

SEGMENTATION OF LARGE UNSTRUCTURED POINT CLOUDS USING OCTREE-BASED REGION GROWING AND CONDITIONAL RANDOM FIELDS

M. Bassier*, M. Bonduel, B. Van Genechten and M. Vergauwen

Dept. of Civil Engineering, TC Construction - Geomatics
KU Leuven - Faculty of Engineering Technology
Ghent, Belgium
(maarten.bassier, mathias.bonduel, bjorn.vangenechten, maarten.vergauwen)@kuleuven.be

Commission II

KEY WORDS: Segmentation, Region Growing, Conditional Random Fields, Point Clouds

ABSTRACT:

Point cloud segmentation is a crucial step in scene understanding and interpretation. The goal is to decompose the initial data into sets of workable clusters with similar properties. Additionally, it is a key aspect in the automated procedure from point cloud data to BIM. Current approaches typically only segment a single type of primitive such as planes or cylinders. Also, current algorithms suffer from oversegmenting the data and are often sensor or scene dependent.

In this work, a method is presented to automatically segment large unstructured point clouds of buildings. More specifically, the segmentation is formulated as a graph optimisation problem. First, the data is oversegmented with a greedy octree-based region growing method. The growing is conditioned on the segmentation of planes as well as smooth surfaces. Next, the candidate clusters are represented by a Conditional Random Field after which the most likely configuration of candidate clusters is computed given a set of local and contextual features. The experiments prove that the used method is a fast and reliable framework for unstructured point cloud segmentation. Processing speeds up to 40,000 points per second are recorded for the region growing. Additionally, the recall and precision of the graph clustering is approximately 80%. Overall, nearly 22% of oversegmentation is reduced by clustering the data. These clusters will be classified and used as a basis for the reconstruction of BIM models.

1. INTRODUCTION

Point cloud segmentation is a widely discussed topic in the research community. The objective is to associatively cluster points with similar characteristics. Not only does it drastically reduce the amount of data, it also allows for better data interpretation. For instance, Once the point cloud is segmented, it is processed by reasoning frameworks to identify different building elements for the purpose of reconstruction Building Information Models (BIM's) (Bassier et al., 2017). Point cloud segmentation is also used in fields such as computer vision, remote sensing, robotics and other applications that require the partitioning of 3D point cloud data (Volk et al., 2014).

Due to the sheer size of the point cloud information, automated segmentation is becoming increasingly computationally challenging. For instance, point clouds captured by Terrestrial Laser Scanners (TLS) consist of tens of millions of points per scan. Also, the scanned objects are often partly occluded, the data is unevenly distributed due to the range to the sensor and there is noise present in the data set (Tang et al., 2010). These problems prove challenging when developing segmentation algorithms (Nguyen and Le, 2013).

The emphasis of this work is on the segmentation of large unstructured point clouds of buildings. More specifically, we look to identify building object parts for classification purposes. The proposed method is able to properly partition the data even in highly cluttered and noisy environments. Also, our approach is capable of detecting smooth surface including planes and complex geometry.

The remainder of this work is structured as follows. The background is presented in Section 2. In Section 3 the related work is discussed. In Section 3. the methodology is presented. The test design and experimental results are proposed in Section 4. The approach is discussed in Section 5. Finally, the conclusions are presented in Section 6.

2. BACKGROUND & RELATED WORK

Several methods have been proposed for efficient point cloud segmentation including region growing, edge detection, model-based methods, machine learning techniques, graphical models, etc. (Nguyen and Le, 2013; Lin et al., 2015; Fan et al., 2017; Vosselman et al., 2017). In this research, an efficient region growing algorithm is implemented for the segmentation similar to Yau et al. (Yau et al., 2013), Vo et al. (Vo et al., 2015), Habib et al. (Habib and Lin, 2016) and Lin et al. (Lin et al., 2015). They propose algorithms for the segmentation of unstructured point clouds of small-scale objects and urban environments. We expand their approach by parallel processing subsets of the initial point cloud. By doing so, the number of points can be greatly increased. Our method closely aligns with the work of Su et al. (Su et al., 2016). They also perform an initial oversegmentation after which they use graph theory to cluster the segment. While they focus on the processing of cylindrical pipes, we segment both planes as well as complex geometry.

