CP4979

Vo-Norvana: Versatile Framework for Efficient Segmentation of Large Point Cloud Datasets

Erzhuo Che[1] and Michael J. Olsen[2]

Abstract

Dense 3D point clouds collected from rapidly evolving data acquisition techniques such as light detection and ranging (lidar) and structure from motion (SfM) multi-view stereo (MVS) photogrammetry contain detailed geometric information of a scene suitable for a wide variety of applications. Amongst the many processes within a typical point cloud processing workflow, segmentation is often a crucial step to group points with similar attributes to support more advanced modeling and analysis. Segmenting large point cloud datasets (i.e., hundreds of millions to billions of points) can be extremely time consuming and tedious to execute with current tools, which primarily rely on significant manual effort. While many automated methods have been proposed, the practicality, scalability, and versatility of these approaches remain a bottleneck stifling processing of large datasets. To overcome these challenges, this paper introduces a novel, generalized segmentation framework called *Vo-Norvana*, which incorporates a new voxelization technique, a normal variation analysis considering the positioning uncertainty of the point cloud, and a custom region growing process for clustering. The proposed framework was tested with several large-volume datasets collected in diverse scene types using several data acquisition platforms including terrestrial lidar, mobile lidar, airborne lidar, and drone-based SfM-MVS photogrammetry. In evaluating the accuracy of models generated from *Vo-Norvana* against manual segmentation, the average error of the position, orientation, and dimensions are 2.7 mm, 0.083°, and 0.9 mm, respectively. Over 0.2

[1] Assistant Professor (Senior Research), School of Civil and Construction Engineering, Oregon State University, 101 Kearney Hall, 1491 SW Campus Way, Corvallis, OR 97331. Email: Erzhuo.Che@oregonstate.edu

[2] Professor, School of Civil and Construction Engineering, Oregon State University, 101 Kearney Hall, 1491 SW Campus Way, Corvallis, OR 97331. Email: Michael.Olsen@oregonstate.edu

million points per second and 36 thousand voxels per second can be achieved when segmenting an airborne lidar dataset containing over 639 million points to about 1 million segments.

Keywords: point cloud, lidar, SfM, segmentation, voxelization, feature extraction

Introduction

Three-dimensional (3D) point clouds are a common form to digitally represent 3D objects or scenes. Techniques such as light detection and ranging (lidar) and structure from motion (SfM) multi-view stereo (MVS) photogrammetry have been adopted to collect 3D point clouds for numerous applications (Olsen et al., 2013) supporting a wide range of disciplines. Lidar systems can be generally categorized based on the acquisition platform into terrestrial laser scanning (TLS), mobile laser scanning (MLS), and airborne laser scanning (ALS). Point clouds contain the precise 3D location of each data record and sometimes include other basic information such as color, intensity (strength of return signal from lidar sensors), and so on. Additional processes including feature extraction, classification, and modeling are often required to extract higher-level information for a variety of applications. Point cloud segmentation is the process of grouping points based on common attributes such that instead of handling each individual data record, subsequent process can operate with each segment as the basic unit rather. Increasing spatial resolution and area of coverage in mapping efforts can raise data volumes exponentially, which poses a substantial challenge for both manual and automated segmentation approaches.

Herein, we propose a novel, generalized, and efficient point cloud segmentation framework to cope with a variety of scenes (e.g., urban, rural, industrial sites, etc.) and systems (e.g., TLS, MLS, ALS, SfM/MVS photogrammetry, etc.). This paper describes the three primary steps of this automated framework: data organization with a voxelization technique that can preserve geometric details, Normal Variation Analysis (*Norvana*), and point clustering. *Vo-Norvana* offers several advances over existing segmentation algorithms, including the ability to (1) cope with a variety of shapes in different sizes and produce high quality modelling results, (2) consider of positional uncertainty of the data both locally and globally, reducing improper segmentation results from noise when combining scans or flightlines from different locations, (3) achieve a much higher computational performance on large datasets (both in terms of number of points and spatial extent), and (4) be versatile in processing data from a variety of systems and scenes with minimal fine-tuning of parameters. These contributions are explored and demonstrated through extensive testing of several representative datasets where the approached showed high accuracy modelling results (within a few mm of manual modelling procedures), computational efficiency (over 0.2 million points per second and 36 thousand voxels per second for a large dataset of 639 million points).

Related Work

This section summarizes related research in point cloud segmentation. It will commence with a brief background on segmentation followed by a description of common approaches to implement segmentation including planar feature detection techniques, scan-to-bim processes, voxelization, data structuring, and artificial intelligence.

