POINT CLOUD SEGMENTATION USING HIERARCHICAL TREE FOR ARCHITECTURAL MODELS

Omair Hassaan, Abeera Shamail, Zain Butt, Murtaza Taj

Department of Computer Science.
LUMS School of Science and Engineering. Lahore, Pakistan

ABSTRACT

Over the past few years, gathering massive volume of 3D data has become straightforward due to the proliferation of laser scanners and acquisition devices. Segmentation of such large data into meaningful segments, however, remains a challenge. Raw scans usually have missing data and varying density. In this work, we present a simple yet effective method to semantically decompose and reconstruct 3D models from point clouds. Using a hierarchical tree approach, we segment and reconstruct planar as well as non-planar scenes in an outdoor environment. This tree uses an exclusive energy function and a 3D convolutional neural network, HollowNets, to classify the segments. We test the efficacy of our proposed approach on a variety of complex real and synthetic data samples, obtaining an improvement of 7.9% in mean IOU over the state of the art approaches.

1. INTRODUCTION

The increasing use of terrestrial and aerial laser scanning along with photogrammetry has enabled mapping of large-scale outdoor scenes and has paved way for emerging applications such as drone-based delivery of goods and self-driving cars that require detailed 3D understanding of large-scale outdoor scenarios.

Armeni [1] et al. segments a scene using strong geometric priors for space estimation and then performs spatial parsing to segment known structures. However, their work is confined to indoor scenes only. Martinovic [2] proposes an approach that combines image based classification with 3D object classification for point cloud segmentation. Such approaches are not applicable to point cloud data in general as a dense set of 2D photographs is not always available. Similarly, hierarchical semantic segmentation [3] is based on learning a Merge Classifier that predicts whether a combination of segments belongs to the same object instance or not. This is a bottom-up approach which suffers from combinatorial complexity.

Lin et al. [4] uses LiDAR data on low-rise houses using planar primitives, patches and symmetric blocks to segment a point cloud. This approach is confined to symmetric houses and planar surfaces and therefore, can not be generalized. Similarly, RAPter [5] exploits regular arrangement of planes

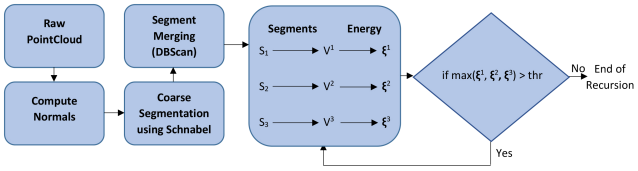


Fig. 1. Block diagram showing the whole process of our approach.

to obtain multiple planar segments from a point cloud. Bajwa et al. [6] proposes an interactive coarse-to-fine segmentation approach using three fundamental Manhattan World constraints. In their work, the projection applied on each point cloud segment is selected manually.

Random Sample Consensus (RANSAC) [7] randomly draws minimal data points to construct shape primitives and determines the best fit. Although RANSAC based algorithms generally perform well, they fall short on complex structures e.g architectural models. All these segmentation approaches are generalized for urban structures and they do not produce meaningful results on complex architectural scenes. Our work, however, uses a hybrid approach which includes model fitting, Manhattan world based projection sequences and a machine learning algorithm to generate a more accurate mix than the existing segmentation techniques. We focus on 3D outdoor scenes with no restrictions towards planar surfaces or geometric buildings.

2. HIERARCHICAL TREE MODEL

In detection, exhaustive scanning of an entire data is computationally infeasible for large-scale scenes while space based partitioning such as Octree is prone to incorrect splitting of objects. We, instead, perform segmentation by recursively partitioning the points in a subspace spanned by the projection of the points into a 1D signal.

The input to our algorithm is coarse segments obtained using RANSAC based primitive fitting [8] and spatial clustering. Our partitioning space is inspired by the work of Bajwa et al. [6] that employs the Manhattan world assumption and projects the 3D data into one or more orthogonal planes.

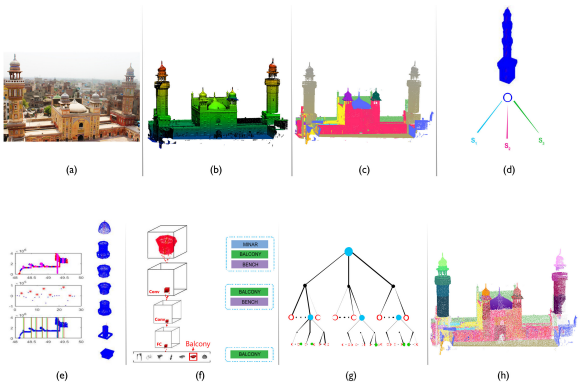


Fig. 2. Flow diagram of the proposed approach. (a) Sample Image of the site. (b) Input point cloud. (c) Coarse segments obtained using RANSAC [8] and spatial clustering (DBScan [9]). (d) Projection sequence. (e) Peak finding on profile curve of minar showing 7 peaks as asterix and vertical line. (f) Classification of one of the segments. (g) Tree update as more segments being added. (h) Obtained segments.

