AUTOMATIC BUILDING FOOTPRINT EXTRACTION FROM 3D LASERSCANS

P. Rottmann*, J.-H. Haunert and Y. Dehbi*

Institute for Geodesy and Geoinformation, University of Bonn, Germany
(rottmann, haunert, dehbi)@igg.uni-bonn.de

Commission IV, WG IV/9

KEY WORDS: Building Footprint, Travelling Salesperson Problem, 3D Point Cloud, Kernel Density Estimation

ABSTRACT:

Building footprints are a prerequisite for many tasks such as urban mapping and planning. Such structures are mostly derived using airborne laser scanning which reveals rather roof structures than the underlying hidden footprint boundary. This paper introduces an approach to extract a 2D building boundary from a 3D point cloud stemming from either terrestrial scanning or via close-range sensing using a mobile platform, e.g. drone. To this end, a pipeline of methods including non-parametric kernel density estimation (KDE) of an underlying probability density function, a solution of the Travelling Salesperson Problem (TSP), outlier elimination and line segmentation are presented to extract the underlying building footprint. KDE turns out to be suitable to automatically determine a horizontal cut in the point cloud. An ordering of the resulting points in this cut using a shortest possible tour based on TSP allows for the application of existing line segmentation algorithms, otherwise dedicated to indoor segmentation. Outliers in the resulting segments are removed using Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The segments are then generalized leading to the final footprint geometry. We applied our approach on real-world examples and achieved an IoU between 0.930 and 0.998 assessed by ground truth footprints from both authoritative and volunteered geographic information (VGI) data.

1. MOTIVATION AND CONTEXT

Building footprints represent an indispensable requirement for a wide range of applications such as urban planning, monitoring and disaster management. They are, in particular, a prerequisite for the 3D reconstruction of building and city models (Henn et al., 2013; Biljecki et al., 2015; Arroyo Ohori et al., 2015). In this context, the automatic extraction of building footprints from different remote sensing data such as high resolution ortho photo images (2D Semantic Labeling Challenge, 2022), satellite images (Shi et al., 2018) and airborne laser scans (Mongus et al., 2014) gained more and more attention. Most of the existing approaches yield roof structures from the remotely sensed data. However, the detailed geometry of the underlying footprints which are often hidden by the roof surfaces is difficult to extract. Moreover, the automatic detection of roof outlines itself is even challenging in densely populated areas such as slums or in presence of occluding objects like trees and power lines.

To achieve a more accurate and detailed extraction of building footprints, this paper introduces a novel approach which is based on 3D point clouds covering the buildings' walls. Such point clouds could be acquired with terrestrial laser scanners or laser scanners mounted on drones. Moreover, slant-angle laser scanning from airplanes could be used. The first step towards the 2D footprint is to determine an appropriate height in the 3D point cloud to extract a cross section preserving the geometric footprint characteristics. This is a crucial and decisive step which highly influences the quality of the final result. To this end, we applied a non-parametric estimation of probability density functions (PDF) without relying on any assumptions on the a-priori unknown underlying distribution. Using a kernel density estimation (Wang and Suter, 2004), the estimated PDF reflects the spatial behaviours of the underlying data and a suitable height to select. After an outlier detection and elimination based on Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996), an instance of the well-known Travelling Salesperson Problem (Dantzig et al., 1954; Cormen et al., 2009) is solved to build the boundary of the acquired cross section leading to an ordered segment sequence. This gives access to existing line segmentation approaches of 2D structures. In particular, we used the algorithm of Peter et al. (2017) which is originally dedicated to indoor scans. Our method delivers a regularized footprint outline after a generalization step adjusting the segments from the previous stage.

To demonstrate the feasibility of our proposed method, we applied it on different real-world instances leading to highly detailed footprint structures. Despite the presence of occluding objects such as vegetation, the comparison with ground truth based on authoritative as well as volunteered geographic information (VGI) data states a high confidence in terms of geometric accuracy assessed by *Intersection over Union* (IoU) which is often used as quality value for the area-based building outline extraction (Potůčková and Hofman, 2016). Our approach could contribute to the update of as-planed states of building footprints expanding them by an as-built assessment and in a verification step within a change detection process.

The remainder of this paper is structured as follows: Section 2 shows an overview on the related work whereas Section 3 presents our methods pipeline for the generation of building footprints from 3D point clouds. More specifically, the Subsections 3.1, 3.2, 3.3 and 3.4 deal with the consecutive preprocessing, travelling salesperson, edge segmentation and postprocessing steps respectively. Section 4 demonstrates the feasibility of our approach based on real-world examples. The paper is discussed and concluded in Section 5.

*Corresponding author

2. RELATED WORK

Automatic processing of 3D point clouds of buildings has been an active field of research for years. This often involves the automatic identification and reconstruction of specific building elements, such as windows and doors (Pu and Vosselman, 2007), roofs (Dehbi et al., 2021), or extraction of the building's floorplan (Gankhuyag and Han, 2020). Since our approach aims at the automatic generation of the building footprints, we first highlight recent work in this area.

Light detection and ranging (LiDAR) points are usually used as the data basis for most building footprint recognition approaches. However, Weidner and Förstner (1995) have shown that this can be also achieved with a digital terrain model. In this terrain model, the heights of the buildings are incorporated and used to detect the jumps in their values which allows for inferring the according boundaries. However, the low resolution leads to inaccurate results.