Segmentation background

At its basic level, segmentation is a process to group points based on one or more common attributes (e.g., intensity, color, normal, etc.). This process can divide millions of discrete points into clusters such that instead of handling each individual point, subsequent processes can be efficiently performed with each segment as the basic operating element, significantly reducing the data volume. For example, generating thousands of geometric models directly from the original point cloud can be very challenging given the ambiguity and complexity within a scene. In contrast, with a segmented point cloud, the modeling or fitting technique only needs to fit one or a few primitives to the points within each segment, resulting in more robust and efficient processing. Segmentation results can also be used as input for object-based classification due to richer information provided within each segment compared with individual points (e.g., Poux et al., 2020, Che et al., 2021a). Because manual processing of point cloud data can be extremely tedious and time-consuming, much research has focused on different aspects of automating point cloud segmentation in addition to point cloud processing more broadly.

We recently conducted a thorough literature review on the subject of object recognition, feature extraction, segmentation, and classification for MLS data (Che et al., 2019). Considering prior work documented by other researchers and other detailed reviews of the state of the art (e.g., Grilli et al., 2017, Xia et al., 2020), there has been a reasonably comprehensive coverage of existing segmentation methods for both TLS and MLS. Thus, for the remainder of this section, we will focus on segmentation approaches that apply to ALS and SfM/MVS point clouds, as well as provide a summary of more recent or highly relevant research associated with TLS and MLS to this work.

Compared with TLS and MLS, ALS data usually covers a much larger area and has a relatively more consistent point density but generally suffers from limited coverage on vertical faces of objects (e.g., buildings, trees, etc.) as well as lower resolution overall. Some of these issues can be reduced through optimized flight planning such as lower flight altitude, higher overlap, and orientation of the flight path (Vo et al., 2021). Given the large data volume and covered area, most ALS algorithms focus on specific applications such as land cover classification, 3D reconstruction, or 3D urban modeling (Wang, 2013; Wang et al., 2018).

Planar features

Within an urban environment, the vast majority of anthropogenic objects captured by ALS (e.g., roofs) consist of planar surfaces. Xia et al. (2020) thoroughly reviewed the state-of-the-art in extracting geometric primitives such as planes from 3D point clouds. Since that review was published, several new algorithms have been proposed (e.g., Poz et al., 2020, Wang et al., 2020, Zhu et al., 2021, Zhang et al., 2021). Towards the objective of fitting a plane, one or multiple techniques are employed for segmentation including the Hough transform, principal component analysis (PCA), random sample consensus (RANSAC), and/or region growing. In application to ALS data, most of these approaches have been focused on extracting roofs. Zhao et al. (2021) compared these techniques for roof segmentation and found that most approaches proved effective for their test datasets. Nevertheless, these approaches focused on only extracting planar surfaces such as rooftops, which can result in many artefacts of under- and over-segmentation for other geometric shapes (e.g., spheres).

A generalized segmentation algorithm capable of working with many different types of surfaces is necessary for reliable execution of subsequent processes such as classification (Vosselman, 2013). Other methods eliminate small planar segments by classifying point clouds into smooth and rough surface segments (e.g., Ni et al., 2017) in an effort to improve the segmentation. Point cloud segmentation algorithms for forest scenes are focused on individual tree segmentation (e.g., Wang et al., 2019, Yang et al., 2020).

In principle, most of the aforementioned methods should also be adaptable to point clouds generated with lidar or SfM/MVS data obtained from an uncrewed aerial system (UAS) platform because it has a similar scan acquisition geometry. However, to-date very few methods have been rigorously tested and demonstrated on UAS-lidar or SfM/MVS data likely due to differences in applications and variant noise levels from these systems.

Scan-to-BIM segmentation

Some approaches have been developed specifically for ground-based lidar point clouds, primarily for handling planar surfaces to extract features from building façades, indoor environments, bridges, and other anthropogenic objects. These features are then used in applications such as 3D model reconstruction, building information modeling (BIM), quality control inspection, progress tracking, cultural heritage, and so forth (Wang & Kim, 2019). As an example, Maalek et al. (2018) proposed a technique utilizing PCA that segments the point cloud into planar and linear features for the purpose of tracking progress on a construction site. Bassier et al. (2017) proposed an approach based on region growing and conditional random fields for the reconstruction of BIM models.

Voxelization

The voxelization process often organizes the point cloud into 3D grids or cubes containing a number of points whereas supervoxels can be derived by refining the boundary considering the homogeneity in terms of predefined attributes. The majority of generic segmentation approaches have been developed based on voxelization (e.g., Vo et al., 2015) and super-voxelization (e.g., Mahmoudabadi et al. 2013, Lin et al., 2017, Dong et al., 2018, Huang et al., 2019). For example, Xu et al., (2021) performed a comprehensive review on voxelization and super-voxelization methods and applications. These voxels or supervoxels themselves inherently provide an over-segmented result, requiring an additional step to cluster the supervoxels. It is worth noting that the supervoxelization is particularly effective for buildings and scenes containing large planar surfaces. Unfortunately, investigation of its performance for other curved surfaces has been limited.

