

HIERARCHICAL PROXIMITY-BASED OVER-SEGMENTATION OF 3-D POINT CLOUDS FOR EFFICIENT GRAPH FEATURE DETECTION

Z. Shtain,* S. Filin

Mapping and Geo-Information Engineering, Technion – Israel Institute of Technology, Haifa, Israel
(zachis, filin)@technion.ac.il

Commission II

KEY WORDS: Segmentation, Point clouds, Ball-tree, Graph-based clustering, Feature extraction

ABSTRACT:

Point cloud simplification is empowered by the definition of similarity metrics which we aim to identify homogeneous regions within the point-cloud. Nonetheless, the variety of shapes and clutter in natural scenes, along with the significant resolution variations, occlusions, and noise, contribute to inconsistencies in the geometric properties, thereby making the homogeneity measurement challenging. Thus, the objective of this paper is to develop a point-cloud simplification model by means of data segmentation and to extract information in a better-suited way. The literature shows that most approaches either apply volumetric data strategies and/or resort to simplified planar geometries, which relate to only part of the entities found within a natural scene. To provide a more general strategy, we propose a proximity-based approach that allows an efficient and reliable surface characterization with no limitation on the number or shape of the primitives which in turn, enables detecting free-form objects. To achieve this, a local, computationally efficient and scalable metric is developed, which captures resolution variation and allows for short processing time. Our proposed scheme is demonstrated on datasets featuring a variety of surface types and characteristics. Experiments show high precision rates while exhibiting robustness to the varying resolution, texture, and occlusions that exist within the sets.

1. INTRODUCTION

Laser ranging based mapping has gained increased popularity in recent years, mainly due to the accurate and detailed surface characterization that these systems provide. The resolution and accuracy of laser-scanning-based point clouds make them a preferable source for a broad spectrum of applications. Yet, a great challenge is associated with processing and interacting with these clouds because of their volume and wide span in 3-D space. As a consequence, the aggregation of homogeneous regions into segments has become a customary step towards information extraction, modeling, and object detection, to name only a few applications (Adam et al., 2019; Xing et al., 2019). Point cloud segmentation is empowered by definitions of similarity and aims to identify homogeneous regions and elements within the data. Similarity is often measured in terms of consistency in geometric attributes among neighbouring elements. However, the variety of shapes and the existence of clutter, along with intrinsic data related challenges like variations in resolution, uneven spacing, occlusions and, noise, contribute to inconsistencies in the geometric properties and complicate homogeneity measurement.

With the increasing volumes of point clouds, the computational demands for their segmentation rise as well. Using a region-based partitioning where pre-defined patches consisting of subsets of the points are used as the basic elements, can significantly reduce the computation cost (Yang et al., 2015). As a result, such approaches become more attractive. Examples of such atomic units include: voxels, slices and planar fragments (Wu et al., 2013; Zolanvari and Laefer, 2016; Fan et al., 2018). While the classic voxelization provides an index-based partitioning, it neglects the point clouds properties such as varying point density. Therefore, an octree structure is the

more commonly used approach (Vo et al., 2015; Su et al., 2016; Dong et al., 2018; Xu et al., 2018). Nonetheless, selecting an appropriate *voxel* resolution (i.e., the termination criterion for the octree construction) is crucial to the accuracy of the extracted features. As an alternative, supervoxels methods have been proposed, where the original voxels (which are usually defined with a fixed resolution) were aggregated to structural primitives based on similarity (Papon et al., 2013; Lin et al., 2018). However, these approaches are sensitive in sparse areas where the total number of points is limited (i.e., erroneous attribute computation) or in the presence of linear objects (Li and Wu, 2019). Despite the significance of initial partitioning of the point-cloud into regions, it is important to keep in mind that it is merely an over-segmentation of the data, clustering these over-segmented patches into meaningful objects still remains the main challenge. Therefore, such partitioning should aim to reduce the overall computational demands of the segmentation process without increasing its own. Such an optimal over-segmentation is lacking, as evident by increasing number of studies that tackle that aspect (e.g., Landrieu and Simonovsky, 2018; Lin et al., 2018; Li and Wu, 2019).

Most recent methods proposed graph representation for segmentation purposes. In this form, the graph nodes represent the basic units, while the edges define and measure the similarity between neighboring ones. A computational efficient graph construction often relies on the data-structure used to extract the basic units (Dong et al., 2018; Xu et al., 2016). This representation allows using graph-partitioning methods such as Graph-Cuts (Green and Grobler, 2015; Landrieu and Simonovsky, 2018) or probabilistic methods like Markov Random Fields (MRF; Niemeyer et al., 2014; Li et al., 2019) or Conditional Random Fields (CRF; Niemeyer et al., 2014; Landrieu et al., 2017; Vosselman et al., 2017). The graph topology impacts the resulting segmentation is two-ways: a large

*Corresponding author

number of edges can yield better results, but implies a heavier computational burden.

In this paper, we propose a proximity-based strategy for hierarchical partitioning of 3-D point clouds. Our method utilizes a data-structure that not only captures the actual distribution of the points in their native space and provides efficient neighbor querying, but also allows immediate extraction of the basic elements of our segmentation scheme. The proposed partitioning supports a more reliable geometric attribution. This, in turn, enables a computational efficient graph-based clustering of the basic elements into meaningful segments.

The remainder of this paper is organized as follows: a details description of the proposed segmentation scheme is found in Sec. 2. Sec. 3 analyzes the performance of the proposed methodology on a set of point clouds each featuring unique properties (e.g., scanning density, varying object complexity, etc.). Finally, the conclusion of this work is presented in Sec. 4.