For the determination of footprints, terrestrial laser scans can be used as well as airborne LiDAR data recorded from mobile platforms, e.g. drones or airplanes. In the case of data acquisition from the ground, i.e. terrestrial laser scanning and data acquisition from vehicles, the collected data points lie primarily on the facades of the buildings while these points are mainly belonging to the roof surfaces in the case of airborne laser scanning. Such resulting point clouds of roofs are used by Wei et al. (2008) who present an approach based on α-shapes (Edelsbrunner et al., 1983) to determine the preliminary points' boundary followed by a regularization step. Shen et al. (2011) follows a comparable paradigm by using α-shapes as well and adding height to reconstruct 3D shapes. However, both approaches use merely points of the roof and therefore tend to oversize the footprint. In addition to using α-shapes, Duckham et al. (2008) proposed χ-hulls as another generalization of convex hulls which could be used for the reconstructing. This generalization is as well achieved by shortest-path hulls by de Berg et al. (2011). The difference of χ-hulls from α-shape is the lack of possibility to model holes inside the geometry. Zhang et al. (2006) proposed an approach that first separates points on the ground and buildings, eliminates vegetation and determines the direction of the building based on the longest line segments. The proposed approach of Mongus et al. (2014) follows a similar procedure by focusing on the separation of building and ground points. Afterwards, they compare the resulting areas to ground truth data which shows outliers in regions of vegetation on top of the roofs. These artifacts were tackled by Awrangjeb and Lu (2014) by improving the schematization of airborne scanned buildings, but the correct detection of the direction of the building turned out to be a crucial task which can not be performed consistently.

Using terrestrial laser scans as data source and a Delaunay triangulation of the according 3D point cloud, Pu and Vosselman (2007) extracted geometries, e.g. windows and wall boundaries. They used this information later on for the creation of building outlines (Pu, 2008). However, the approach lacks the ability to deal with occlusions and is primarily developed for facades. A different approach was developed by Hammoudi et al. (2009) using a Hough transform (Hough, 1962). Similar to our approach, they produce a 2D point cloud in a first step which is subsequently used for the Hough transform and serving as input for a KMeans (MacQueen et al., 1967) clustering step. This setup can deal with larger structures of a building but struggles with low point densities and finer details of a building. In comparison, beyond dense point clouds our approach is able to deal with lower point densities as well. As mentioned, a Hough transform can be used to determine edges of such a 2D point cloud (Hammoudi et al., 2009). Nguyen et al. (2005) used additionally a Kalman-filter combined with Random Sampling and Consensus (RANSAC) which, however, led to less reliable results than using an Iterative End Point Fit. This is closely related to the approach of Peter et al. (2017) which uses a range of residuals based on a 2D profile laser scanner and detect different wall segments of building interiors. In our context, we use this algorithm for edge segmentation.

In this paper, we address the challenge of constructing a 2D footprint by using terrestrial laser scans or a mobile platform. Since we cannot always assume normal distributions we applied a kernel density estimation (KDE) to retrieve the best horizontal cut in the point cloud allowing for preserving the detailed structure of the sought footprint. This has been applied similarly by Dehbi et al. (2016) who derived location and shape parameters of building objects, e.g. windows, based on KDE. Before we can apply the edge segmentation, we order the points according to their appearance in a shortest path. This corresponds to the well-known Travelling Salesperson Problem (TSP) (Dantzig et al., 1954). Although it has been shown that the TSP can be solved in nearly linear time (Arora, 1997), we make use of Christofides' algorithm (Christofides, 1975) which guarantees a maximum path length of 50% over the optimal solution.

3. METHODOLOGY

In this section, we present our approach to extract a 2D footprint of a building from its corresponding 3D point cloud. First, we automatically select a cross section of the 3D point cloud leading to a 2D point cloud (Section 3.1) which is ordered using a solution of the Travelling Salesperson Problem (Section 3.2). A generalization of wall segments is then performed building upon a point segmentation which delivers preliminary segments (Section 3.3). In a post-processing step, the process is concluded based on the segmented edges to create a schematic building boundary (Section 3.4). Figure 1 summarises the whole process of our approach which will be explained step by step in the following sections.

3.1 Preprocessing

The input of our processing pipeline is a 3D point cloud P of a given building where all facade walls have been surveyed (cf. step ① of Figure 1). As the resulting building boundary is a 2D geometry, we reduce the dimensionality of the underlying point cloud cutting it horizontally at a specific height h_c. A prerequisite for this cut is the alignment of the point cloud with the horizontal plane. This can be done by leveling the terrestrial laser scanner capturing the data or using additional sensors in mobile applications. The selection of the suitable height h_c of the cut is a crucial step which influences the quality of the acquired result later on (cf. step ②). At heights with many windows there are many gaps in the cross section. This is attributed to the underlying glass surfaces. Such gaps can be avoided setting the cross section, for instance, on a height between two floors. We solve this problem by using a non-parametric approach to estimate a probability density function (PDF) to find a height that corresponds to a high number of points. To this end, we apply kernel density estimation (KDE) (Wang and Suter, 2004) exclusively

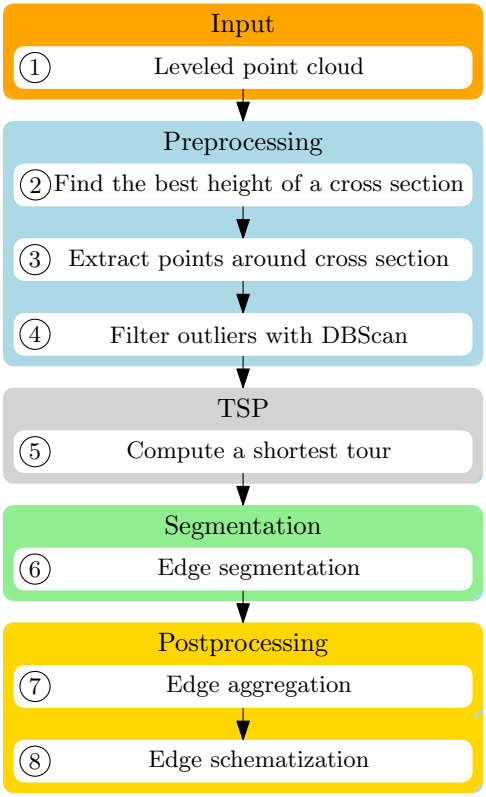


Figure 1: Summary of the processing pipeline of our ap-proach which automatically extracts a 2D footprint from a 3D point cloud.