They proposed a semi-interactive approach in which the object class information was manually provided for each of the point cloud segments. The correct projection was then applied based on heuristics for each object category.

On the other hand, we have used three generalized projection sequences each of which results in a sub-segmentation of the point cloud data. Hierarchical organization of structural elements at different scales and locations are commonly seen in man-made structures [10]. Contrary to Lin's [4] bottom-up hierarchical tree of planar patches, our's is a top-down model of hierarchical segmentation. We recursively build a tree $G\{V, E\}$ where the edge E is one of the projection sequences and the nodes V are the obtained sub-segments. The weight of each of these edges is computed using an objective function $\xi(v^n_{i,j}, v^m_{i-1,k})$ defined as:

$$\xi(v^n_{i,j}, v^m_{i-1,k}) = \omega^T \epsilon, \tag{1}$$

where $\omega = \{\omega_1, \cdots, \omega_5\}$ are the weights for each of the five energy terms $\epsilon = \{e_1 \cdots, e_5\}$. These weights are obtained using simple linear regression. The node $v^n_{i,j}$ is the j^{th} segment of the i^{th} iteration (tree depth) obtained via n^{th} projection sequence such that $v^n_{i,j} \subset v^m_{i-1,k}$ and $v^m_{i-1,k}$ is the parent node of v_i. Also $v^m_{i-1,k} = \{\cup v^n_{i,j}\}$ where, for all segments s, $j \in \{1, 2, \ldots, s\}$, $n \in \{1, 2, 3\}$. The set V^n_i contains all segments of $v^m_{i-1,k}$ obtained via n^{th} projection sequence. Each of the projection sequences is discussed next and energy terms are discussed in Section 2.2.

2.1. Projection sequences

We receive coarse segments from the Schnabel's algorithm [8], which are then clustered based on spatial proximity. In the next step, the tree methodology is applied on each coarse segment. As shown in the Fig. 3, on each segment, three different projection sequences are applied. Peak finding $\rho(.)$ is then applied on the obtained low-dimensional signal to obtain the segments. The input point cloud is converted into low-dimensional signal using four projections namely vertical (v_1&v_2), horizontal (h), circular profile (p) and circular un-warping (u).

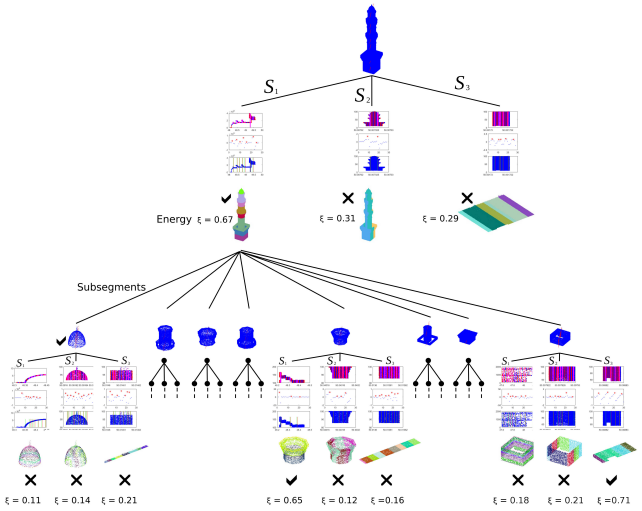


Fig. 3. Tree visualization when segmentation is applied on a minar. The Minar is segmented using all three projection sequences and then the 8 segments produced by S_1 are selected based on energy. Each of them are then re-segmented.