Scan Pattern Grid

Recently, we proposed an efficient segmentation approach, namely *Norvana* (Che & Olsen, 2018). *Norvana* operated specifically on TLS data to take advantage of the scan pattern grid, which enabled all points within a scan to be stored into a compact 2D grid representation without any information loss. We then extended this conceptual idea to MLS data and developed *Mo-Norvana* (Che & Olsen, 2019), by introducing a robust trajectory reconstruction method that regenerates the scan pattern from an unorganized MLS point cloud. Both approaches proved to be very efficient, scalable, and capable of tackling a variety of shapes and objects.

Artificial Intelligence

Lately, due to the rapid development of computer vision, deep learning, and artificial intelligence (AI) technology combined with the growing availability of computing resources and public datasets, numerous studies have used AI for 3D point cloud processing. Rather than segmenting the point cloud into groups (i.e., instance segmentation) based on the surface characteristics, the computer vision and AI communities tend to be more concerned with classifying or labeling each individual point directly (called semantic segmentation) (Poux & Billen, 2019, Xie et al., 2020). Before deep learning became a widely popular technique in semantic segmentation for 3D point clouds, some work utilized machine learning methods. For example, Weinmann et al. (2015) analyzed different techniques in each step of a machine learning framework for semantic segmentation including neighborhood selection, feature extraction, feature selection, and supervised classifier. Recently, Bello et al. (2020) conducted a critical review on the use of deep learning in processing point clouds. In addition to this work, several relevant state-of-the-art review papers have been published that focus on specific applications such as autonomous driving (e.g., Li et al., 2021) and 3D heritage (Matrone et al., 2020).

Although it is undeniable that the deep learning technique is powerful and has tremendous potential in the context of point cloud processing, it is worth noting that most of the datasets used to develop and test these deep learning frameworks are often substantially sparser with significantly less geometric details than the dense point clouds acquired with survey-grade systems (Hackel et al., 2017). Additionally, most deep learning approaches require extensive compilations of high-quality training datasets to tackle a variety of systems and scenes, which is difficult given that the publicly available 3D training dataset cover very limited scenarios. Unfortunately, labeling 3D point cloud manually to generate training datasets can be tedious, time consuming, and often subjective. This immense effort required is rarely reported quantitatively, resulting in difficulties in predicting the cost of establishing an effective benchmark dataset. Hence, given this reliance on substantial training datasets, deep learning approaches face challenges of scalability and versatility for processing typical lidar datasets collected for the complex built environment.

Summary of Limitations

Although the aforementioned studies present reasonable segmentation results to different degrees, three important limitations in the state-of-the-art can be summarized as follows:

(1) The overwhelming majority of approaches focus on extracting planar patches or surfaces. Consequently, such assumptions substantially impact their performance on other basic shapes or more complex objects common within the built environment. In particular, this limitation significantly hinders the application to outdoor scenes, which consist of both geometric primitives and irregular shapes.

(2) Most existing approaches were only tested on a single dataset or several small datasets on the order of hundreds of thousands to a few million points. Many have also only been tested for a relatively small area where basic down-sampling can substantially reduce the data volume without significant loss of information. In practice, typical point cloud datasets often obtain hundreds of millions to billions of points and cover large areas. The efficiency of the majority of these methods do not scale linearly with

point size and sufficient information is not often reported to understand how they scale to larger datasets. Processing time typically increases exponentially with larger volumes of data resulting from the higher computational complexity, often involving many iterations and global optimizations.

(3) Very few studies test their segmentation methods on a variety of datasets from different platforms; most are geared towards to a specific system and application. Because deploying multiple lidar and/or drone systems on a single project is becoming more and more common to maximize coverage and completeness as well as improve efficiency throughout the area of interest, it is crucial to have a method available that can simultaneously handle data from different sources. Lastly, a general-purpose segmentation approach can also improve data reuse and increase the value of the point cloud data as it supports many downstream analyses and applications. For instance, to be able to reconstruct 3D as-built models from high resolution point clouds to create digital twins, an automated and scalable segmentation is required to reduce the data complexity and simplify the modeling process by dividing the unorganized data into more manageable and meaningful groups.

Objectives

To overcome these limitations, we propose a novel, generalized segmentation framework, namely *Vo-Norvana*, that: (1) copes with a variety of regular and irregular shapes and objects; (2) reliably processes and efficiently scales to handle expansive, unorganized point cloud data containing hundreds of millions of points; (3) considers the level of uncertainty of the point cloud data to improve the segmentation, and (4) robustly handles data collected from different scenes and systems. We will illustrate the workflow in the methodology section and then present a series of experiments to demonstrate the effectiveness of the proposed approach both qualitatively and quantitatively.