to the heights of the underlying n points as follows:

$$\widehat{f}_K(z) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{z - z_i}{h}\right) \tag{1}$$

where h stands for a bandwidth whereas K represents the used kernel function placed on every data point. Such a non-parametric estimation does not require any knowledge about the underlying distribution of the input data. When estimating the kernel density, it is important to keep in mind that flat surfaces such as the ground around a building as well as terraces and balconies are densely distributed along the horizontal plane. This causes the distribution to peak at those locations. However, these horizontal planes are not part of the building boundary. Such regions correspond to the highest peaks as can be seen in Figure 2. Hence, we ignore these narrow-band peaks for the determination of the cross-sectional height h_c and rather aim to select points directly below the terraces and balconies. Selecting such kind of points has the advantage of being between two floors and, hence, avoiding window levels. Furthermore, those areas are less likely to be occluded by greenery, e.g. bushes and small trees, in front of the building.

To address this task automatically, we first select the maximum m_{pdf} of the estimated kernel density function of the one-dimensional point cloud. Second, we identify the local maximum $m_{\text{pdf}'}$ of the first derivative of the PDF around m_{pdf} as depicted in Figure 2b (red cross). This height of the maximum in the derivative splits best between an area with balconies and windows which is therefore selected as the height h_c.

After the selection of the cross-sectional height h_c, we define a buffer around the cut with a width w_c (cf. step ③). This allows for obtaining more point candidates C for a subsequent segmentation step. This parameter w_c depends on the point density of the point cloud P. For a high density point cloud, the parameter w_c can be lowered, while it has to be increased for low density point clouds. We denote the resulting point cloud as

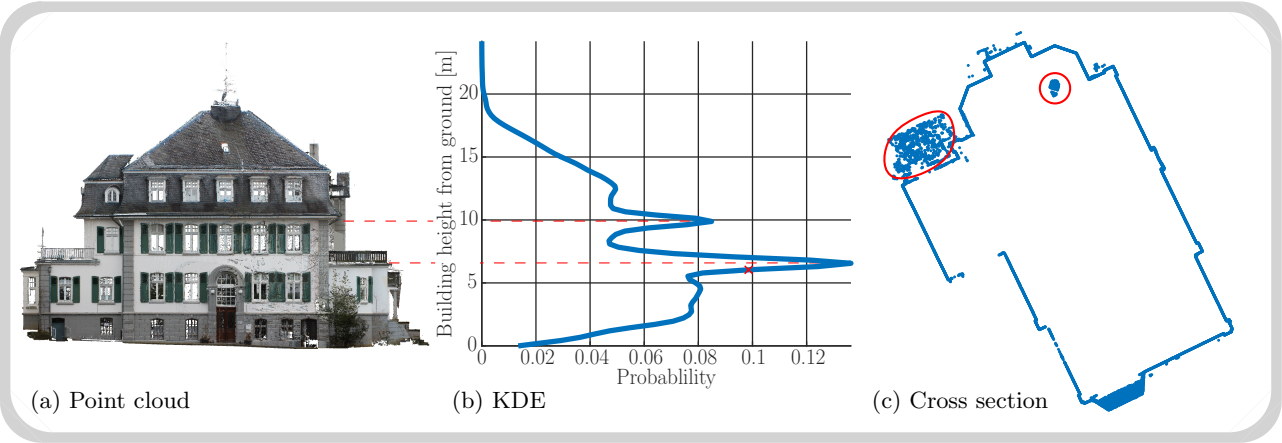
$$C = \{p \in P \,|h_c - \frac{w_c}{2} \leq p_z \leq h_c + \frac{w_c}{2}\}. \tag{2}$$

The set of points C contains not only points belonging to the building boundary but also those which passed the z-coordinate filter of Equation 2, but however, are not part of the building, e.g. through a window captured inside points or vegetation outside points. An example of the extracted cross section can be seen in Figure 2c with vegetation and points surrounded by the red outlines. To filter such irrelevant points, we cluster the points of the selected cross section using DBSCAN (Ester et al., 1996). We then remove the outliers, i.e. all points that cannot be assigned to a cluster (cf. step ④ of Figure 1). For segmenting these remaining points, we want to adapt an approach originally developed for profile laser scanners (Peter et al., 2017). This approach assumed that the points are ordered according to the rotation of the profile scanner and are therefore in a sequential order. To meet such a requirement in our cross section, we compute a shortest round trip through all points of the cross section solving a Travelling Salesperson Problem instance which is explained in the next section (cf. step ⑤).