Both structured and unstructured data sets are processed: The former directly exploits the data structure of the used sensor while the latter integrates costly nearest neighbour searches to process point cloud data. While being very efficient, processing structured data is often sensor and case specific (Zhou et al., 2014). For instance, Grant et al. achieve near real-time segmentation

*Corresponding author

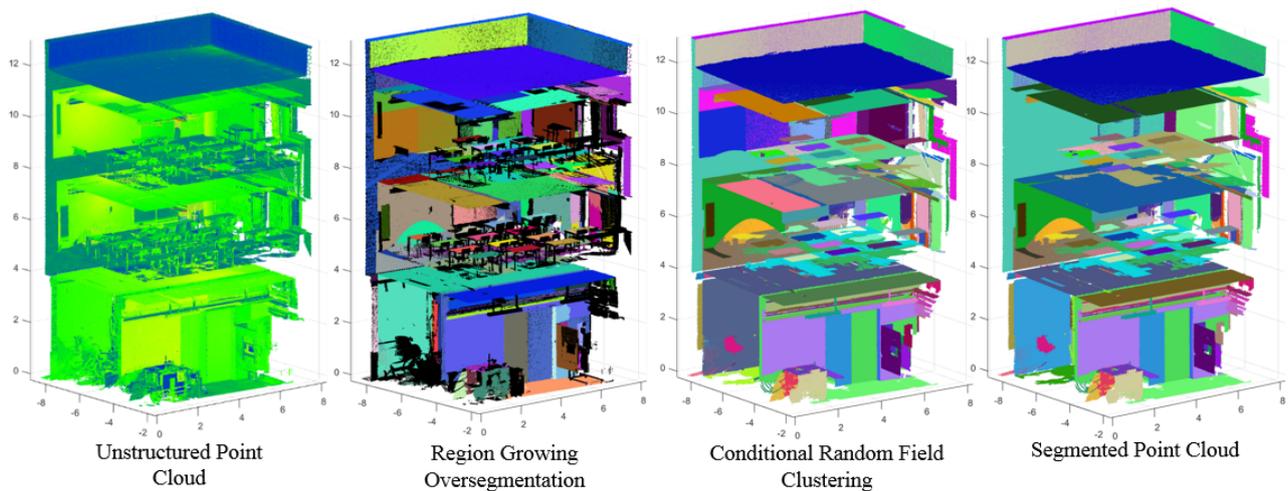


Figure 1. Overview workflow and intermediate results. Initial point cloud (a), oversegmented clusters with non-valid clusters in black (b), graph representation of clusters (c) and the final grouped clusters (d).

of successive laser scans using a Hough-transform implementation (Grant et al., 2013). Holz D. and Behnke S. rapidly segment RGB-D sensor data using region growing (Holz and Behnke, 2012). In contrast, processing unstructured data is more general but requires preprocessing of the data. Typically, the data is restructured into voxel octrees for efficient nearest neighbour searches (Vo et al., 2015; Su et al., 2016). We integrate a similar approach as the emphasis of this research is on the processing of unstructured data in order to be sensor independent.

A promising segmentation approach is the use of graph theory. By representing the inputs as a set of nodes connected by edges, the segmentation can be treated as a graph optimization problem. Popular implementations of graphical models are Conditional Random Fields (Wolf et al., 2015; Xiong et al., 2013; Niemeyer et al., 2012) and Markov Random Fields (Munoz et al., 2009). These probabilistic models solve the segmentation by modelling the posteriori probability of the outputs given a set of feature values. In our approach, we implement a Conditional Random Field for the purpose of clustering our initial segments to enhance the segmentation process.

The goal of our research after segmentation is to process the clustered segments by reasoning frameworks to compute class labels for each cluster. Some researchers consider segmentation and classification as a single step process (Hackel et al., 2016, Landrieu et al., 2017). However, the features extracted from a single point and its neighbourhood typically encode only local characteristics. In our work, segmentation and semantic labelling are treated as individual steps. By reducing the point cloud to a set of clusters, the number of samples is greatly reduced and more global and distinct features can be computed from the groups of points.

3. METHODOLOGY

In this paper, a two-step segmentation algorithm is proposed that associatively clusters points and computes the properties for each group. An overview of the general workflow is depicted in Fig. 1. First, an efficient octree-based region growing method is implemented that rapidly oversegments the data. Next, a clustering algorithm is used based on graph theory to reduce the number of segments created. Both steps are discussed in detail in the following paragraphs.