2. METHODOLOGY

Motivated by the redundancy that characterizes the data, our segmentation strategy revolves around aggregates of points. To an extent, the segmentation is applied to entities of spatial context, rather than the individual points, while benefiting from the reduction in volume and analyses related to point-wise evaluations. To manage the data – points and aggregates – a structure should, *i*) allow for an efficient access and data retrieval, *ii*) facilitate an efficient geometric features computation, and *iii*) establish partitioning both density and proximity. The literature shows a variety of structures for the management of laser scanning data, and it is clear that no ‘consensus’ structure exists for laser scans. The discussion on data structures for point cloud management can be roughly divided into two separate ones: atomic region defining for segmentation purposes and efficient neighbor querying. In the former, the voxels and octrees data-structures are the dominant ones while the kd-tree is not even part of the conversation, whereas for nearest neighbor querying the case is inverted. An example of this is shown in Hackel et al. (2016) where the voxels data-structure is used for extracting geometric features while the nearest neighbor querying is performed with a kd-tree one. However, volumetric partitioning (as in the case of octrees) is lacking for laser scanning derived point clouds, considering that the data can be regarded as 3-D manifolds. Therefore, we first address data arrangement as part of our segmentation process. We propose here a proximity-based structure that allows splitting points into subsets that correspond to the scanned elements. Moreover, we show that this structure can be tuned to fit, query, store, and access point aggregates that respond well to the characteristics of laser scans, which enhances the geometric characterization of the scanned entities.

An outline of our proposed segmentation algorithm is as follows: first, the point cloud is arranged in a ball-tree data-structure. Next, equally-sized nodes, corresponding to the smallest detectable objects within the scene, are extracted to be utilized as basic elements in the segmentation scheme. Then, a connectivity graph is established amongst the extracted features, where a pairwise similarity determines the weights of the edges. The final segmentation is achieved by a two-level connectivity analysis.

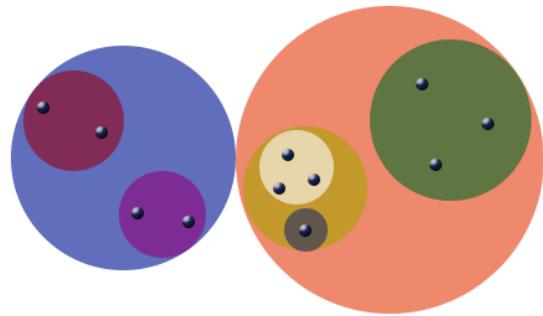


Figure 1: Hierarchical partitioning of a point set (blue dots) using the ball-tree model. Nodes are represented as colored circles

2.1 Data arrangement

In our segmentation scheme, we adopt the ball-tree representation (Fig. 1), a binary structure that maintains spatial data hierarchically. Each node in the tree defines a region that consists of all the points bounded by a hyper-sphere (in fact plane) whose interior leaves are chosen so that the distance between them is maximal. Construction of the tree is performed by first computing the centroid of the point-set, then setting the point with the greatest distance from this centroid as the center of the first child and setting the second child as the farthest point from the first one. Points are then assigned to each child-node by their proximity. This procedure is repeated for each node until a termination criterion is reached. Sub-division at each node can be understood as finding the orthogonal hyperplane that bisects the line connecting the two centers. Unlike kd-tree construction, balancing the number of points per node is not prioritized as there is no constraint on the number of points per child-node. While the unbalanced tree is larger (and construction time is longer) than its balanced counterpart, querying with such trees might be significantly more efficient as they capture the true distribution of the points in their native space (Kumar et al., 2008). The hierarchical subdivision that this representation provides, generates nodes with decreasing size as the node-level increase. Notably, the number of points on a given node-level varies, based on their distribution. When comparing the partitioning derived from the ball-tree data-structure to the traditional (hierarchical) voxelization (Fig. 2), the advantage of the former over the latter is evident, as the ball-tree sub-division is invariant to coordinate values, therefore, minimizing the aggregation of points from multiple surfaces (e.g., building corner) to the same node.

Algorithm 1 Ball-tree construction

```

1: procedure BALL-TREE CONSTRUCTION
2: Input: set of 3-D points and  $n_{thr}$  – max number of points
   in node
3: Result: organized point set in ball-tree data-structure
4:   insert entire set to root node
5:   while number of points in node  $\geq n_{thr}$  do
6:     compute center of mass in node
7:     set  $x$  as the farthest point in node from center
8:     set  $y$  as the farthest point in node from  $x$ 
9:     move points closer to  $x$  to childnode A
10:    move points closer to  $y$  to childnode B
11:    goto 5 for childnode A
12:    goto 5 for childnode B
13:   return ball-tree data-structure

```

In our implementation, the tree construction is guided by its *i*) to basic units for the segmentation, *ii*) establish connectivity (topology) among these elements; *iii*) conduct a point-level

nearest-neighbor search; and *iv)* the quality of the segment. These usages guide termination criterion, and we identify three alternatives: the minimal area (radius); by conformity to a given shape, and by the number of points per leaf. The first two are synonymous, while the third has no spatial motivation, but influences the querying efficiency, e.g., the (*k*-) nearest neighbors. The usefulness of the tree can be observed here, as area, shape, and size (number of points per node) can be addressed simultaneously and interchangeably. The hierarchical nature of the ball-tree suggests that each node has area-(radius) and shape-related properties while its overall depth can be dictated (or be governed by) the maximal number of points. The termination criterion is set as the number of permissible points within a node (Alg. 1), and thereby guarantee an efficient neighborhood querying. At the same time, we approach shape confidence related analyses by extracting surface elements as a function of both the shape conformity property of a given node in the tree and its area one.