To clarify the novelty of the proposed approach, especially compared against our prior work including *Norvana* (Che & Olsen, 2018) and *Mo-Norvana* (Che & Olsen, 2019), we will explain key differences and innovations. First, *Vo-Norvana* structures the unorganized point cloud data via a new voxelization approach whereas *Norvana* and *Mo-Norvana* exploit the scan pattern grid, requiring specific sensor parameters and/or organized input data. Consequently, the proposed *Vo-Norvana* algorithm is able to cope with a much broader range of 3D point cloud data from any platform (both individually and merged together) while the previous approaches can only handle specific data collected from TLS and MLS systems, respectively. Additionally, without being limited to work solely within a scan pattern grid, the proposed approach supports analysis at a custom scale that can be different from the acquisition resolution. Lastly, it is also worth pointing out that although all these three methods share a similar concept in the normal variation analysis, the implementation of the *Vo-Norvana* is fundamentally different in how it handles the data to compute the normals as well as its ability to consider the data uncertainty, enabling rougher surfaces to be effectively extracted.

Methodology

The *Vo-Norvana* segmentation takes full-resolution georeferenced or registered point clouds as input without requiring any prior cleaning or subsampling. The segmentation consists of three primary steps (Figure 1), including: data organization (Section 2.1), normal variation analysis (Section 2.2), and point clustering (Section 2.3). Firstly, the input point cloud data is partitioned into 3D tiles with overlap along the boundary followed by a new voxelization approach that can reduce the data volume while preserving more geometric details compared with traditional voxelization methods. Secondly, in each 3D tile, each point will be analyzed with its neighbors to classify it as a *smooth*, *rough*, or *invalid* surface point. Then, based on the classification result, a custom region growing algorithm groups each class of points. Finally, the point classification and clustering results are mapped back to the original input datasets to ensure the integrity of the data and generate the full segmented point cloud.

Data Organization

Many voxelization approaches suffer from excessive memory consumption due to numerous empty voxels. Consequentially, methods directly exploiting voxelization for organizing a point cloud must balance the voxel size with the spatial extent of the point cloud. For example, for a 100 by 100 m area with a vertical extent of 50 m, an analysis at a scale (i.e., voxel size) of 0.05 m requires 4 billion voxels to be constructed. This large quantity of voxels is difficult to manage and significantly hinders processing including the point cloud segmentation as each voxel has to be analyzed. Sometimes multiple iterations are required, further compounding the processing time. Additionally, using the aggregated information from voxels (e.g., single point at the center or centroid of a voxel) instead of the actual points can result in loss of geometric details. For instance, some methods resample the point cloud with the center coordinates of each voxel occupied by one or more points, while others use the average, median, or centroid coordinates of all the points within each voxel. Although these methods can simplify the analysis and reduce the computation complexity, they result in difficulty in precisely representing the geometry of an object. Hence, we propose a dynamic voxelization process to organize and structure the point cloud while preserving more geometric details.

3D Tiling

First, in the tiling process, we align the point cloud with the principal axis computed from principal component analysis (PCA) to reduce the total number of voxels required (Figure 2). In many cases, a rotation of the point cloud data about the Z-axis only would suffice for reducing the memory consumption. Secondly, we partition the point cloud data via coarse 3D tiles such that the more intense computations can take place in each tile to achieve high efficiency and low memory consumption. Then, to avoid boundary artifacts and ensure subsequent processing is seamless across adjacent tiles, we buffer each side of a 3D tile to provide overlap. The buffer size should be determined by considering the size of the searching window utilized in the subsequent processes to ensure a seamless analysis throughout the workflow. In the presented implementation of the proposed segmentation method, the minimum width of the buffer is 2 voxels on each side, and the dimension of each 3D tile is initialized as 200 times of the voxel size, S_v, such that each 3D tile contains ~8.5 million (204 x 204 x 204) voxels in total. This allows the processing to balance between multiple factors including the extent of the data, typical size of a segment, the computing capacity (e.g., RAM, number of threads) and so on. It is also worth noting that the proposed framework includes the process of merging all the analysis results at the point clustering stage to cope with the large segments covered in multiple tiles.

Voxelization

Recently, we introduced a related voxelization approach to down-sample point cloud data and demonstrated its effectiveness for ground filtering (Che et al., 2021b). Herein, we significantly improved this new voxelization approach to organize the point cloud data to enable more efficient data processing, especially in terms of memory consumption.