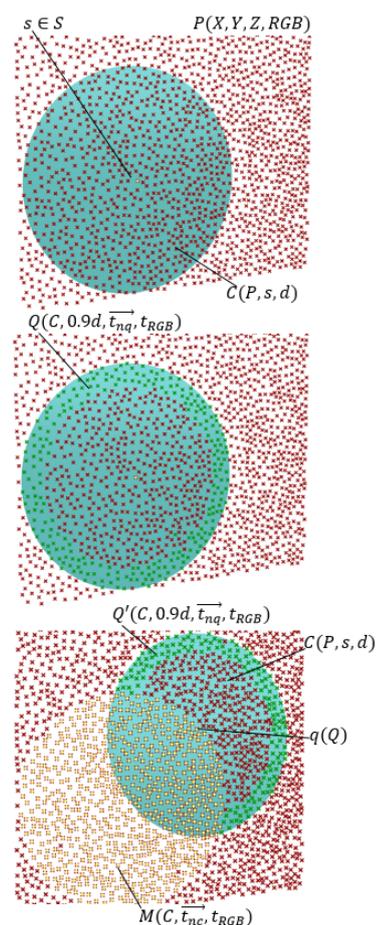


Figure 2. Overview workflow Region Growing. Candidate segment points C in P (top), Extraction of growing points Q from C (mid) and update of Q , M and C during each iteration (bottom).

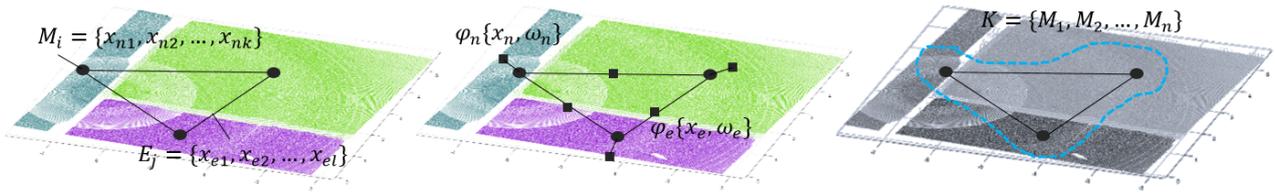


Figure 3. Overview workflow Clustering. Graph representation (left), Conditional Random Field representation (mid) and the clustered segments (right).

Region growing The oversegmentation of the data is performed using an iterative growing algorithm based on the normal and colour information of the point cloud. As input, our algorithm takes any point cloud in a widely accepted format. The initial data is restructured as a voxel octree P for efficient neighbourhood searches. Also, the data is tiled and parallel processed for computational efficiency. Each tile is uniformly seeded $s \in S$ after which segments M are iteratively grown. During every iteration, candidate points C are considered in a dynamic search radius d from s (Fig. 2 top):

$$C = \{p \in P : \|p - s\| < d\} \quad (1)$$

The inliers of M are given by the points in C that meet the following conditions

$$M = \left\{ c \in C : \left(\vec{n}_c \cdot \vec{n}_s \right) > t_{\vec{n}c} \right. \\ \left. \wedge \|RGB(c) - RGB(s)\| < t_{RGB} \right\} \quad (2)$$

where $t_{\vec{n}c}$ and t_{RGB} are respectively thresholds for the normal similarity and the colour similarity of c compared to s . The growing is performed by considering new points in M as the new seeds for the segment. To make the algorithm more efficient, only a subset of C is used for the growing (Fig. 2 mid). A set of candidate growing points Q is defined along the border of C that are conditioned on

$$Q = \left\{ c \in C : \|c - s\| > 0.9d \wedge \left(\vec{n}_c \cdot \vec{n}_s \right) > t_{\vec{n}q} \right. \\ \left. \wedge \|RGB(c) - RGB(s)\| < t_{RGB} \right\} \quad (3)$$

where d is the distance from c to s and the normal threshold $t_{\vec{n}q}$ is more strict than $t_{\vec{n}c}$ to ensure good growing candidates. During the growing stage, the members of Q are iteratively used to seek new growing points Q' and candidates C until Q is empty. For each $q \in Q$ the following steps are performed (Fig. 2 bottom)