3.2 Travelling Salesperson Problem

At first sight, a solution yielding the building boundaries consists in calculating a shortest path connecting all points of the cross-sectional point set C. This corresponds to solving the well-known Travelling Salesperson Problem (TSP). However, this solution would incorporate all available points of C and, therefore, undesired outliers with regard to the building boundary. The preliminary TSP solution serves as input for a subsequent edge segmentation towards the final solution. For solving the TSP, we use the approximation of Christofides (1975) which results in a boundary which is at most 50% longer than the optimal boundary. This approximation algorithm has a polynomial running time, which is crucial for large point clouds. Figure 3 summarises the whole process of Christofides' algorithm which will be explained step by step in the following. For the computation of the shortest tour, we generate an edge-weighted undirected graph $G = (V, E)$ where the set of vertices V corresponds to the points of the point cloud. An edge $e \in E$ connects two consecutive vertices weighted by an edge weight $w(e)$. For our task, we define the weight $w(e)$ for the edge $e = (v_i, v_j)$ as the Euclidean distance between v_i and v_j. The edge set E is initialised as empty set in the beginning.

The final goal of the algorithm of Christofides is the computation of a short Hamiltonian cycle. The Hamiltonian cycle is a cycle which is visiting each vertex of the path exactly once. The first step of this algorithm for finding such a cycle is a minimum spanning tree (MST) (cf. step ① of Figure 3). For computing the MST T, we use the algorithm of Prim (1957) and add the edges E_T which are part of T to our graph G. Adding an edge e to G increases the degree, i.e. number of edges, of both nodes of the edge by one. Since we want to achieve a round trip at the end only even degrees are allowed for each node. Therefore, we select the nodes $V' \subseteq V$ of G which have an odd degree (cf. step ②) and compute a matching of minimum weight for



(a) Point cloud (b) KDE (c) Cross section

Figure 2: Automatic determination of the suitable height for a cross section exemplified by the building ESTATE. Based on the 3D point cloud, a one-dimensional, z-directional, probability density function (PDF) is computed, provided by a kernel density estimator (KDE). This is used as a prior for the selection of a cross-section to be chosen. The red cross in the probability density function is the selected height of h_c of our approach. The red outlines of the cross section highlight areas with many outliers as vegetation or points inside the building.

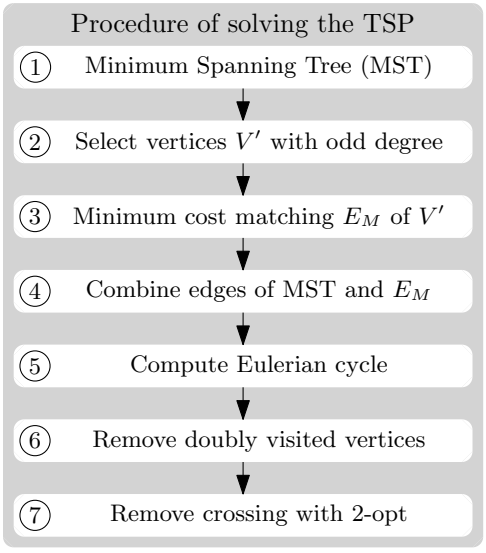


Figure 3: Summary of Christoffides' algorithm with an ad-ditional step of removing crossings.

the nodes V' (cf. step ③). The matching results in new edges E_M which minimize the Euclidean distance:

$$\min \sum_{e \in E_M} w(e). \tag{3}$$

The added edge set E_M increases the degree of nodes V' to be even. After adding both E_T and E_M to G (cf. step ④), we compute an *Eulerian* cycle (cf. step ⑤). In this cycle, all nodes can be visited at least once due to the underlying MST. However, this cycle can contain nodes multiple times which unnecessary increases the boundary length. By removing multiple visits of the same node, we achieve a Hamiltonian cycle as result of Christofides' algorithm (cf. step ⑥).

However, a crossing-free resulting path is not guaranteed by Christofides' algorithm. Crossings contribute not only to an increased path length but also lead to topologically incorrect boundaries. Hence, we perform the k-Opt heuristic (Lin, 1965) to remove the crossings (cf. step ⑦). Although the k-Opt heuristic can result in an exponential running time, we opt for an early stop after a single pass of the algorithm which is sufficient in the context of our task. Performing k-Opt once is sufficient for our approach as we already solved the TSP and apply the k-Opt heuristic for resolving crossings only. Hence, this results in a running time of $O(n^2)$ in our use case rather than the expected exponential running time.

3.3 Edge Segmentation

After the ordering of points according to the appearance in the boundary, the task can be reduced to a 2D problem allowing for the application of methods originally proposed for 2D laser scans (cf. step ⑥ of Figure 1). In this context, we make use of an algorithm of Peter et al. (2017), originally dedicated to 2D profile laser scans for indoor environments, to obtain a segmentation of building boundaries. The edge segmentation is based on estimating lines through n points incorporating mean residual values. This mean residual is expected to be $\sigma/\sqrt{n}$ where σ is referring to the sensors accuracy. When adding a new point to a segment of n points, we check if the point's mean residual is less than $3\sigma/\sqrt{n}$. If this holds, the point is added to the segment. Using 3σ as threshold corresponds to a 99.7% confidence interval. In case the mean residual of the point is greater than the specified threshold, the line is fitted through the segment consisting of n points and added to the final list of lines. However, each segment has to fulfill a specified minimum segment length. This aims at reducing small segments in cluttered and noisy regions of the point cloud. Furthermore, the segments are initialized with a minimum length N.