2.2 Elemental units extraction

Once the point cloud is arranged in the ball-tree data structure, we aim to exploit its hierarchical division strategy to define the atomic units of our segmentation. As the tree construction is distance driven, it is invariant to translation and rotation. Its construction is also uninfluenced by the varying spatial resolution of laser-scan related point-clouds. Because of the irregular point distribution and varying resolution, node-depth and radii have no exact correspondence – nodes of similar depth may have different radii, and conversely, a given radius may be found at different depths. A question that arises is how these leaves/nodes may be interpreted and analyzed. Since smaller leaves do not represent an actual spatial entity, their interpretation is equivalent to a point-based analysis. A surface-element of a given dimension requires to identify nodes whose radius is of the sought-after size. These may appear at different depths and the number of points they contain may vary. However, the number of points per surface element has no effect as points are collected for a given node, irrespective of its depth. Conversely, if the size-related query has reached a leaf (namely, size exceeds the predefined radius) we may assume that these points have been grouped merely as no proximal points exist and that their aggregation is consequential. We may assume that these points relate to noise or clutter and that it is unlikely that they relate to an actual surface.

Angular resolution vs. size – since size is the deciding factor, the spatial resolution of the scan should also be discussed. With laser scanners, spatial resolution corresponds to the angular spacing that is set. Thus, distant objects will be sparsely sampled compared to closer ones. In that respect, a tradeoff may exist between the physical or spatial context in which a surface-element may be defined and the scanning resolution. Motivated by the understanding that segmentation aims to identify geometrically consistent physical entities; our implementation defines a surface element by a minimal radius (size) but relaxes it to permit up to three-times its size, as long as the minimal number of points criterion is met. Our evaluations have shown that spatial resolution considerations did not lead to misses in aggregating surface points into surface-elements and that as long as the radius size was reasonable (e.g., 5-20 cm) its relaxation extended to regions with sparse point distribution (at the maximal range limits).

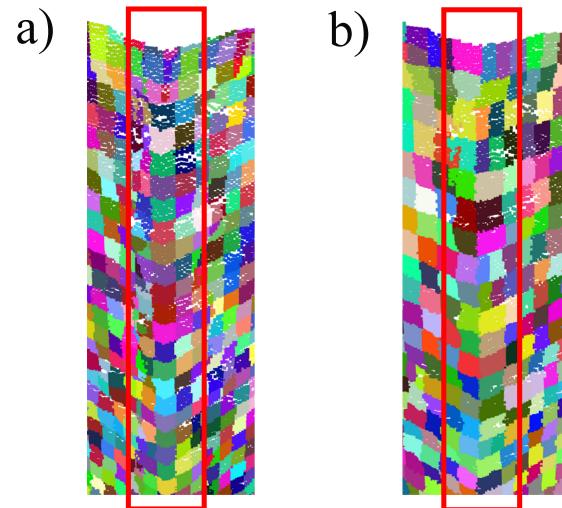


Figure 2: Point cloud partitioning into cellular units along a building corner with voxels (a) compared to ball-tree (b).

2.3 Geometric cue computation

Next, we define means to similarity mensuration between the extracted elemental units. To geometrically characterize them, we use second-order tensor which is defined as the sum of outer products with respect to a reference point (Eq. 1).

$$\mathbf{T} = \frac{1}{n} \sum_i^n (\mathbf{x}_i - \bar{\mathbf{x}}) \otimes (\mathbf{x}_i - \bar{\mathbf{x}}) \quad (1)$$

where \mathbf{x}_i , $\bar{\mathbf{x}}$ are the 3-D points of the elemental unit and its reference point, respectively. Here, the centroid of the elemental unit acts as its reference as well. The equal significance of the points composing the element in its characterization motivates that choice. We analyze the spatial distribution by using the spectral decomposition of \mathbf{T} into its eigenvalues and eigenvectors (λ_i and \mathbf{v}_i , respectively). An axial distribution would yield a triplet of the form $\lambda_1 > 0 \& \lambda_2, \lambda_3 \approx 0$, whereas a coplanar one would yield a set of the form, $\lambda_1, \lambda_2 > 0 \& \lambda_3 \approx 0$. The ratio $\sqrt{\lambda_2/\lambda_1}$ measures the evenness of the distribution and allows to distinguish planar patches from linear entities.

Taking advantage of the fact that a closed-form solution exists for the characteristic polynomial, the eigenvalues are computed as follows (Kopp, 2008):

$$\begin{aligned} \lambda_1 &= q + 2p \cdot \cos \theta \\ \lambda_2 &= q + 2p \cdot \cos(\theta + 120^\circ) \\ \lambda_3 &= 3q - \lambda_1 - \lambda_2 \end{aligned} \quad (2)$$

with:

$$\begin{aligned} q &= \frac{1}{3} \text{tr}(\mathbf{T}) \\ p &= \sqrt{\frac{1}{6} \text{tr}((q \cdot \mathbf{I} - \mathbf{T})^2)} \\ \theta &= \frac{1}{3} \arccos\left(\frac{1}{2} \det(\mathbf{T} - q\mathbf{I})\right) \end{aligned} \quad (3)$$

where \mathbf{I} is the 3×3 identity matrix.