$$C = \{p \in P \setminus M : \|p - q\| < d\} \\ Q' = \left\{ c \in C : \|c - q\| > 0.9d \wedge \left(\vec{n}_c \cdot \vec{n}_q \right) > t_{\vec{n}q} \right. \\ \left. \wedge \|RGB(c) - RGB(q)\| < t_{\vec{n}q} \right\} \quad (4) \\ M' = \left\{ c \in C : \left(\vec{n}_c \cdot \vec{n}_q \right) > t_{\vec{n}c} \right. \\ \left. \wedge \|RGB(c) - RGB(q)\| < t_{\vec{n}c} \right\} \\ M = M \cup M' \\ Q = Q \cup Q' \setminus \{Q \in C\}$$

where \vec{n}_q and $RGB(q)$ are respectively the mean normal and mean colour of the nearby points of c in M . Potential segments are only accepted when containing sufficient points based on a threshold. Both the seeds and candidate groups are treated efficiently so no unnecessary neighbourhood searches are performed. The result is a set of segments M that contain points with similar properties.

Clustering Once the data is oversegmented, the individual segments are processed by an associative classification framework to enhance the clustering. More specifically, the clustering of the segments is considered a binary graph optimization problem of a Conditional Random Field (CRF) (Sutton and McCallum, 2011). Binary class labels $y \in \zeta = \{0, 1\}$ are computed for each observed segment to determine whether or not a segment is willing to cluster with its neighbours. A graph $G = (M, E)$ is constructed whose nodes M are the segments of the initial region growing and a set of edges E , derived from the adjacency matrix that connects each node to its four nearest neighbours. As a criterion for adjacent neighbours, the Euclidean distance between the boundaries of $u \in M_i$ and $v \in M_j$ is observed. A set of local features $x_n = \{x_{n1}, x_{n2}, \dots, x_{nk}\}$ is computed for each node M in G including the surface type and size (Fig. 3 left). Additionally, contextual features $x_e = \{x_{e1}, x_{e2}, \dots, x_{el}\}$ are extracted based on the edges such as surface type similarity, coplanarity and the distance between the boundaries of the nodes. Given the graph and the local and contextual features, the unary and pairwise potentials are initialised (Fig. 3 mid). The conditional probability $p(\mathbf{y}|\mathbf{x})$ of the class labels $y \in \zeta = \{0, 1\}$ is given by (Eq. 5).

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \left(\sum_n \phi_n(x_n, \omega_n) + \sum_e \phi_e(x_e, \omega_e) \right) \quad (5)$$

where Z is the partitioning function that normalises the input parameters, $\omega = \{\omega_n, \omega_e\}$ the pre-trained node and edge weights and ϕ_n, ϕ_e the log-linear potentials given the feature vectors. The final class labels are computed by minimising the negative log-likelihood in the Conditional Random Field (Eq. 6) given the feature vectors \mathbf{x} and ω . However, exact inference is impossible in a densely connected CRF. Therefore, we approximate the log-likelihood using a loopy belief propagation method (Frey and MacKay, 1998) as implemented by Schmidt (Schmidt, 2010). After marginal inference is computed, each cluster is assigned the label with the highest probability. The result is a set of labelled 3D point clusters.

$$\mathbf{y} = \underset{y \in \zeta}{\operatorname{argmin}} P(\mathbf{y}|\mathbf{x}, \omega) \quad (6)$$

Once the class labels are computed, the segments that are labelled as potential clustering candidates are grouped together based on (Fig. 3 right). Iteratively, the most likely connected clustering candidates are merged given their edge potentials $\phi_e(x_e, \omega_e)$. The result is a set of clustered segments with similar properties.