3.4 Postprocessing

The edge segmentation is prone to a cluttered result line set. This is the result of finishing a segment when the next investigated point has a distance to the line of more than $3\sigma/\sqrt{n}$ with n being the number of points of the segment. To overcome this small deviations in the building boundary, we perform a schematization step (cf. step ⑦). In this last step of the processing pipeline, we estimate a line through every segment and

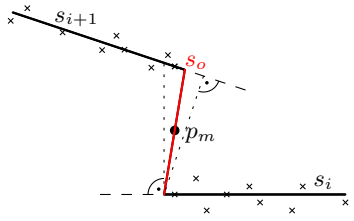


Figure 4: Determination of the distance d_o between two nearly parallel consecutive line segments s_i and s_{i+1} based on the midpoint p_m between the endpoints of the line seg-ments.

compare it to the subsequent lines. In case the direction change between consecutive lines is greater than a specified threshold, we keep both individual lines. If the angle between those two adjacent lines is below the threshold and the distance d_o between those line segments is below a distance threshold, we combine the points of both segments to a single segment and estimate a new line though the merged segment. With this process we are able to merge co-linear lines to schematize the footprint and create a single polygon without gaps. For our experiments, we set this angle threshold to 20^o.

To calculate the distance d_o between the two line segments s_i and s_{i+1}, we first create an orthogonal line for each of the line segments; see Figure 4. The average slope of these two line segments is used to create a new line segment s_o which has the same angle in comparison to s_i and s_{i+1}. Then we determine the midpoint p_m between the end of the first segment and the beginning of the second segment, which together with the slope forms a new line segment s_o between the two given segments. The line segment s_o is limited by the intersections with the line segments s_i and s_{i+1} while its length is used as the distance d_o. If the distance d_o is above the specified distance threshold, s_o is added in between s_i and s_{i+1}. Using the combined lines, we estimate intersection points of consecutive lines and limit the lines to line segments between those intersections. In the end, the building boundary is created by the line segments between adjacent intersections.

4. EXPERIMENTS

For analyzing the results of our approach, we outline experiments on two point clouds captured with a Leica HDS6100. As first building we execute our approach on a building which belong to the Campus of the University of Bonn in Germany. This building will be denoted as UNIVERSITY in the rest of this paper. UNIVERSITY consists of long, straight edges with all bends being nearly 90^o, no balcony and very little occlusions; see Figure 5a. The point cloud consists of 10^6 points. In order to have an equally distributed point cloud, we opt to subsample beforehand. The subsampled point cloud has $527k$ points with a distance between points of at least 5 cm. The second building, which we call ESTATE, has a more complex shape and its point cloud consists of nearly $44 \cdot 10^6$ points which is sampled in the same manner as for UNIVERSITY to $511k$ points. The original point cloud of ESTATE is depicted in Figure 2a. As the original resolution of the point cloud of ESTATE is higher in comparison to UNIVERSITY, the downsampling of the points for ESTATE is significantly higher with respect to UNIVERSITY. Our aim is to demonstrate that a relatively sparse point cloud is also suitable to estimate the building boundary. Hence, we opt to sample points with pitch of $5cm$ which is also beneficial in term of processing time and memory consumption. We applied a space-based sampling rather than a random sampling as this results in a more uniformly distributed point cloud. This space-based sampling over random sampling enables us to extract meaningful heights from the estimated probability density function as this allows us to successfully detect the according holes of windows in the data. If there is an increased point density at an elevation where windows are present, this would not be possible with our approach because no decrease or increase would be detectable in the estimated probability density function. However, there are enough points left to estimate the building boundary. The decreased point density can also be a side effect of faster scanning with lower resolution.

4.1 Selecting a cross section

The first step of our processing pipeline is the detection of a height for a reasonable cross section. In order to do this, we determine the PDF from both datasets as can be seen in the middle of Figure 2 and Figure 5b. We reduce the points by the absolute height so that the lowest point of the point cloud is at 0 m. This, however, only affects the height scale of the plot but does not influence the resulting probability density function. Considering the KDE result of UNIVERSITY, three local maxima at heights of 4.0 m, 7.1 m and 10.3 m as well as local minima at 1.8 m, 6.0 m and 9.3 m are present. The local minima correspond to the heights where windows are present, and thus, fewer points are recorded. We can therefore conclude that the building consists of three floors. In the case of the maxima, the building outline is without major data gaps, which increases the number of captured points. In order to select a cross section at a window-free height, we apply the method described in Section 3.1. This results in a height h_c of 10.0 m. With the selected height h_c at the top of the third floor it is not possible to detect the entrance of the building. The entrance is only present if a height h_c at the top of the first floor has been selected. Unfortunately, in the underlying point cloud a large portion of the wall in the lower right of the building is missing for the lower two floors. For this reason, our approach does not select the corresponding height. Our extracted cross section for UNIVERSITY is shown in Figure 5c. We perform a similar procedure for ESTATE. The building has two terraces, which results in dense clusters on the horizontal level accordingly. The challenge for such a building is to select an area as high as possible below the terraces; see Figure 7. Our approach turns out to be also suitable for such circumstances. The horizontal planes of the terraces lead to high probability density values at those heights. Our approach is able to automatically select a cut in the shown red corridor above the windows and below the ceiling. In this case, the resulting parameter of our approach is $h_c = 6.08$ m with the same $w_c = 0.15$ m; see Figure 2b. In both examples, we choose the width of the cut $w_c = 0.15$ m which corresponds to a typical thickness of ceilings in such building style. The cross sections of the two buildings contain 12017 points in case of UNIVERSITY and 6989 points for ESTATE respectively.

The selected cross section still reveals artifacts from the underlying point cloud such as points within the building that were captured through windows or vegetation present around the building. For this reason we apply the DBSCAN which has been executed with a radius of 1 m and a cluster size of 15. In both cases, ESTATE and UNIVERSITY, not all outliers can be removed, however, the eliminated outliers turn out to be beneficial for improving the quality of the building outlines.