Because each 3D tile can be processed independently, only the memory associated with the voxels within the specific tile under analysis need to be allocated at a time. Thus, we can dynamically voxelize the point cloud within each 3D tile while still liking the information and analysis results associated with each point back to the original point cloud throughout the entire process. In other words, the 3D tiles and voxels only serve as a structure to organize the point cloud, but do not actually down-sample the data for the final results as is commonly performed with most voxelization techniques. To further improve the efficiency, we record the indices of the 3D tile and voxel for each point such that the points can be directly mapped to the proper voxel and 3D tile with limited computational expense in case multiple iterations or processing steps are needed. To preserve the geometric details, we mark the point that is closest to the center of the corresponding voxel as *core* point candidates to represent the 3D coordinates of that specific voxel (Figure 3). Notice that we utilize the voxel center rather than the barycenter or median coordinates because the point clouds can have highly variable point density, especially for lidar data. As a result, other approaches can bias the sampling result and result in challenges associated with modelling multiple scans and/or data sources. Next, to further normalize the point density, at each *core* point candidate, we search its neighbors with a diameter of half of the voxel scale and determine if it is indeed the *core* point with the shortest distance to the corresponding voxel center. If so, this *core* point candidate is marked as a *core* point while all the other points in the voxel are classified as *accessory* points.

After labeling the point cloud as *core* and *accessory* points, we can directly simplify the data robustly with a consistent point density by sampling the *core* points for the initial analysis before linking the results back to the full point cloud for the full segmentation. Compared with most other voxelization techniques, the proposed approach can preserve more geometric details because it samples from the original points rather than re-sampling using voxels or aggregating points. For example, given the point cloud in

Figure 3, simply using the voxel centers for re-sampling (occupied voxels) would completely fail to represent the zigzag pattern. If the re-sampling is conducted by taking the average or median coordinates of all points in each voxel, the sharp corners would likely be undesirably smoothed. Moreover, to ensure the robustness of the computation, a minimum number of points is often required. In other cases, a larger voxel size S_{v} is needed, which would compromise details. Fortunately, the proposed voxelization framework keeps the original point cloud and only the *core* points are distributed with consistent spacing. Thus, further analysis for point cloud segmentation can be applied to the *core* points only with reduced computational complexity, followed by projecting the analysis and segmentation results from the *core* points to the *accessory* points to ensure the completeness.

Norvana

Our prior work of segmentation methods developed exclusively for terrestrial laser scanning (TLS) and mobile laser scanning (MLS) data both organize the point cloud data into a 2D scan pattern grid, followed by a normal variation analysis (*Norvana*), which exploits this data structure (Che & Olsen, 2018; Che & Olsen, 2019). In contrast, in this work our objective is to develop a general segmentation approach that can handle any type of point cloud. Hence, we generalize this technique by extending the same concept into 3D space as well as introduce several significant improvements to allow more flexibility in handling handle different types and qualities of point clouds. For example, while the 2D scan pattern grid embeds a lot of constraints in terms of both geometry and topology in neighbor searching, additional geometric constraints need to be added when generating a local triangular mesh in the proposed generalized *Norvana* stage. In addition, we now consider the data uncertainty including both local (e.g., ranging precision) and global (e.g., registration accuracy) errors during the *Norvana* process. Furthermore, although the previous versions of *Norvana* were able to identify smooth surfaces, they were sensitive to vegetation and other rough surfaces. To overcome this limitation, we implemented a multi-step feature classification to categorize the point cloud into *smooth*, *rough*, or *invalid* surfaces as well as unclassified points such that they are handled differently in the segmentation.

With the voxels serving as indices of the point cloud, a variety of neighbor searching strategies can be employed efficiently. Given a point or coordinates, we define its neighbor as the points lying within its corresponding and adjacent voxels. As a result, a total of 26 voxels needs to be jointly examined for neighbor searching. The normal vector at each point can be estimated by computing the eigenvector corresponding to the smallest eigenvalue derived from this point and its neighboring *core* points utilizing singular value decomposition (SVD). Notice that the normal estimation can take place at a different scale than the selected voxel size. The *core* points will be labeled as unclassified if the normal estimation does not yield a valid result (e.g., no close neighbors). Otherwise, the normal variation analysis is performed to each *core* point with its neighboring *core* points searched from its adjacent neighbor voxels. When the number of the *core* neighbor points for a *core* point is less than the given threshold ($T_{\mathrm{N\_Neighbors}}$), this *core* point is directly classified as an *invalid* surface point because it cannot form a reliable local surface for further analysis. Otherwise, the 3D coordinates of a *core* point and its neighbors are projected to a local coordinate system to align with the normal vector where the core point under analysis is defined as the origin, the normal vector is defined as z'-axis, and the x' and y'-axis are set arbitrarily.

One limitation of the segmentation analyzing the normal variations is its sensitivity to the positional errors and surface roughness (Che & Olsen, 2018). To cope with the positional uncertainty within point clouds, we adjust the positions of the neighboring *core* points towards the *core* point under analysis in z' direction (Figure 4). Note that this adjustment is only performed temporarily for this local normal computation, and the actual coordinates of the point remain unchanged. We consider two types of uncertainty and simplified them by utilizing two constant parameters, σ_{local} and σ_{global}. If the neighboring *core* point is from the same source (e.g., scan, flight line, sensor), we utilize σ_{local}, which can be set as a function of the ranging accuracy according to the specifications of the system (mm- to cm-level). When combining data from different sources, in addition to σ_{local}, we further consider the errors from data processing (e.g., registration, georeferencing) by defining the global uncertainty σ_{global} (mm- to cm-level), which is often available in a data processing report. In some cases, the system specifications and/or data reports are not available, the parameters can be estimated by measuring deviations of the point cloud in localized areas until a reasonable sample is obtained. To further consider the uncertainty at the *core* point in the following local analysis, the maximum allowable adjustment of its neighbor *core* points is set as two times of the σ_{local} or σ_{global}.