The projections v_1 and h involves eliminating one of the dimensions. Circular and n-gonal RANSAC $\psi(.)$ are then applied on the projected data to recover the location, position and orientation. Since peak finding $\rho(.)$ can return multiple peaks belonging to the same segmentation boundary, operation v_2 is first applied to convert the data into a limited bin histogram. The optimal number of bins differs for each point cloud and is computed by counting the zero-crossings in the derivative signal. Finally, peak finding $\rho(.)$ is performed to obtain the segments. As shown in Fig. 4

Sequence S_1: $V^1_i = \rho(v_2(p(\psi(v_1(v^m_{i-1,k})))))$

Sequence S_2: $V^2_i = \rho(v_2(h(v^m_{i-1,k})))$

Sequence S_3: $V^3_i = \rho(v_2(u(\psi(v_1(v^m_{i-1,k})))))$

All three projection sequences are applied recursively on each of the obtained sub-segments. Each projection sequence generates a set of sub-segments. This is a greedy approach and only the set of sub-segments having the highest weighting edge is selected for further segmentation. The remaining two segment sets are discarded; if the energy of all the segment sets is below a specified threshold the node $v^m_{i-1,k}$ is declared as a leaf node. Once the required recursion depth is reached, all the segments $v^n_{i,j}$ in the set having the highest energy are added as leaf nodes.

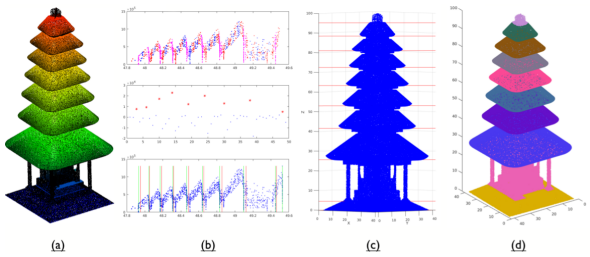


Fig. 4. Detailed modeling via profile curve. (a) Point cloud. (b) Peak finding on profile curve. (c) Peak locations on point cloud. (d) Detailed segmentation.

2.2. Objective function

Since the goal of projection sequence is to simplify the data thus facilitating the segmentation, an incorrect projection sequence, e.g. a circular unwrap of an archway or bridge, would have adverse effects on the data. We estimate the information content and goodness of the obtained segment set based on 5 criteria. Hence, the energy function $\xi(.)$ that we use contains 5 terms based on these criterion:

- Correct segmentation will have more or less uniform distribution of points among segments. In order to avoid skewed distribution of points among segments, the normalized deviation between the segment population ϵ_1 is computed and is defined as:

$$\epsilon_1 = 1 - \frac{\sigma([N_1, N_2, \ldots, N_s])}{\sigma([1, N_p - 1])}. \quad (2)$$

- Correct segments can only be obtained from a segment having considerably large number of points. Hence, the parent node population score ϵ_2 is defined as the ratio between number of points N_p and the set threshold on N_{min}. This energy is maximum in case $N_p \geq N_{min}$ and is defined as:

$$\epsilon_2 = \frac{\min(N_p, N_{min})}{N_{min}}. \quad (3)$$

- Correct segmentation will neither produce a large number of segments nor will it give a single segment only. Segmentation resulting in only a single segment is penalized using a Gaussian distribution having $(\mu, \sigma) = (1, 1)$ and the resulting energy ϵ_3 is defined as:

$$\epsilon_3 = 1 - \sim \mathcal{N}(s|\mu, \sigma^2). \quad (4)$$

- Some projection sequences will always be inapplicable on certain object categories. To introduce a semantic relationship between nodes, ϵ_4 incorporates a prior probability of observing a certain projection sequence given the class information of the initial segment and is defined as:

$$\epsilon_4 = W(r_{ID}(v^m_{i-1,k}), seq.ID), \quad (5)$$

where W is a $K \times 3$ matrix containing prior probabilities of each path for each object class.

- Each sub-segment of a correctly segmented point cloud will have a higher recognition score. The last energy term ϵ_5, is thus based on the classification score r_{scr} of each segment and is defined as:

$$\epsilon_5 = \frac{1}{s} \sum r_{scr}(v^n_{i,j}),\ j \in \{1, 2, \ldots, s\}. \quad (6)$$

2.3. Deep Learning: HollowNet

Unlike 2D images, a point cloud has irregular dimensions in 3D space. Thus, in order to use it in machine learning algorithms we sub-sample data into a regular grid. Seminal work in 3D object recognition such as VoxNet[11] and ShapeNet[12] uses a volumetric representation of objects, instead of PointNet[13], LPCCNet[14] which uses a fixed set of N points.

Voxel Representation: We scale each point cloud object to our voxel size and fit the point cloud inside a 3D cube thus mapping each (x, y, z) point location to (i, j, k) index of a 3D regular grid. Laser scanners provide surface representation of surrounding objects only, thus, we call our voxel representation as Hollow voxel.

