Table 1 lists the obtained quality results for all polygons.

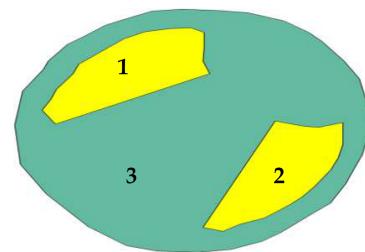


Figure 4. Polygons extracted in test area via the concave hull approach

Quality Metric	Polygon 1	Polygon 2	Polygon 3
LS	0.98	0.96	0.99
CS	0.95	0.95	0.89
SS	0.77	0.90	0.89
PA	0.79	0.86	0.77

Table 1. Calculated quality measures for each polygon calculated using Concave Hull approach (Scenario 1)

As can be seen in Table 2 the obtained quality scores are high, since the robot did not suffer from displacement shifts or errors. We can notice, that the Shape Similarity (SS) and Polygon Assessment (PA) values are low compared to other metrics. SS is highly dependable on vertices because it is computed based on a turning function. We could observe that large polygons containing large number of vertices achieve the lowest SS values. Also PA metric uses vertices as an integral part of its computation in conjunction with the Shape Similarity.

### 4.1 Scenario 2

The second scenario builds on the outdoor area from the Russell Road in London (Figure 5). This scenario was chosen due to the presence of inner and outer angles that follow L or T shape. Such shapes allow to evaluate the limits of the proposed algorithm when it is applied to the areas encompassing concave and convex hulls. Additionally, in this scenario objects have more direct neighbors. This is expected to influence the quality of the result due to the additional displacement of the objects. However, more neighbors imply more relations which increases the map processing time.

In Figure 6 can be seen the identified polygons, while in Table 3 the results for the second scenario using polygon specification

	Concave	Convex	Dijkstra
LS	0.98	0.98	0.97
SCS	0.99	0.98	0.98
CS	0.93	0.92	0.84
SS	0.78	0.60	0.29
PA	0.80	0.78	0.56

Table 2. Map quality calculated based on introduced quality measures (Scenario 1)



Figure 5. Scenario 2 - Customized Russel Road Buildings

with Concave Hull, Convex Hull, and Dijkstra Algorithm are presented. The quality evaluation provides a better picture of how the polygons affect the quality of the extracted map, using various metrics Table 4.

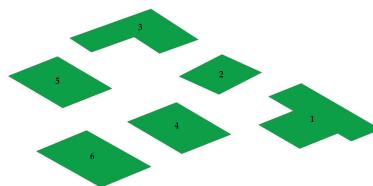


Figure 6. Polygons extracted in test area via the concave hull approach.

## 5. DISCUSSION

The collected results enabled us to notice that the following properties of both explained scenarios affect specific quality metrics:

**Arc Trajectories:** One may notice that the robot did not manage to capture exact vertices for this type of trajectory. In turn, this directly affected the shape similarity and the polygon assessment quality metrics. The first metric is affected as a result of cumulative angles being represented in the corresponding turning function. The second one is affected because the shape similarity is one of its components and it already considers the number of Vertices in its computations. Concave and Convex Hull polygons were less affected by this property than Dijkstra because of Dijkstra's ability to capture sharp segments that contribute to a higher deformation of the shape. The tightness parameter reflects how far the difference is between Concave and Convex Hulls approaches. Increasing the tightness factor (lower smoothness val-

	Concave	Convex	Dijkstra
LS	0.99	0.99	0.99
SCS	0.99	0.97	0.99
CS	0.97	0.94	0.87
SS	0.37	0.59	0.03
PA	0.62	0.70	0.45

Table 3. Scenario 2 - Final Results

Qual. Metr.	Pol 1	Pol 2	Pol 3	Pol 4	Pol 5	Pol 6
LS	0.99	0.99	0.99	0.99	0.99	0.98
CS	0.98	0.97	0.98	0.99	0.98	0.93
SS	0.13	0.53	0.24	0.80	0.21	0.261
PA	0.54	0.7	0.60	0.76	0.59	0.58

Table 4. Calculated quality measures for each polygon calculated using Concave Hull approach (Scenario 2)

ues for the algorithm) may result in a situation where line segments start breaking around local cantilever points creating more additional vertices deforming the shape which in turn deforms the quality of the shape. This observation implies that a smoothness value should be chosen to avoid two extremes of having many broken segments producing new unnecessary vertices and cutting inner angles in a way that deforms shapes.

### 5.1 Straight Trajectories:

This property is similar to the previous one in its effect upon quality metrics. The main observation is that when a relatively long straight trajectory is considered, a robot may add additional vertices that are not displayed on a corresponding reference map. Another observation is that when a robot passes the same location more than once, new vertices may be introduced, each with a small added shift in the corresponding turning function. Shape Similarity and Polygon Assessment metrics are mostly affected by this type of trajectory.

**Overlapping Polygons:** These polygons increase the difficulty to capture boundaries between objects. They also may change how a relative location for one object is computed as a Location Similarity metric may take other forms to represent an object (other than a center of gravity). Location Similarity is the most affected by this feature of the map (Table 1, low LS value for polygon 3).

### 5.2 Narrow Corridors

Presence of a narrow passage on a map may force two independent polygons to merge. This will result in a decreased object count and relative importance after scan. Such a map feature affects Semantic Content Similarity metric as it takes into account how objects are categorized before and after a scan. Considered examples of narrow passages in both scenarios did not, however, generate cases with merged or eliminated objects.