Because we assumed that the point cloud only captures the surface of an object, we remove the neighbor *core* points that are closer than $0.25S_{\text{v}}$ on the x'-y' plane to avoid creating a complex triangular mesh locally. Next, we sort the neighbor points by their projected horizontal angle (θ) within the x'-y' plane and generate a triangular mesh around the *core* point under analysis. To avoid sharp triangles and improve the robustness of the segmentation, when the angle θ exceeds a threshold T_{θ} (maximum tolerant angle), the *core* point under analysis is marked as an *invalid* surface. Otherwise, the normals of each triangle can be computed, and the normal gradients at the *core* point under analysis in different directions are computed with each pair of triangles in the local mesh. We further compare the largest normal gradient against a threshold of $T_{\Delta\text{Norm}}$ (maximum tolerated normal gradient) to label the *core* point under analysis as a *smooth* ($\leq T_{\Delta\text{Norm}}$) or *rough* surface point ($> T_{\Delta\text{Norm}}$).

Point Clustering

Several algorithms have been proposed to cluster points based on common attributes. For example, connected components is a common approach to efficiently group linked voxels with limited constraints due to the straightforward and fast neighbor searching process (Olsen et al., 2015) from the organization provided by the voxelization. Meanwhile, region growing is another common point clustering method very similar to connected components; however, it typically requires more constraints (e.g., difference in normals), providing more flexibility (Che & Olsen, 2018). In our proposed method, we extract *smooth*, *rough*, and *invalid* surfaces in order by utilizing some of the core concepts of the connected components approach to segment the point cloud but with different constraints and criteria (Table 1) similar to a region growing process. For each class of surfaces, the *core* points are first clustered and then mapped to their nearby *core* and *accessory* points. Such a process enables high efficiency via the voxel-based neighbor searching while each surface class can still be segmented based on their general geometric characteristics.

Specifically, we initiate the *core* point clustering by grouping the *smooth* surface *core* points. To determine whether a cluster from a core point can grow to a connected one, we first compute the difference of normals at these two core points ($\Delta\mathrm{Norm}$ in Figure 5). There are cases in which points lying on different surfaces have similar normal vectors. As a result, only checking the normal gradient can result in under-segmentation issues where multiple surfaces can be grouped into the same segment. To cope with this situation, we temporarily adjust the position the one *core* point (point B in Figure 5) under analysis along the direction of the normal vector of the other point (point A in Figure 5) following the same process described in Section 2.3. Next, the normal vector as well as the adjusted coordinates of point B (point B' in Figure 5) can be used to define a plane. On this plane, we assume point A and point B' both lie on an arc where the normal vector at B' can be computed. This yields another estimation of normal difference between these two *core* points ($\Delta\mathrm{Norm}'$). The same analysis is then applied with point B swapped for point A to obtain another estimation of normal difference. Ultimately, this analysis essentially combines the estimation of both curvature and the normal gradient, strengthening the robustness by providing a total of three estimations of the normal gradient between points A and B. To grow from one *smooth* surface *core* point to the other, all three estimations need to be equal to or less than the threshold $T_{\Delta\mathrm{Norm}}$.

After clustering all of the *smooth* surface *core* points, we dismiss smaller segments if the number of *core* points within a segment contains are less than a user-given parameter, T_{N_Cores}. Then, for each point that does not belong to a *smooth* surface segment, we first adjust and estimate the normal difference with all of its neighboring *core* points, which are segmented using the same approach as the *core* points. Among all the neighboring *core* points meeting the criteria of growing, we populate the point under analysis using the segment ID of the one with the shortest projected distance along the normal vector. This mapping approach not only groups the accessory points lying on a *smooth* surface to the nearest surface segment but also groups points lying on a sharp edge of multiple surfaces.