The relevant eigenvectors are either \mathbf{v}_3 for surfaces or \mathbf{v}_1 for linear entities, and are computed by the cross product of two

rows of $\mathbf{T} - \lambda_{i \in \{1,3\}} \mathbf{I}$. A smooth but non-planar surface is characterized by the dominant plane of projection, with \mathbf{v}_3 and $\sqrt{\lambda_3}$ as its normal and an estimate of the deviation from planarity. Similarly, linear features are represented by their axial direction \mathbf{v}_1 with $\sqrt{\lambda_2}$ measuring the deviation from the defined line. Using the described formulation provides a direct approach for the computation of the eigenvalues/vectors which can be easily implemented in a parallel fashion.

2.4 Graph-based clustering

We take a graph-based approach for aggregating the elemental units. Nodes in this representation correspond to the units, edges to connectivity, and their weights measure similarity. We establish connectivity based on k-nearest neighbors (k-nn) querying. This choice has several merits: it helps us bridging gaps, e.g., due to occlusions, and it makes sure that no surface element is disjoint. The k-nn does not enforce dual connectivity, as elements are irregularly distributed, and their density varies. Thus, one-way connectivity is established among the surface elements.

With the Euclidean proximity already embedded in the neighborhood definition, there is no need for further consideration of it in establishing the graph. For planar features, we define similarity in terms of coplanarity and smoothness. Coplanarity is expressed as the distance between the centroid of a surface element to the plane defined by its neighbor (Eq. 4):

$$\varepsilon_{ij}^s = \left\| \mathbf{v}_3^i \mathbf{v}_3^{i^T} (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i) \right\| \quad (4)$$

For smoothness, the norm of the difference between the two surface normals is measured by projecting one normal vector on the other (Eq. 5):

$$\varepsilon_{ij}^s = \left\| (\mathbf{I} - \mathbf{v}_3^i \mathbf{v}_3^{i^T}) \mathbf{v}_3^j \right\| \quad (5)$$

For linear elements, the connectivity is measured in similar fashion by measuring the collinearity and smoothness between them, defined as the distance between the centroid of a linear element to the line defined by its neighbor and as the norm difference between two axial directions, respectively (Eq. 6):

$$\begin{aligned} \varepsilon_{ij}^\ell &= \left\| (\mathbf{I} - \mathbf{v}_1^i \mathbf{v}_1^{i^T}) (\bar{\mathbf{x}}_j - \bar{\mathbf{x}}_i) \right\| \\ \theta_{ij}^\ell &= \left\| (\mathbf{I} - \mathbf{v}_1^i \mathbf{v}_1^{i^T}) \mathbf{v}_1^j \right\| \end{aligned} \quad (6)$$

Both measures have equal contribution and impact. It is customary to apply a threshold to both measures and to weigh them in a logical manner where the edge widths are set in a binary fashion (Vo et al., 2015; Dong et al., 2018). However, we argue that these measures are distributed in χ^2 fashion with only one degree-of-freedom (DOF). The single DOF can be derived from Eqs. (4) – (6), as our measures are defined in terms of distance estimates along or across predetermined axes without respect to orientation. From an algebraic perspective, we note that the rank for all the projection matrices is one. Therefore, both ε_{ij} and θ_{ij} are χ^2 distributed with a single DOF. Our experiments show that this behavior is invariant to shape type and/or complexity (c.f. Sec. 3.1). This implies that the edge weights could be evaluated in terms of confidence intervals of a χ^2 distribution, leading to a probabilistic expression of the threshold, i.e., the connectivity amongst the element units is defined by the probability that both ε_{ij} and θ_{ij} are zero.

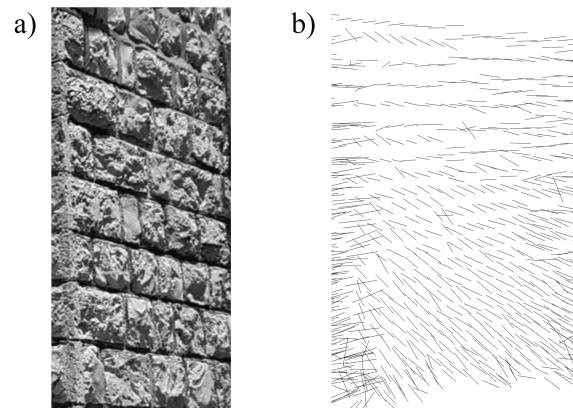


Figure 3: Textured wall (a) and its respective computed normal vectors (b)

Following the connectivity graph construction, an aggregation algorithm is used to extract the core-segments. We use the strongly connected components (SCC; Pearce, 2016) algorithm to establish the clustering, as it requires neither training data nor recursive usage. Moreover, compared to other graph-based algorithms, the computational complexity here is linear and proportional to the number of vertices and edges in the graph, $O(|V| + |E|)$, and in our case $O(k \cdot |V|)$.

2.5 Structural completeness

The established clustering defines core-segments that describe smooth surface patches and curves. Nonetheless, they are incomplete in terms of coverage and/or structure of the actual objects which they represent due to strong deviations in surface normal directions (Fig. 3). To handle these imperfections, we explore the similarity of the core-segments. Doing so requires both the geometric characterization of the core-segments and neighbor definition amongst them. For the latter, we exploit the previously obtained neighborhoods of the elemental units and collapse them to extract the corresponding ones for the core-segments. Notably, the newly established connectivity allows to analyze the similarity between elemental-units that were disjoint purely due to proximity considerations.