### 5.3 Complex Rounding

This map feature is similar to Narrow Corridors in its effect on the ability to separate objects. Dense area of points causes such an issue. Presence of a complex rounding increases the burden

to produce good shapes as the perimeter may be affected by deformed vertices. Shape Similarity and Semantic Content Similarity are affected in these cases. The first scenario shows some cases where the presence of a complex rounding around inner polygons affected the shape a little bit thus introducing a drop in quality metrics.

#### 5.4 Cantilever Points

The presence of these points creates challenges to produce exact perimeter and shapes. It affects Shape Similarity and Characteristic Similarity metrics. These points can be produced as a result of an algorithm used to generate polygons. Each algorithm may differ in how it deals with these points but one may notice that Concave Hull is the least affected by this property.

#### 5.5 Inner Angles

Inner angles deform the whole polygon shape if wrong tightness values are used. As we notice in the second scenario, polygons 1 and 3 have inner angles and they generated low quality values in Concave Polygons case because of high value of tightness. Tightness in these cases managed to capture better inner angles but it results in breaking straight line segments around local cantilever points.

#### 5.6 Relative Neighbors

This property shows how the choice of metrics may affect time needed to generate quality values. At some cases, not all of map objects are important to be included into calculations. For instance, checking quality for specific parts of the map or neglect tiny objects at specific use cases. Moreover, in use cases where location is already known and the shape matters, Location Similarity can be dropped out.

### 6. CONCLUSION

This paper proposed an approach to quantify the quality of robot maps that we obtain from different environments. First, we modeled the environment and acquired its corresponding pointcloud by performing a robot scan. To obtain a 3D indoor map, three different algorithms enabling the modeling of detected objects (Concave Hull, Convex Hull and Dijkstra Algorithm) were explored. As the core contribution of this paper, we provided a series of quality metrics that aim to capture different aspect of the generated polygons. Finally, we analyzed which aspects of map quality can be visible in which metric. Our approach was tested against different environments.

We showed, that proposed quality quantification approaches are suitable to assess a feature based map obtained by a robot. Besides, by defining quality metrics, we are enabled to identify deformations that may happen through generalization processes, also in the reference map.

### 7. FUTURE WORK

A big topic for 3D indoor maps is georeferencing and relation to outdoor maps. This is particularly important for navigation purposes, where seamless transitions between outdoor and indoor environment are necessary. In indoor, a high-quality and up-to-date map is even more important than in outdoor, as indoor navigation cannot rely on GNSS technology and often must be supported by map matching. In this context, the map quality wins

even more importance, as proposed qualities can be used to assess, whether the match with a reference map can be correct or not. To implement such approach, a single quality value would be easier to handle. Therefore, our future efforts will turn towards combining the individual results of each metric results in a final quality value assisting in marking a map as accepted or not. Furthermore, we plan to work on a quality measure for assessment of the outdoor-indoor co-registration.

### References

- Birk, A., 2009. A quantitative assessment of structural errors in grid maps. *Autonomous Robots* 28(2), pp. 187.
- Cakmakov, D. and Celakoska, E., 2004. Estimation of curve similarity using turning functions. *International Journal of Applied Mathematics* 15, pp. 403–416.
- Durrant-Whyte, H., Rye, D. and Nebot, E., 1996. Localization of autonomous guided vehicles. In: G. Giralt and G. Hirzinger (eds), *Robotics Research*, Springer London, London, pp. 613–625.
- Frank, R. and Ester, M., 2006. A quantitative similarity measure for maps. In: *Progress in spatial data handling*, Springer, pp. 435–450.
- Franois Blair, n.d. Everything you always wanted to know about alpha shapes but were afraid to ask. <http://cgm.cs.mcgill.ca/godfried/teaching/projects97/belair/alpha.html>. accessed online 04/2019.
- Gazebo, n.d. <https://gazebosim.org/>. accessed online 04/2019.
- Iwaszcuk, D. and Stilla, U., 2017. Camera pose refinement by matching uncertain 3d building models with thermal infrared image sequences for high quality texture extraction. *ISPRS Journal of Photogrammetry and Remote Sensing* 132, pp. 33–47.
- Karagiorgou, S. and Pfoser, D., 2012. On vehicle tracking data-based road network generation. In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, ACM, pp. 89–98.
- Karam, S., Peter, M., Hosseinyalamdary, S. and Vosselman, G., 2018. An evaluation pipeline for indoor laser scanning point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-1, pp. 85–92.
- OSM2World, n.d. <http://osm2world.org/>. accessed online 04/2019.
- Pipino, L. L., Lee, Y. W. and Wang, R. Y., 2002. Data quality assessment. *Communications of the ACM* 45(4), pp. 211–218.
- Podolskaya, E. S., Anders, K.-H., Haunert, J.-H. and Sester, M., 2009. 16 quality assessment for polygon generalization. *Quality Aspects in Spatial Data Mining* p. 211.
- Schwertfeger, S., 2012. Robotic mapping in the real world: Performance evaluation and system integration. PhD thesis, Jacobs University Bremen.
- Sester, M. and Förstner, W., 1989. Object location based on uncertain models. In: H. Burkhardt, K. Hähne and B. Neumann (eds), *Muster Erkennung 1989, 11<sup>th</sup> DAGM-Symposium*, Springer-Verlag London, pp. 457–464. ISBN:3-540-51748-0.
- Tran, H., Khoshelham, K. and Kealy, A., 2019. Geometric comparison and quality evaluation of 3d models of indoor environments. *ISPRS Journal of Photogrammetry and Remote Sensing* 149, pp. 29 – 39.

Zhang, L., Zhang, L., Mou, X. and Zhang, D., 2011. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing* 20(8), pp. 2378–2386.