The filtered cross section represents the input of the algorithm

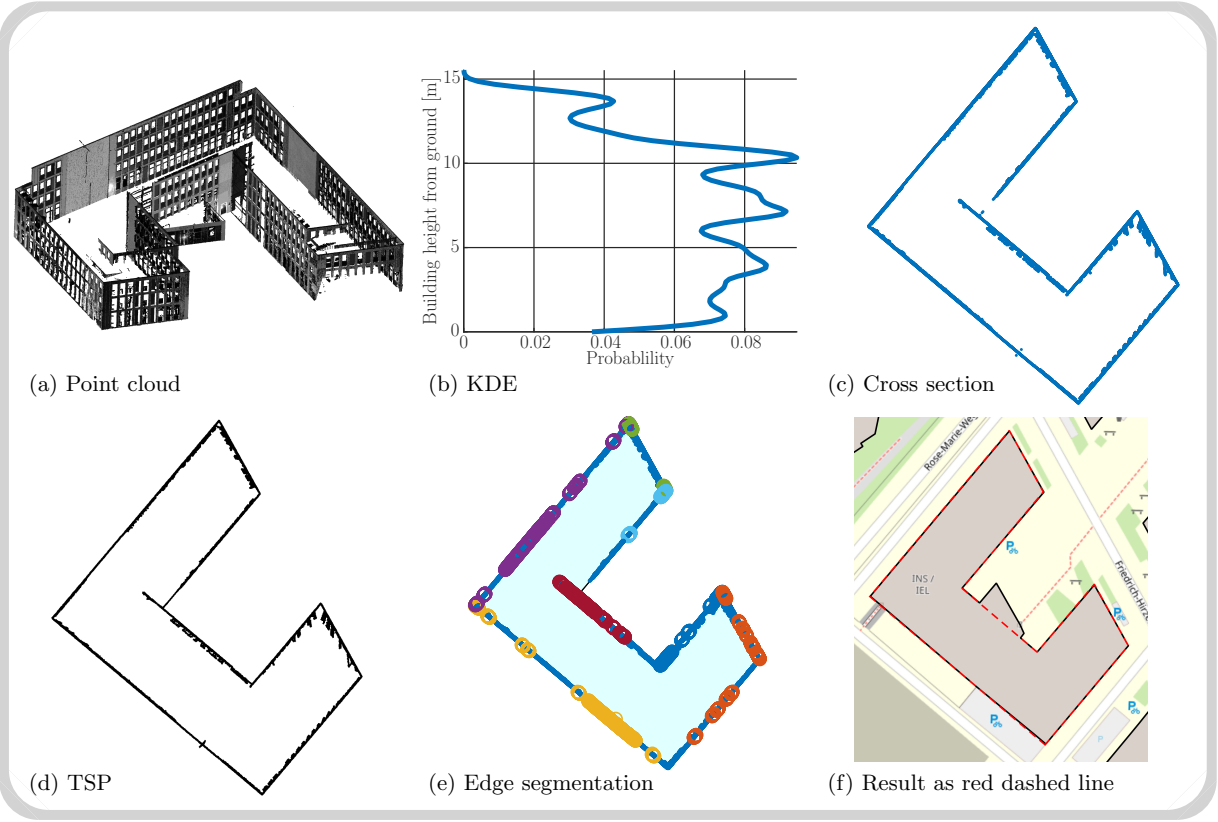


Figure 5: The pipeline of our approach demonstrated on the UNIVERSITY campus in Bonn/Germany. The skillfully selection of the cross section (c) based on a kernel density estimation (b), is a decisive prior to solve a travelling salesperson problem (d) followed by an edge segmentation (e) and leading to a final footprint (f) of high accuracy.

of Christofides. We focus on the result for ESTATE as this shows the effects of the outliers; see Figure 6a. Based on this preliminary result, it is obvious that simply connecting the points is not sufficient to represent an outline of the building, since it includes every point of the cross section. Nevertheless, the retrieved polygon is a simple polygon with no crossings.

4.2 Edge segmentation

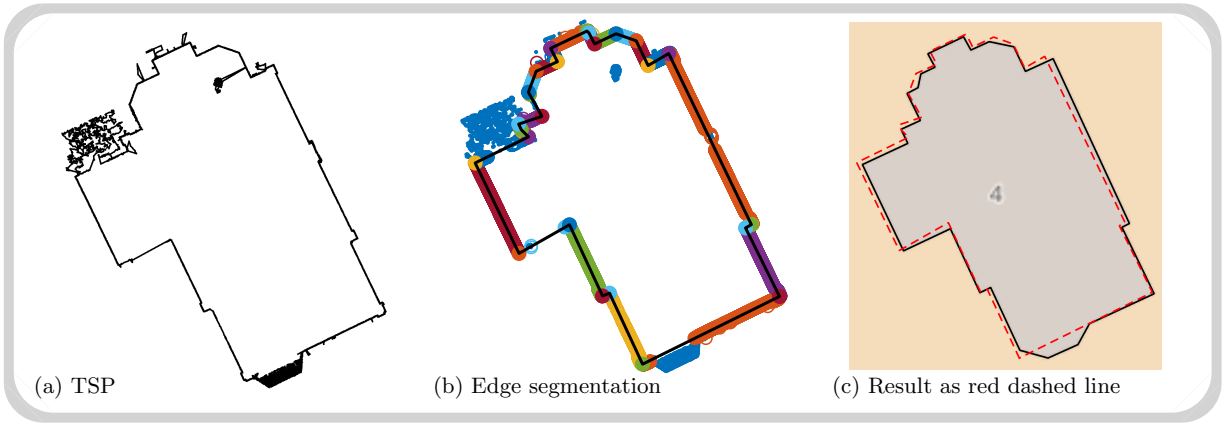
The resulting TSP tour is the basis predefining the order of the points for the subsequent edge segmentation. For σ, we choose 1.2 mm for both buildings, which corresponds to the accuracy of the laser scanner. We set 15 as the minimum number of points per edge, while the segments are initialized with 8 points. Both parameters, i.e. the minimum number of points per segment and number of points for initializing, have to be changed for differing point densities accordingly. Applying this to UNIVERSITY, we obtain 56 segments, while for ESTATE we get 80 segments. However, the majority of those identified segments belong to the same walls, while some wall segments are missing. The increased number of resulting segments is attributed to the presence of outliers, i.e. lying further as $3\sigma/\sqrt{n}$. Such outliers trigger new segments accordingly. Those segments are merged in the postprocessing and in case of parallel segments, new line segments are added. We select 0.2 m as the distance threshold for merging parallel segments. For smaller distances, we merge both segments to a single segment. By increasing this threshold it is possible to increase the degree of generalization of the building boundary. We select the angular threshold as 20^o. This angular threshold is based on the angles of an octilinear grid where all angles between line segments are multiples of 45^o. When schematizing to such a grid, it would be possible to merge all segments with angular deviations up to 22.5^o. However, since we are dealing with missing observations, measurement uncertainties, and occluded segments, we reduce the threshold to 20^o.