To further cluster the *rough* and *invalid* surfaces, we group the *smooth* and *rough* surface *core* points that have not been assigned to a segment first. The same procedure as segmenting *smooth* surfaces is followed but with different criteria to determine whether a point belongs to a *rough* surface. Because a *rough* surface has a larger deviation in the surface normal direction (Points D, E, and F in Figure 5), the metrics that we use for *smooth* surface would over-segment the point cloud in many cases. Hence, we simply compute and compare the normal difference between two points against the threshold $T_{\Delta\mathrm{Norm}}$ to preserve the *rough* surface. This result is refined by examining the number of *core* members T_{N_Cores} which can be given based on the voxel size S_{v} and the minimum dimension of the objects of interest in the scene. Finally, all of the *core* points that have not yet been tagged with a segment ID are grouped into *invalid* surface segments. The same procedures are followed where the criterion is the 3D distance, which has already been embedded in the voxel-based neighbor searching, similar to connected components. The points tagged to a segment that fails to meet the threshold of T_{N_Cores} will be assigned as *unclassified* noise.

Lastly, some *core* points lying along the surface edges as well as some of the accessory points may not yet be segmented. To map the *core* point segmentation results to these unlabeled points, we use the similar criteria to determine whether an unlabeled point belongs to a segment or not. Note that because the estimated normal of a point located on the edge between surfaces can be unreliable, for *smooth* surface segments, we only take one estimation of the normal gradient with the known normal vector of the segmented *core* point. If more than one segment meets the criteria at an unlabeled point, this point will be labeled as the same with the closest labeled *core* point. The projected distance is used for *smooth* and *rough* surface segments whereas the 3D distance is used for *invalid* surfaces.

Experiment

Overview

We tested the proposed *Vo-Norvana* segmentation both quantitatively and qualitatively with five distinct datasets (Figure 6, Table 2) from different systems to examine the effectiveness and versatility of the proposed method. Notably, Table 2 relates key information related to each dataset (e.g., dimensions and point count) and parameters used in processing (e.g., voxel size) to several metrics for evaluating processing efficiency. The first dataset consists of a single TLS scan (Leica ScanStation P40) acquired in an indoor setting and captures basic geometric shapes for quantitatively evaluating and comparing the quality of the segmentation results for modeling purposes. The remaining datasets are significantly larger in terms of both data size and extent to evaluate the scalability and robustness of the method, including a TLS dataset containing 8 scans, a MLS dataset collected by a Leica Pegasus:Two system, an ALS dataset containing 20 flightlines, and a point cloud data from an uncrewed airborne system (UAS) using SfM MVS photogrammetry. Note that the adjusted extent is the dimension of the data after the data is rotated to align to its principal axis (Figure 2) while a voxel containing at least one data point are defined as a valid voxel. All data are stored in an unorganized format, namely LASzip (i.e., LAZ), compressed from the ASPRS LAS format (Isenburg, 2013; ASPRS, 2019). Additional details about each dataset and the selected parameters will be discussed in the following sections. The metrics of quantifying the efficiency of the proposed method (Table 2) are discussed in details in the Computational Performance section.

Computational Performance

The proposed algorithm was implemented using C++ with OpenMP parallel programming within the Visual Studio 2019 platform. All tests were performed on a desktop computer configured with Intel Xeon W-2145 CPU @ 3.70 GHz (8 cores, 16 threads) and 128 GB RAM. The processing times reported in Table 2 for each dataset includes all steps (e.g., data preparation, normal estimation, voxelization, segmentation, etc.) except for data I/O. To holistically evaluate the computational performance of *Vo-Norvana*, in addition to the overall processing time, we calculate the point, voxel, and segment-based performance. The point-based performance is computed using the total number of points to represent the data volume, in general. Because the voxelization process simplifies the data, the spatial extent of the dataset and voxel size should also be considered when analyzing processing times. We calculated the voxel-based performance using the total number of valid voxels so that empty voxels were excluded. Because the highest resolution voxel size is ultimately a function of the point density and scale of the features to be extracted, the voxel-based performance turned out to be somewhat consistent across the different datasets. It is worth noting that although a larger voxel size can increase performance, the lower resolution can substantially limit the use of the data. One reason behind the lower voxel-based efficiency for the TLS dataset is that it has a relatively large extent in the Z direction compared with the MLS, ALS, and UAS-SfM datasets. This imbalance potentially reduces the spatial coherence when loading data into the cache. Lastly, the number of segments represents the overall complexity of the scene given the voxel size; hence, we calculated the segment-based performance (i.e., number of segments per second) to highlight the variety of the testing data in terms of the scene complexity.

In summary, based on the results presented in Table 2, *Vo-Norvana* is highly efficient in processing unorganized point cloud data with a wide range of complexities. In addition, the extraordinary scalability of our approach is demonstrated by successful testing of datasets containing hundreds of millions of points.

Accuracy Assessment

Many researchers simply report the accuracy of a segmentation method using a point-based assessment using common statistical metrics such as recall, precision, F-1 score, accuracy, and so on. Unfortunately, such metrics treat each point with the same weight; as a result, they can be substantially biased by the segment size when a dataset is large where larger segments dominate and the finer details of relatively simple segments are ignored. Thus, we assessed and analyzed the accuracy of *Vo-Norvana* by evaluating fitted geometric models derived from the segmented point clouds. The reference models we used are derived from the manual and *Norvana* segmentations from our prior work of the same benchmark dataset (Che et al. 2018), which consists of one plane, two spheres, two cylinders and two cones.