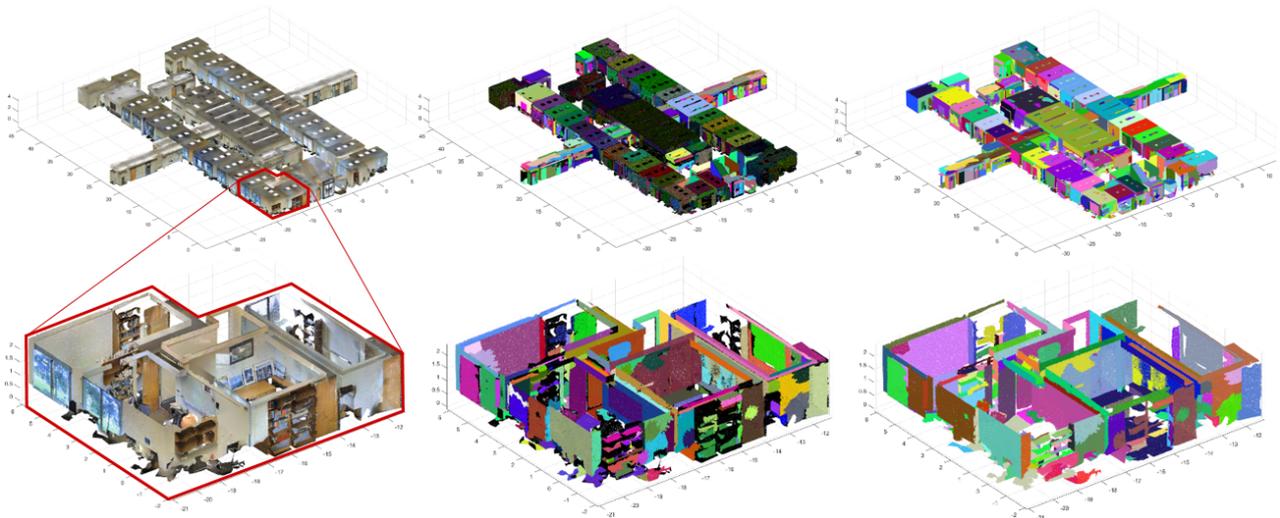


Figure 4. Overview experiments Stanford 2D-3D-Semantics Dataset (2D-3D-S): The top and bottom row respectively depict segmentation and classification results for Area1 and a subset of Area1 for clarification. From left to right can be viewed: Initial point cloud (left), oversegmented clusters with non-valid clusters in black (mid) and the final grouped clusters (right).

Table 1. Segmentation results of the Stanford 2D-3D-Semantics Dataset (2D-3D-S)

Data set	3D Points	Area [m ²]	Time [s]	Segments
Area1	44,026,810	965	1100	2968
Area2	47,294,447	1100	1091	2883
Area3	18,662,173	450	490	1216
Area4	43,682,507	870	1043	2810

Table 2. Clustering results of the Stanford 2D-3D-Semantics Dataset (2D-3D-S)

Data set	Segments	Clusters	Reduction[%]	Time [s]
Area1	2968	2301	22.5	280
Area2	2883	2185	24.2	267
Area3	1216	450	19.2	93
Area4	2810	2147	23.6	221

4. EXPERIMENTS

The algorithm was trained and tested on different data sets. The candidate data was not de-noised and a wide variety of objects was present in the point clouds (Fig. 1 and 4). All the input data was stored in binary file formats for efficient I/O. In addition to our own data, the 2D-3D-Semantics (2D-3D-S) benchmark data of Stanford was evaluated (Armeni et al., 2017) (Fig. 4). Areas 1 to 4 were used for the testing with more than 18-50 million 3D points each. All data was processed on a laptop with an Intel Core i7-4900MQ CPU @ 2.8Ghz with 4 cores, 7 active threads and 32 GB RAM. The algorithm was implemented in Matlab 2017 and uses the parallel computing toolbox, the Computer vision toolbox and the UGM toolbox (Schmidt, 2010).

Segmentation The segmentation results are shown in table 1 and Fig. 4. By default, the tile size was set to 1 million 3D points. This fairly small tile size was chosen to avoid idle cores on the CPU as some tiles take more time to process due to the complexity of the geometry. The dynamic search radius for the nearest neighbours was initialised at 0.1m, the angular thresholds $t_{nc} = 45^\circ$ and $t_{nq} = 15^\circ$ and the colour threshold $t_{RGB} = 20$. Additionally, a minimal point count of 2000 points was defined for potential segments. The test results show promising results with an approximate processing speed of 40,000 points/s including I/O, normal computation and segmentation. By parallel processing multiple tiles, the segmentation speed is multiplied by the number of available threads which speeds up the process significantly. On average, 1 cluster is created for every 15,000 points. Most clusters are properly found as shown in Fig. 4. The surface types other than the planar segments showed increased oversegmentation and misclustering. This is expected given the fact that these objects suffer greatly from occlusions and noise. Additionally, the data was oversegmented due to feature variance and because of the data tiling. Also, nearly 30% of the data was unfit

to cluster due to small clusters sizes, noise and clutter such as vegetation.