Tensors are once again used to geometrically characterize the core-segments. Instead of the naive computation of Eq. (1), we exploit the already computed tensors of the elemental units, to efficiently obtain those for the core-segments. Suppose a core-segment is composed of m elemental units that have their tensors \mathbf{T}_i , which are referred to their own centroids and sizes ($\bar{\mathbf{x}}_i$ and n_i , respectively). The corresponding tensor of the core-segment, \mathbf{T}_\cup , is obtained by:

$$\mathbf{T}_\cup = \frac{1}{N} \sum_i^m [n_i \cdot \mathbf{T}_i + n_i (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_\cup) \otimes (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_\cup)] \quad (7)$$

where $\bar{\mathbf{x}}_\cup$ is the centroid of the core-segment. A detailed proof of the correctness of Eq. (7) is given in the appendix.

We exploit the duality in the geometric characterization of the core-segments to analyze the similarity between neighboring ones. The computation of a unified tensor provides a higher-order representation and the means to cope with texture as the normals are smoothed throughout the core-segments. Moreover, the duality of our characterization allows us recognizing linear objects such as poles, that were considered as planar

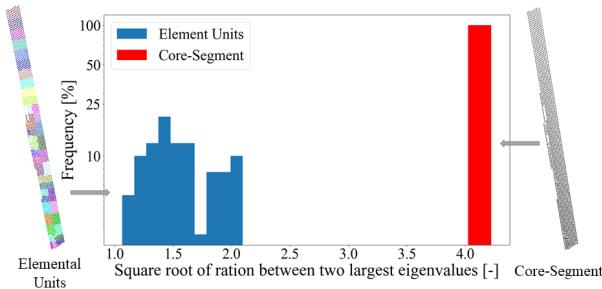


Figure 4: Analysis of object shape: Extracted elemental units compare to and their respective core-segment

patches when analyzed in their elemental unit form Fig. (4). On the other hand, falling back to the elemental analysis allows the detection of curved objects, as they describe a smooth change in the surface normal. Similarity is measured using the aforementioned methodology of Eqs. (4) – (6), and is followed by a second SCC analysis to obtain the segmentation.

Cluttered segments are easily distinguished as ones composed of a couple of surface elements at most. Their corresponding points are re-assigned in a point-wise fashion to the closest segment that is found within a certain distance in terms of either coplanarity or colinearity (depending on the shape of the segment).

To sum, our proposed segmentation is derived as follows: first, the point cloud is arranged with a ball-tree data structure. Next, the elemental units are extracted as equally-sized nodes corresponding to the smallest detectable objects within the scene. Then, the k-nearest neighbors of these units are determined and the connectivity graph is established. Edge weights are set based on similarity mensuration, followed by a SCC analysis to extract the core-segments. To cope with structural fragmentation, a second SCC is performed over the collapsed connectivity graph, where the edge weights are re-evaluated based on the characterization of the established core-segments. Finally, cluttered segments are dissolved, to establish the final segmentation.

3. ANALYSIS

To evaluate the performance of the proposed methodology, the segmentation was applied on two datasets, which were acquired by a Leica C10 terrestrial scanner with an angular resolution of at least 0.025° . While both cases contain occlusions of some magnitude, the scanned scenes are distinguished by their unique characteristics which provide a more thorough examination of the segmentation scheme. The first dataset consists of a building with textured walls, as they are coated with stone-carved tiles. Each tile is $20 \times 30 \text{ cm}^2$ in size and the variations in depth exceed in some case 10 cm. Moreover, the dataset contains low vegetation in proximity to the building faces. The second dataset we tested against included a complex industrial structure. The building is constructed with multiple faces including curved ones. Due to the building's extents ($\sim 70 \text{ m}$ in length and $\sim 15 \text{ m}$ in height), the variation in scanning resolution is greater than the previous one.

3.1 Learning the interactions amongst elemental unit

As previously stated (Sec. 2.4), the pair-wise similarity of elemental-units is distributed in a χ^2 manner with a single

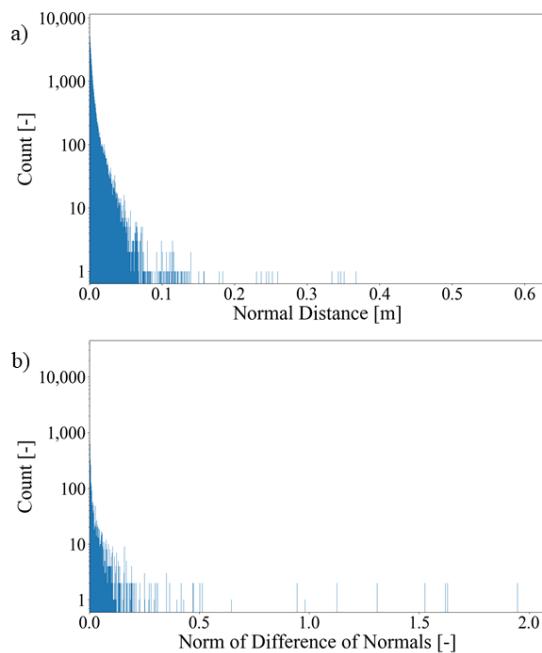


Figure 5: The pairwise similarity distribution between elemental units over various surface types: measured distances (a) and differences of normals (b)

DOF. To verify this behavior, multiple surfaces were extracted from the datasets with different characteristics in terms of orientation, texture, and curviness. Each surface was studied separately by computing its corresponding surface elements and measuring the similarity between neighboring ones in terms of both coplanarity and smoothness. Since all the elements lie on the same surface, the SCC analysis should produce a single segment. This assumption is the basis of our examination and is used to investigate the maximal confidence level to apply while evaluating the connectivity amongst elemental units. Our experiments included more than 40 objects consisting of approximately 60,000 unique elemental unit pairings. For each pair, the distance (Eq. 4) and difference of normals (Eq. 5) were measured, and the distributions of both values were studied over the entire sample set. As expected, the distances behave in χ^2 -like distribution with a mean and standard deviation of 0 cm and 1 cm, respectively and a single DOF. The differences of normals exhibited similar behavior with a mean and standard deviation of 0.01 and 0.05, respectively. In terms of similarity, our experiment shows (Fig. 5) that two elements are to be aggregated with a 99% confidence level if the measured distance and difference of normals are under 8 cm and 0.2 (corresponding to a maximum angular difference of $\sim 20^\circ$), respectively.