4.3 Results

The line segment merging results in 8 line segments for UNIVERSITY and 24 line segments for ESTATE respectively; see Figure 5f and Figure 6c. The result of our approach fits well to the underlying structure of the cross section in case of UNIVERSITY. However, in case of ESTATE the result is not satisfactory in the top left part. This is due to the presence of bushes in front of the building occluding parts of the wall and responsible for additional undesirable points. Furthermore, our result appears to be slightly rotated in comparison to the ground truth data of OpenStreetMap (OSM) (OpenStreetMap contributors, 2022). This deviation results from the inaccurate registration via GPS measurements, which causes a mismatch between the underlying point cloud and the corresponding ground truth footprint.

We further quantify the deviations of our results in comparison to footprints from OpenStreetMap and Geobasis NRW (2022) of the state of North Rhine-Westphalia in Germany (Federal Agency for Cartography and Geodesy, 2022) by computing the Jaccard Index (Jaccard, 1901), also known as *Intersection of Union*,

$$IoU = \frac{\text{Area}(F_1 \cap F_2)}{\text{Area}(F_1 \cup F_2)} \tag{4}$$



(a) TSP (b) Edge segmentation (c) Result as red dashed line

Figure 6: ESTATE building: Outliers, i.e. points not belonging to facade surfaces, present in the TSP solution highly influence the quality of the extracted footprint. Our approach can although deliver a satisfactory result.

Figure 7: The automatically identified cross section level (in red color) for ESTATE.

and discrete Hausdorff Distance

$$d_{\mathrm{H}} = \max_{v_1 \in V(F_1)} \{ \min_{v_2 \in V(F_2)} \{ d(v_1, v_2) \} \}. \tag{5}$$

In both equations, F_1 and F_2 denote the two compared footprint polygons and $V(F_1)$ and $V(F_2)$ are their corresponding vertex sets. To compute the distance $d(v_1, v_2)$ between two vertices v_1 and v_2, the Euclidean distance has been used.

Comparing ESTATE to OpenStreetMap and footprints of Geobasis NRWGeobasis NRW (2022), the results are very similar using IoU as quality measure with $IoU = 0.9304$ and $IoU = 0.9305$ respectively. The Hausdorff distance is $d_{\mathrm{H}} = 1.161$ m and $d_{\mathrm{H}} = 1.173$ m accordingly. The same comparison for UNIVERSITY leads to IoU scores of 0.9620 and 0.9698 and Hausdorff distances of 4.852 m and 4.782 m. The higher Hausdorff distance of UNIVERSITY is a consequence of the missing entry of the building in the cross section and therefore in our result. Comparing the remaining part of the footprint reveals, however, higher IoU values. Using OpenStreetMap, we can compare the result of UNIVERSITY to the ground truth without the entrance which is given as a separate footprint. This obviously improves both metrics to 0.9980 and 0.041 m respectively. This fits well to the accuracy of the ground truth, i.e. 0.03 m and, hence, very suitable for a wide variety of tasks such as cadastre related applications. In this context, our approach can, for instance, serve for the update of the as-is state of footprints after an automatic verification process.

5. CONCLUSION AND OUTLOOK

This paper introduced a novel approach for the automatic extraction of 2D footprints from 3D laser scans. In contrast to common approaches which use airborne point clouds, terrestrial point clouds have been used allowing for a more accurate detection of the footprint geometry which is otherwise occluded by roof structures. We proposed a pipeline of methods starting by an automatic and accurate determination of a height position for a horizontal cross section based on a one-dimensional, i.e. in z-direction, kernel density estimation (KDE). The acquired points are ordered and connected solving a Travelling Salesperson Problem (TSP) which opens up opportunities to apply a segmentation algorithm previously used in the context of indoor mapping. The resulting detailed footprint is acquired after an outlier elimination using DBSCAN and a generalisation step which merges nearly parallel segments. Our approach has been successfully applied on challenging real-world examples and turns out to be suitable for the extraction of footprints with a high accuracy assessed with regard to both authoritative and VGI data. In an ongoing work, a wider range of case studies will be performed to confirm the presented results so far.