Clustering The clustering results are shown in table 2 and Fig. 4. The node and edge features were standardised for generalisation purposes. Two node features were used encoding the size and the surface type of each segment. Three edge features were introduced that represented the coplanarity, the distance between boundaries and the surface type similarity. The weights of the model were trained by minimizing the negative log-likelihood given the feature vectors \mathbf{x} and the labels \mathbf{y} of known observations. In total, 3000 surfaces of an office building were used to train the weights (Fig. 1). In order to improve the performance of the model, an equal number of observations of each class was observed. Additionally, a regularization parameter λ was introduced that penalises deviating weights.

The model was tested on the Stanford 2D-3D-Semantics Dataset (2D-3D-S). The performance is depicted in Fig. 5. The average recall and precision is 78% and 79% respectively. This is very accurate given the large feature variance and the configuration of the observed segments. However, a portion of the inliers are easily found due to the data tiling. Planar surfaces belonging to different tiles will cluster more easily as they initially belonged to the same object. Overall, an average reduction of 22.4% was recorded for the clustering of the segments. The mean time for the feature extraction of each cluster and computing inference in the graph is 80 milliseconds per segment. The experiments revealed that the majority of clustered components are of the planar surface type. This is expected due to the used features and the characteristics of these objects. However, several pipe segments properly clustered due to surface similarity and connectivity.

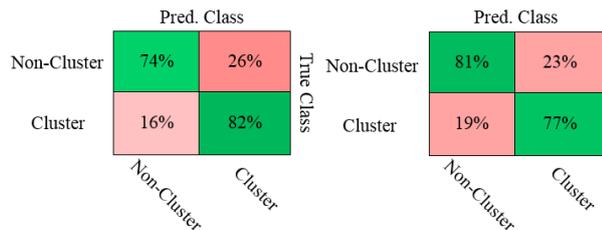


Figure 5. Overview classification Stanford 2D-3D-Semantics Dataset (2D-3D-S): The averaged recall matrix (left) and the averaged precision matrix (right).

5. DISCUSSION

Point cloud segmentation can be performed with numerous 2D and 3D algorithms. Most methods are capable of finding planar or other primitives in a scene. However, there are few methods that deal with large unstructured data. A promising approach is data tiling which separates the initial point cloud into more workable chunks and allows for efficient parallel processing. As in our work, the segmentation process can be sped up nearly equal to the number of available threads. This approach can be further extended towards GPU integration.

A key issue in point cloud segmentation is oversegmentation. While other approaches typically propose heuristics to cluster the individual segments, we tackle this problem using graph theory. By considering the segments as a Conditional Random Field, a probabilistic classification is made to determine which nodes should be clustered. A major advantage is the associative behaviour of neighbouring clusters which leads to more accurate classification. Also, the model parameters are learned from extensive training data which allows better generalisation.

6. CONCLUSION

This paper presents an unsupervised method to segment large unstructured point clouds of buildings. The segmentation is considered as an optimisation problem that can be solved with graph theory. First, the data is oversegmented into planes and complex geometry. A greedy octree-based region growing method is implemented to associatively group points. For efficiency purposes, the data is tiled and parallel processed on multiple CPU threads. Next, the candidate segments are represented by a Conditional Random Field (CRF) with both local and contextual features to determine the most likely clustering candidates. The result of the method is a set of segmented clusters of the initial point cloud data.

The experiments prove that the used method is a fast and reliable segmentation framework for unstructured point cloud data. Segmentation speeds up to 40,000 points per second are recorded for the region growing. Additionally, the recall and precision of graph clustering is approximately 80%. Overall, it can be stated that the used algorithm shows promising results for point cloud segmentation.

In future work, the presented approach will be investigated further to reduce oversegmentation and have better results on surface types other than planar segments. Also, GPU integration will be considered to further speed up the process. The use of Conditional Random Fields for the clustering of the initial segments will be extended to yield better clustering results and to be applicable to other research topics such as classification and reconstruction of BIM objects from point cloud data.

7. ACKNOWLEDGEMENTS

The research group Geomatics of the Department of Civil Engineering, TC Construction at the KU Leuven in Belgium is gratefully acknowledged.