3.2 Evaluation

To evaluate the outcome of our algorithm, it is tested both by means of visual inspection, but more importantly via a set of quantitative measures with respect to manually obtained ground-truth labelings. When applied on the first dataset (Fig. 6), our proposed segmentation scheme managed to extract the building faces as complete objects, despite the variations in normal directions due to stone-carved tiling. With respect to the low vegetation, the corresponding points were well separated from those of the building facades. Once again, proving the advantage of using a proximity-based partitioning for the

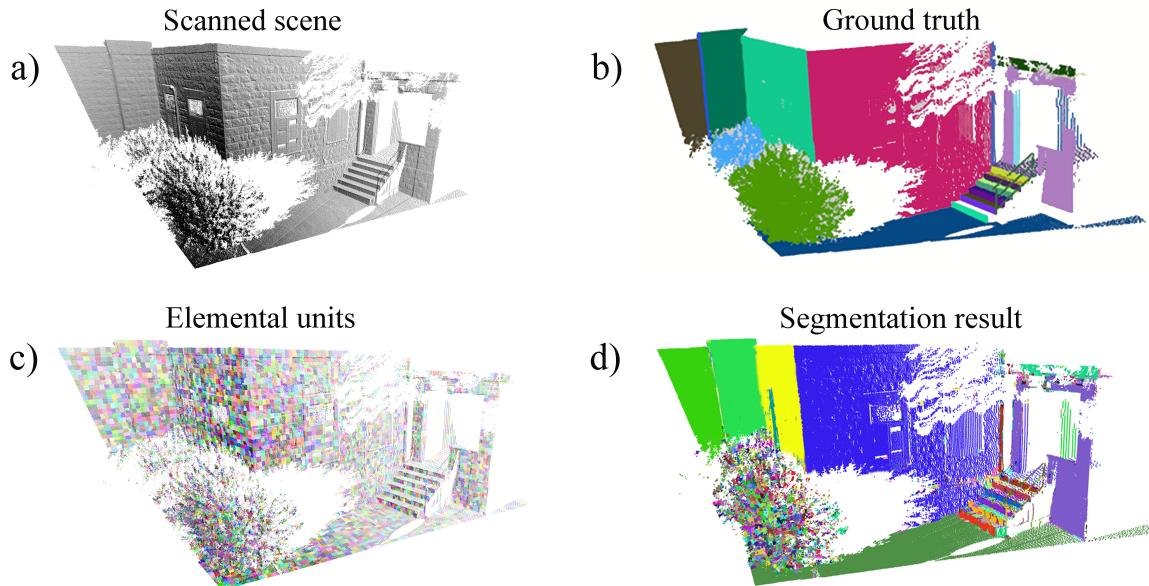


Figure 6: Segmentation results of first dataset. Object colors are chosen randomly

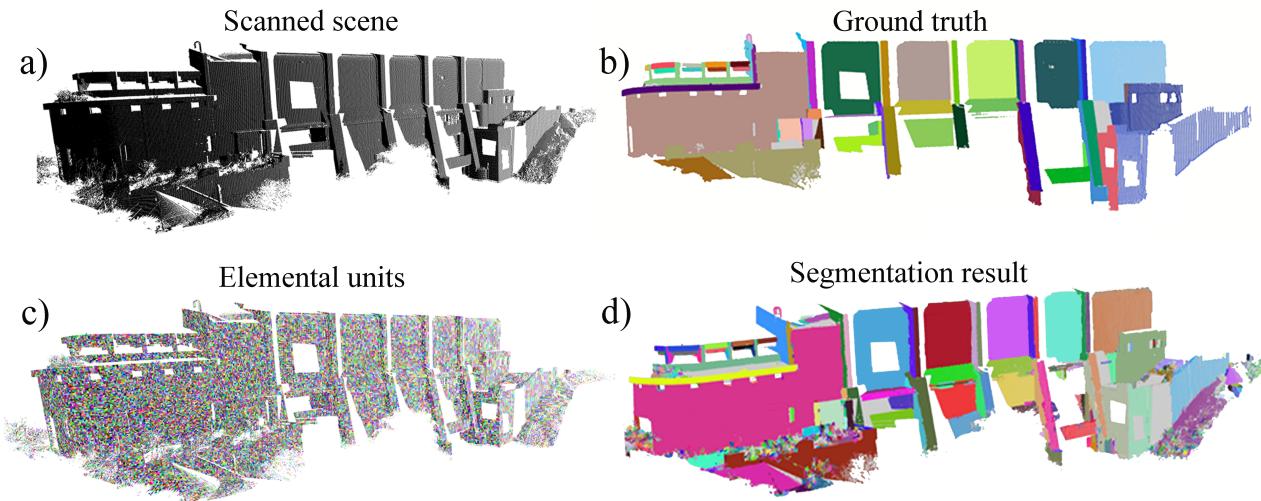


Figure 7: Segmentation results of second dataset. Object colors are chosen randomly

initial over-segmentation. Notably, despite their planar characterization, the extracted elemental-units of the low vegetation do not share any similarity to their neighbors other than Euclidean proximity, and therefore, remained detached from their surroundings. Despite the overall accuracy of the obtained segments, those representing the staircase still remain fragmented. Our investigation indicates that this is mainly due to the resolution at which it was scanned. While the vertical parts have a sufficient amount of points to detect them as individual objects, the horizontal ones are sparsely represented within the point cloud and, therefore, the geometric characterization in these areas is faulty, leading to being initially the respective points assigned to the features they are closest to.