Despite the presence of outliers, e.g. due to vegetation, our approach turns out to be a good mean for the footprint extraction task. However, the TSP step could be expanded in order to deal with such outliers and ignoring them in the preliminary TSP result accordingly. Additionally, the segmentation is only able to deal with line segments and could be expanded to segment curved and non-linear footprint sections as well. Our method has been demonstrated on terrestrial point clouds and can be similarly applied on point clouds stemming from mobile platforms, such as drones. The investigation of the transferability of the approach can be studied for photogrammetric point clouds which opens up the opportunity to reason on potential outliers and eliminate them based on their line of sight with regard to the capturing camera position.

ACKNOWLEDGEMENTS

The authors are grateful to Martin Blome for the provided point clouds and Noémie Treff for performing some implementation tasks. We gratefully acknowledge the open datasets from OpenStreetMap and Geobasis NRW.

REFERENCES

2D Semantic Labeling Challenge, 2022. http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html. Accessed: 2022-07-05.

Arora, S., 1997. Nearly linear time approximation schemes for euclidean tsp and other geometric problems. *Proceedings 38th Annual Symposium on Foundations of Computer Science*, IEEE, 554–563.

Arroyo Ohori, K., Ledoux, H., Stoter, J., 2015. A dimension-independent extrusion algorithm using generalised maps. *International Journal of Geographical Information Science*, 29(7), 1166–1186.

Awrangjeb, M., Lu, G., 2014. Automatic building footprint extraction and regularisation from lidar point cloud data. *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, IEEE, 1–8.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889.

Christofides, N., 1975. *Graph theory: An algorithmic approach (Computer science and applied mathematics)*. Academic Press, Inc.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2009. *Introduction to Algorithms*. MIT press.

Dantzig, G., Fulkerson, R., Johnson, S., 1954. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2(4), 393–410.

de Berg, M., Meulemans, W., Speckmann, B., 2011. Delineating imprecise regions via shortest-path graphs. *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, 271–280.

Dehbi, Y., Henn, A., Gröger, G., Stroh, V., Plümer, L., 2021. Robust and fast reconstruction of complex roofs with active sampling from 3D point clouds. *Transactions in GIS*, 25(1), 112–133.

Dehbi, Y., Staat, C., Mandtler, L., Plümer, L., 2016. Incremental refinement o f facade m odels w ith a ttribute g rammar from 3d point clouds. *Proc. XXIII ISPRS Congress*, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, III-3, 311–316.

Duckham, M., Kulik, L., Worboys, M., Galton, A., 2008. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern recognition*, 41(10), 3224–3236.

Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4), 551–559.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *kdd*, 96, 226–231.

Federal Agency for Cartography and Geodesy, 2022. https://www.bkg.bund.de/EN/Home/home.html. Accessed: 2022-07-05.

Gankhuyag, U., Han, J.-H., 2020. Automatic 2d floorplan cad generation from 3d point clouds. *Applied Sciences*, 10(8), 2817.

Geobasis NRW, 2022. Open data – digitale geobasisdaten nrw. https://www.opengeodata.nrw.de/produkte/. Accessed: 2022-07-05.

Hammoudi, K., Dornaika, F., Paparoditis, N., 2009. Extracting building footprints from 3D point clouds using terrestrial laser scanning at street level. *ISPRS/CMRT09*, 38, 65–70.

Henn, A., Gröger, G., Stroh, V., Plümer, L., 2013. Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76(0), 17-29.

Hough, P. V., 1962. Method and means for recognizing complex patterns. US Patent 3,069,654.

Jaccard, P., 1901. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull Soc Vaudoise Sci Nat*, 37, 547–579.

Lin, S., 1965. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, 44(10), 2245–2269.

MacQueen, J. et al., 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1number 14, Oakland, CA, USA, 281–297.

Mongus, D., Lukač, N., Žalik, B., 2014. Ground and building extraction from LiDAR data based on differential morphological profiles and locally fitted surfaces. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 145–156.

Nguyen, V., Martinelli, A., Tomatis, N., Siegwart, R., 2005. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 1929–1934.

OpenStreetMap contributors, 2022. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org. Accessed: 2022-07-05.

Peter, M., Jafri, S., Vosselman, G., 2017. Line segmentation of 2D laser scanner point clouds for indoor slam based of a range of residuals. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Potůčková, M., Hofman, P., 2016. Comparison of Quality Measures for Building Outline Extraction. *The Photogrammetric Record*, 31(154), 193-209.

Prim, R. C., 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6), 1389–1401.

Pu, S., 2008. Generating building outlines from terrestrial laser scanning. *Proc. XXI ISPRS Congress*, 5, International Society for Photogrammetry and Remote Sensing (ISPRS), 451–455.

Pu, S., Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. *Intl Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, 12–14.

Shen, W., Zhang, J., Yuan, F., 2011. A new algorithm of building boundary extraction based on lidar data. *2011 19th International Conference on Geoinformatics*, IEEE, 1–4.

Shi, Y., Li, Q., Zhu, X. X., 2018. Building footprint generation using improved generative adversarial networks. *IEEE Geoscience and Remote Sensing Letters*, 16(4), 603–607.

Wang, H., Suter, D., 2004. Robust adaptive-scale parametric model estimation for computer vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(11), 1459–1474.

Wei, S. et al., 2008. Building boundary extraction based on lidar point clouds data. *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37, 157–161.

Weidner, U., Förstner, W., 1995. Towards automatic building extraction from high-resolution digital elevation models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 50(4), 38-49.

Zhang, K., Yan, J., Chen, S.-C., 2006. Automatic Construction of Building Footprints From Airborne LIDAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9), 2523-2533.