REFERENCES

- Armeni, I., Sax, S., Zamir, A. R. and Savarese, S., 2017. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *arXiv:1702.01105 [cs.CV]*.
- Bassier, M., Vergauwen, M. and Genechten, B. V., 2017. Automated Classification of Heritage Buildings for As-Built BIM using Machine Learning Techniques. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV*(September), pp. 25-30.
- Fan, Y., Wang, M., Geng, N., He, D., Chang, J. and Zhang, J. J., 2017. A self-adaptive segmentation method for a point cloud. *Visual Computer* pp. 1-15.
- Frey, B. and MacKay, D. J. C., 1998. A Revolution: Belief Propagation in Graphs With Cycles. *Advances in neural information processing systems*.
- Grant, W. S., Voorhies, R. C. and Itti, L., 2013. Finding planes in LiDAR point clouds for real-time registration. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 4347-4354.
- Habib, A. and Lin, Y. J., 2016. Multi-class simultaneous adaptive segmentation and quality control of point cloud data. *Remote Sensing* 8(2), pp. 1-23.
- Hackel, T., Wegner, J. D. and Schindler, K., 2016. Fast Semantic Segmentation of 3D Point Clouds With Strongly Varying Density. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences* 3.
- Holz, D. and Behnke, S., 2012. Fast Range Image Segmentation and Smoothing using Approximate Surface Reconstruction and Region Growing. *Intelligent autonomous systems 12 (2013)* pp. 61-73.
- Landrieu, L., Mallet, C. and Weinmann, M., 2017. Comparison of belief propagation and graph-cut approaches for contextual classification of 3D lidar point cloud data. *IGARSS'2017*.
- Lin, Y., Wang, C., Cheng, J., Chen, B., Jia, F., Chen, Z. and Li, J., 2015. Line segment extraction for large scale unorganized point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 102(March), pp. 172-183.
- Munoz, D., Vandapel, N. and Hebert, M., 2009. Onboard Contextual Classification of 3-D Point Clouds with Learned High-order Markov Random. *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. IEEE*.
- Nguyen, A. and Le, B., 2013. 3D Point Cloud Segmentation : A survey. *6th IEEE Conference on Robotics, Automation and Mechatronics* pp. 225-230.
- Niemeyer, J., Rottensteiner, F. and Soergel, U., 2012. CONDITIONAL RANDOM FIELDS FOR LIDAR POINT CLOUD CLASSIFICATION IN COMPLEX URBAN AREAS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences I*(September), pp. 263-268.
- Schmidt, M., 2010. *Graphical Model Structure Learning with l1-regularization*.

- Su, Y. T., Bethel, J. and Hu, S., 2016. Octree-based segmentation for terrestrial LiDAR point cloud data in industrial applications. *ISPRS Journal of Photogrammetry and Remote Sensing* 113, pp. 59-74.
- Sutton, C. and McCallum, A., 2011. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning* 4(4), pp. 267-373.
- Tang, P., Huber, D., Akinci, B., Lipman, R. and Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction* 19(7), pp. 829-843.
- Vo, A. V., Truong-Hong, L., Laefer, D. F. and Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing* 104, pp. 88-100.
- Volk, R., Stengel, J. and Schultmann, F., 2014. Building Information Modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction* 38, pp. 109-127.
- Vosselman, G., Coenen, M. and Rottensteiner, F., 2017. Contextual segment-based classification of airborne laser scanner data. *ISPRS Journal of Photogrammetry and Remote Sensing* 128, pp. 354-371.
- Wolf, D., Prankl, J. and Vincze, M., 2015. Fast Semantic Segmentation of 3D Point Clouds using a Dense CRF with Learned Parameters. *IEEE International Conference on Robotics and Automation (ICRA)*.
- Xiong, X., Adan, A., Akinci, B. and Huber, D., 2013. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction* 31, pp. 325-337.
- Yau, H.-T., Yang, T.-J. and Jian, H.-Z., 2013. A region-growing algorithm using parallel computing for surface reconstruction from unorganized points. *Advances in Engineering Software* 59, pp. 29-37.
- Zhou, Y., Wang, D., Xie, X., Ren, Y., Li, G., Deng, Y. and Wang, Z., 2014. A fast and accurate segmentation method for ordered LiDAR point cloud of large-scale scenes. *IEEE Geoscience and Remote Sensing Letters* 11(11), pp. 1981-1985.