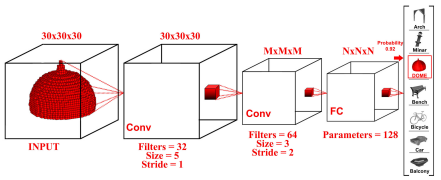
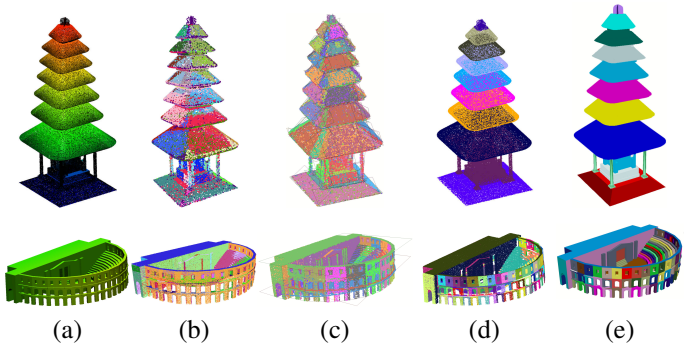


Fig. 5. Layered architecture used to train HollowNet.

CNN: The input to our neural network is a voxel volume of size $L \times W \times H$, where L is the length, W is the width and H is the height of voxel data. Here $H = L = W = 30$. The prediction problem requires producing a target output of size K which is the number of classes we have used to train our network. For our problem we have chosen $K = 7$ each having 300 and 50 training and testing samples, respectively collected from 3D Warehouse and ModelNet10 [12]. We obtained an accuracy of 99.03% on 1000 EPOCHS at a learning rate of $\eta = 0.001$.

Each cross-section of this representation can be considered as an output of convolution by an edge filter making it inherently suitable for commonly used deep convolution neural networks (CNN). We scale the values of voxel between $[-1, 5]$ allowing the neural network to learn discriminative features of binary data as suggested in [11].

Layered Architecture: Contrary to existing volumetric voxel-based CNN, learning on hollow-voxels can be performed on a much simpler network architecture due to their inherent gradient like representation. Our model (see Fig. 5(b)) consists of 2 convolutional layers, each performing 3D



(a) (b) (c) (d) (e)

Fig. 6. Comparisons. (a) Point cloud. (b) RANSAC. [8] (c) RAPter. [5], (d) Ours. (e) ground truth.

strided convolution with a bank of 5^3 dimensional filters, starting with 32 filters and then doubling in the next layer. Each layer uses leaky rectified linear units (Leaky ReLUs) as a non-linearity. This neural network consists of 2 fully connected layers. For the second fully connected layer, the number of neurons is equal to number of desired output classes i.e. K= 7.

3. RESULTS

3.1. Experimental Setup

We evaluated our algorithm both on synthetic as well as real data. The synthetic scenes, taken from 3D warehouse, are a Temple and Roman building. For real data, three sites were scanned, namely, Derawar Fort, Masjid Wazir Khan and Masjid Khudabad using Leica Scan Station P20 Terrestrial Laser Scanner. Our algorithm divides the temple into 10 segments. Out of 79 primitives of the Roman building, our algorithm produces 77 primitives (Fig. 6(d)). Masjid Wazir khan contained 12 domes, 4 balconies, 6 minartets, 14 main arches and 32 small arches. Our algorithm recognized and classified 12 domes, 6 balconies, 4 minars and 13 arches (Fig. 7). The details of these sites and obtained results are shown in Table 1.

3.2. Comparisons

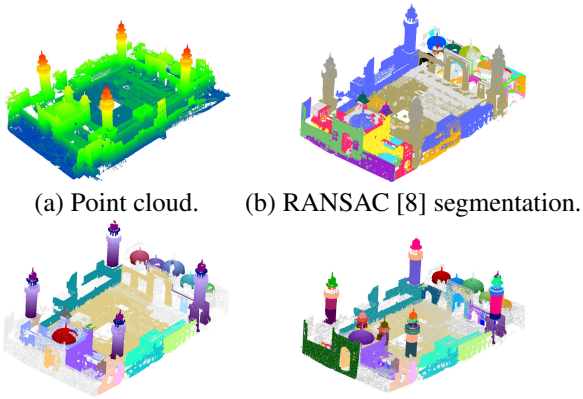
We have compared our hierarchical segmentation algorithm with both plane fitting as well as primitive fitting technique. For plane fitting we have compared with the recently published approach using regular arrangement of planes (RAPter) [5]. For primitive fitting technique, we compared with the seminal RANSAC based approach Schnabel[8]. RAPter achieves high accuracy while reconstructing scenes as regular arrangement of planes but fails to represent a single segment as one complete entity. On the other hand, our algorithm successfully separates out each segment as a whole. Fig. 6 (d) shows the result of our proposed method while Table 2 shows the quantitative comparison of these approaches. We report the mean intersection over union of 2 sites using the above mentioned approaches.