When inspecting the obtained segmentation for the second dataset (Fig. 6), the proposed model was able to extract all the building faces as individual objects despite the curved tops of the vertical walls. The results accurately identify the walls and extract the supporting columns and bars, irrespective of their distance from the scanner. Notice also that the curved wall was

identified as a single entity despite its bending. Here again, low vegetation was separated from the segments independently of their resolution and distance from the scanner or from the surrounding objects.

Quantitative evaluation – several measures are utilized here, including three standard metrics for assessing the object detection and point cloud classification scheme, namely the precision, recall, and a test accuracy measures (termed also F1-score). The quantitative evaluation results are summarized in Table (1), along with an overall performance estimation, indicating the high accuracy of the proposed scheme.

Dataset	No. of points	Precision	Recall	F1
1	258,000	92.6%	88.0%	90.3%
2	364,000	92.4%	91.7%	92.1%
Overall	–	92.5%	89.8%	91.2%

Table 1: Segmentation quantitative evaluation results

Efficiency evaluation – for a complete assessment of the pro-

posed model, its efficiency is studied in terms of run-time performance. The execution times of each step in our algorithm was studied against both datasets. Testing was performed on a i5-3230M CPU with 4 GB RAM executing a *python* implementation. The results are summarized in Table (2) and show that majority of run-time is invested in arranging the data within a ball-tree data structure and the extraction of elemental units. These stages involve reducing the data volume and therefore they are expected to be slower compared to the rest of the methodology. Run-time for both stages would significantly improved when implemented in a compiler based programming language. Note the efficiency of the graph-based clustering, where both its construction and clustering is almost indifferent to data size. Overall, the segmentation for both datasets required less than a minute, which shows the potential of our method.

Dataset	1	2
Ball-tree Construction	23%	36%
Elemental Units Extraction	47%	42%
Connectivity Graph Establishment	7%	5%
Graph-Clustering via SCC	1%	2%
Structural Completeness Analysis	22%	15%
Total Runtime [sec]	18.7	50.6

Table 2: Segmentation performance evaluation results

4. CONCLUSIONS

In this paper, we proposed a segmentation approach that is attentive to the characteristics of laser point clouds; Recognizing that the high point density has no direct bearing on the interpretation of geospatial objects, an approach that is applied to elemental units of ‘physical’ value is proposed. The choice of a range-based tree structure in the form of a ball-tree allows not only to define partitioning according to size and distances, in an invariant form to position and orientation (as opposed to using of actual coordinates) but also to maintain properties of size so that proximity-based queries are performed efficiently. The integration of range and size allows to generate a structure that has a natural ability to handle variations in resolution and occluded areas in the scans while maintaining meaningful units for subsequent computations. By not restricting itself to a designated surface type, our segmentation approach allows us to group together smooth non-planar surfaces and linear entities, not necessarily straight ones. Its adaptation to varying resolutions allows applying it under different scenarios. Future work will include extending our segmentation scheme to better cope with elongated planar objects such as stairs and sidewalk pavements and testing it against other state-of-the-art methods and benchmark datasets.

REFERENCES

Adam, A., Grammatikopoulos, L., Karras, G., Protopapadakis, E., Karantzalos, K., 2019. A semantic 3D point cloud segmentation approach based on optimal view selection for 2d image feature extraction. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 9–14.

Dong, Z., Yang, B., Hu, P., Scherer, S., 2018. An efficient global energy optimization approach for robust 3D

plane segmentation of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 137, 112 - 133.

Fan, S., Huang, N., Fang, P., Zhang, J., 2018. A 3d point cloud segmentation method based on local convexity and dimension features. *2018 Chinese Control And Decision Conference (CCDC)*, 5012–5017.

Green, W. R., Grobler, H., 2015. Normal distribution transform graph-based point cloud segmentation. *2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, 54–59.

Hackel, T., Wegner, J. D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 3(3), 177–184.

Kopp, J., 2008. Efficient numerical diagonalization of Hermitian 3x3 matrices. *International Journal of Modern Physics C*, 19(03), 523–548.

Kumar, N., Zhang, L., Nayar, S., 2008. What is a good nearest neighbors algorithm for finding similar patches in images? *Lecture Notes in Computer Science*, 5303 LNCS(PART 2), 364–378.

Landrieu, L., Raguet, H., Vallet, B., Mallet, C., Weinmann, M., 2017. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132, 102 - 118.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, Y., Wu, B., 2019. Structural segmentation of point clouds with varying density based on multi-size supervoxels. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, 389–396.

Li, Y., Wu, B., Ge, X., 2019. Structural segmentation and classification of mobile laser scanning point clouds with large variations in point density. *ISPRS Journal of Photogrammetry and Remote Sensing*, 153, 151 - 165.

Lin, Y., Wang, C., Zhai, D., Li, W., Li, J., 2018. Toward better boundary preserved supervoxel segmentation for 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143, 39 - 47. ISPRS Journal of Photogrammetry and Remote Sensing Theme Issue “Point Cloud Processing”.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 152 - 165.