4. CONCLUSIONS AND FUTURE WORK

The solution proposed for 3D scene segmentation, in the form of the hierarchical tree approach, is simple but has proved to be effective for the reconstruction of planar and non-planar scenes. We have successfully used the energy function to explore the data in more detail, while ensuring the control over the semi-automatic selection of correct segments. This work is applicable to geometry that exhibits structural regularity. Seminal work on structural regularity by Pauly 2008 [10] quantifies regularity in 7 different categories and our projection sequences are applicable to 4 of them: Rotation, Translation, Rot $\times$ Trans and Trans $\times$ Trans. If coarse segmentation successfully detects diagonal structures/beams, proposed projection sequences can be used after performing axis alignment of the structure. Otherwise, such structures are not segmented further. We take it as a pointer for our future work.

Table 2. Mean intersection over union (mIOU) for synthetic and real data using 3 different algorithms.

Dataset	**Points**	**Mean IOU**		
		Schnabel [8]	RAPter [5]	Ours
Temple	100,031	46.89	51.04	**64.87**
Drawar Fort	550,382	48.31	56.43	**58.41**



(a) Point cloud. (b) RANSAC [8] segmentation.

(c) DBScan [9]. (d) Our hierarchical segmentation.

Fig. 7. Segmentation of Masjid Wazir Khan.

Table 1. Comparison of automatically generated (AG) primitives with ground truth (GT) and accuracy (Acc%).

Site	**Dimensions**	**Points**	**Arches**			**Domes**			**Minarets/Pillars**		
	$L \times W \times H\ m^3$	in Bn.	GT.	AG.	Acc.	GT.	AG.	Acc.	GT.	AG.	Acc.
Masjid Wazir Khan	$91 \times 53 \times 33$	0.288	46	19	41.30	12	12	100	6	6	100
Masjid Khuadabad	$60 \times 36 \times 16$	0.548	12	7	58.33	21	19	90.4	0	0	NA
Derawar Fort	$1500 \times 1300 \times 30$	0.43	0	0	NA	0	0	NA	38	38	100
Roman building	$20 \times 36 \times 14$	0.02548	79	77	97.44	0	0	NA	0	0	NA
Temple	$10 \times 16 \times 44$	0.0154	0	0	NA	7	7	100	4	0	0

5. REFERENCES

[1] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *IEEE CVPR*, 2016.

[2] A. Martinovic, J. Knopp, H. Riemenschneider, and L.J.V. Gool, "3d all the way: Semantic segmentation of urban scenes from start to end in 3d," in *IEEE CVPR*, 2015.

[3] D. Dohan, B. Matejek, and T.Funkhouser, "Learning hierarchical semantic segmentations of LIDAR data," in *3DV*, Oct. 2015.

[4] H. Lin, J. Gao, Y. Zhou, G. Lu, M. Ye, C. Zhang, L. Liu, and R. Yang, "Semantic decomposition and reconstruction of residential scenes from lidar data," *ACM Trans. Graph.*, 2013.

[5] A. Monszpart, N. Mellado, G. Brostow, and N. Mitra, "RAPter: Rebuilding man-made scenes with regular arrangements of planes," *ACM SIGGRAPH*, 2015.

[6] R. Bajwa, S. R. Gilani, and M. Taj, "3D architectural modeling: Coarse-to-fine model fitting on point cloud," in *Proc 33rd CGI*, 2016.

[7] M. A. Fischler and C. R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Comm ACM*, 1981.

[8] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *CGF*, 2007.

[9] M. Ester, H. Kriegel, J. Sanderörg, Xiaowei Xu, et al., "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *Kdd*, 1996.

[10] M. Pauly, J. N. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas, "Discovering structural regularity in 3d geometry," in *ACM TOG*, 2008.

[11] Maturana, Daniel, and S.Scherer, "VoxNet: A 3d convolutional neural network for real-time object recognition," in *IROS*, 2015.

[12] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *IEEE CVPR, 2015,*.

[13] C. Ruizhongtai, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *CoRR*, 2016.

[14] N. Zhao M. Li, Y. Hu and L. Guo, "LPCCNet: A lightweight network for point cloud classification," *IEEE GRSL*, 2019.