Papon, J., Abramov, A., Schoeler, M., Worgotter, F., 2013. Voxel cloud connectivity segmentation - supervoxels for point clouds. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pearce, D. J., 2016. A space-efficient algorithm for finding strongly connected components. *Information Processing Letters*, 116(1), 47–52.

Su, Y.-T., Bethel, J., Hu, S., 2016. Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 113, 59 - 74.

Vo, A.-V., Truong-Hong, L., Laefer, D. F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 104, 88 - 100.

Vosselman, G., Coenen, M., Rottensteiner, F., 2017. Contextual segment-based classification of airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128, 354 - 371.

Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., Liu, H., 2013. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sensing*, 5(2), 584–611.

Xing, X.-F., Mostafavi, M. A., Edwards, G., Sabo, N., 2019. An improved automatic pointwise semantic segmentation of a 3D urban scene from mobile terrestrial and airborne lidar point clouds: a machine learning approach. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Xu, B., Jiang, W., Shan, J., Zhang, J., Li, L., 2016. Investigation on the weighted ransac approaches for building roof plane segmentation from lidar point clouds. *Remote Sensing*, 8(1), 5.

Xu, Y., Yao, W., Tuttas, S., Hoegner, L., Stilla, U., 2018. Unsupervised segmentation of point clouds from buildings using hierarchical clustering based on Gestalt principles. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11), 4270-4286.

Yang, B., Dong, Z., Zhao, G., Dai, W., 2015. Hierarchical extraction of urban objects from mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 99, 45 - 57.

Zolanvari, S. I., Laefer, D. F., 2016. Slicing Method for curved façade and window extraction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 119, 334 - 346.

APPENDIX – TENSOR ARITHMETICS

Given m point subsets, from Eq. (1) the tensor of each subset is defined as:

$$\mathbf{T}_i = \frac{1}{n_i} \sum_j^{n_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i) \otimes (\mathbf{x}_j - \bar{\mathbf{x}}_i) \quad (8)$$

where \mathbf{x}_j is the j -th point of the subset of size n_i , and $\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_j^{n_i} \mathbf{x}_j$ is the the centroid which acts as the reference point for the tensor computation. Similarly, the tensor for the unified set is defined by:

$$\mathbf{T}_U = \frac{1}{N} \sum_j^N (\mathbf{x}_j - \bar{\mathbf{x}}_U) \otimes (\mathbf{x}_j - \bar{\mathbf{x}}_U) \quad (9)$$

with $N = n_1 + n_2 + \dots + n_m$. However, Eq. (9) can also be expressed in terms of the original subsets:

$$\mathbf{T}_U = \frac{1}{N} \sum_i^m \sum_j^{n_i} (\mathbf{x}_j - \bar{\mathbf{x}}_U) \otimes (\mathbf{x}_j - \bar{\mathbf{x}}_U) \quad (10)$$

We note that the nested summation is essentially, the computation of the tensor for the i -th subset, referenced to the centroid of the complete set of points. For simplicity, we denote that expression as \mathbf{T}_i^* , which yields:

$$\mathbf{T}_U = \frac{1}{N} \sum_i^m \mathbf{T}_i^* \quad (11)$$

Our aim is to obtain a relation between \mathbf{T}_i^* to the originally computed tensor, \mathbf{T}_i . To do so, we augment Eq. (11) with the reference point of the original subset:

$$\mathbf{T}_i^* = \sum_j^{n_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i + \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) \otimes (\mathbf{x}_j - \bar{\mathbf{x}}_i + \bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) \quad (12)$$

The RHS of Eq. (12) can be re-written as:

$$\begin{aligned} \mathbf{T}_i^* = & \sum_j^{n_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i) \otimes (\mathbf{x}_j - \bar{\mathbf{x}}_i) + \\ & + \left[\sum_j^{n_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i) \right] \otimes (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) + \\ & + (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) \otimes \left[\sum_j^{n_i} (\mathbf{x}_j - \bar{\mathbf{x}}_i) \right] + \\ & + \sum_j^{n_i} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) \otimes (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) \end{aligned} \quad (13)$$

The first term in Eq. (13) is the original tensor multiplied by the size of the subset, while the second and third ones equal zero from the average definition. Substituting back into Eq. (11), we get:

$$\mathbf{T}_U = \frac{1}{N} \sum_i^m [n_i \cdot \mathbf{T}_i + n_i (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U) \otimes (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_U)] \quad (14)$$

Thus, the tensor computation for a unified pointset is expressed in terms of those of the individual subsets and translations to the centroid of the complete set. For the unification of a pair of tensors, Eq. (14) can be further simplified, as the overall centroid can be expressed as:

$$\begin{aligned} \bar{\mathbf{x}}_U &= \bar{\mathbf{x}}_1 + \frac{n_1}{n_1+n_2} (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \\ \bar{\mathbf{x}}_U &= \bar{\mathbf{x}}_2 - \frac{n_2}{n_1+n_2} (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \end{aligned} \quad (15)$$

Resulting in:

$$\begin{aligned} \mathbf{T}_U = & \frac{n_1}{n_1+n_2} \mathbf{T}_1 + \frac{n_2}{n_1+n_2} \mathbf{T}_2 + \\ & + \frac{n_1^2 \cdot n_2 + n_1 \cdot n_2^2}{(n_1+n_2)^3} (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \otimes (\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \end{aligned} \quad (16)$$

which depends solely on the original subsets and their corresponding tensors, centroids and sizes.