A voxel size of 0.01 m was used to be consistent with the analysis scale used in our prior work developing *Norvana*. In *Vo-Norvana*, a local error σ_{local} of 3 mm was given based on the scanner specifications. Unlike *Norvana*, *Vo-Norvana* does not have a designated step to remove mixed pixels given that the scanner location is unknown due to the unorganized data format. However, we found increasing the minimum and lowering the maximum neighbor angles T_θ can somewhat mitigate errors caused by mixed pixels. For the comparison, the intermediate results of segmenting the *core* points as well as those from the full segmentation (Figure 7) are both evaluated to demonstrate the ability of *Vo-Norvana* to preserve geometric details with the benefits of mapping the results back to the full point cloud rather than work with a down-sampled version as is common in many other works.

In addition to the manual and *Norvana* segmentation, we also performed segmentation via *RANSAC* and QTPS (Zhu et al., 2021) for comparison. For both methods, we fine-tuned the settings to match the parameters used in *Norvana* and *Vo-Norvana*. *RANSAC* was set to specifically detect planes, cylinders, and cones from the input point cloud. It is also worth noting that we only used the *RANSAC* segmentation/fitting results but not the shape recognition information because of the poor recognition that occurred, especially between planes, spheres, and cylinders. The *RANSAC* segmentation produced 53 segments while QTPS segments the point cloud into 102 segments (Figure 7). Because QTPS was developed primarily for ALS data targeting planar surfaces, it significantly over-segmented the curved surfaces such that a meaningful comprehensive quantitative analysis could not be conducted for the objects of interest.

The modeling process was then performed using the Leica Cyclone software with the RANSAC option disabled to ensure the fitting was fully based on least squares. The error statistics are first reported to validate the fitting quality of each model and method (Table 3). The mean, standard deviation, and absolute mean errors among all the four approaches are mostly on par; however, the absolute maximum errors for the *Vo-Norvana* results are slightly larger than, but still comparable with, the manual, *RANSAC* and *Norvana* segmentation. As most of the absolute maximum errors are near the voxel size (0.01 m) used in the voxelization, such differences can be largely explained by the specified scale of analysis. Next, we compared the number of points in the segments for modeling which shows that the *Vo-Norvana* segmentation is more similar to the manual process because both operate data in a 3D space, whereas *Norvana* organizes data into a 2D scan pattern. Additionally, because *Vo-Norvana* considers the point uncertainty in the process, it is less sensitive to noise compared with *Norvana* in our prior work.

As another approach to further assess the accuracy of the modeling results using the different segmentation approaches, we compared the position, orientation, and shape of the models (Table 4). As the comparison shows, the accuracy of the position and shape is mostly at the millimeter, if not sub-millimeter, level while the errors in orientation are generally lower than 0.1°. These errors vary with the objects because of their shapes, sizes, materials and so forth. While in this simple case, the *RANSAC* segmentation yields slightly better results overall mostly due to the extra input of the target primitives, RANSAC was unsuccessful at obtaining satisfactory results with drastic over- and under-segmentation as well as requiring a very long processing time on the other datasets tested. Also note that the accuracy of the *core*-only segmentation is worse than the full *Vo-Norvana* result, which demonstrates that mapping the segmentation results from the voxels to all of the points helps improve the accuracy in modeling applications. Nevertheless, the *core*-only segmentation result can still be sufficient for many applications to provide higher computational efficiency and lower data volume, if desired.

Versatility Tests

We further performed *Vo-Norvana* segmentation to process four large datasets acquired by TLS, MLS, ALS, and UAS-SfM to evaluate the versatility and scalability of the algorithm. These datasets cover a wide range of data collection methods, scene types, and objects. We also attempted to test several existing methods (e.g., *RANSAC*, *QTPS*) for comparison. Unfortunately, these approaches struggled in processing the large datasets (both point counts and spatial extent) and suffered substantial over and under-segmentation given the complexity of these scenes. Hence a meaningful comparison is not possible.

3.4.1 TLS Testing

The TLS test dataset was collected near Weatherford Hall located on the Oregon State University campus in Corvallis, Oregon, United States. The angular resolution of each scan is 0.02° and the maximum range is 120 m. Given that this dataset contains multiple scans registered together, the reported registration RMS error statistic of 6 mm is used to estimate σ_{global}. The voxel size was set as 0.05 m based on our prior work while the minimum segment size was set to 50 voxels to minimize segmentation of small objects. All other parameters (i.e., σ_{local}, $T_{\Delta\text{Norm}}$, $T_{\text{N\_Neighbors}}$, T_{θ}) were kept the same as the benchmark test to demonstrate that the method is not highly sensitive to parameter selection (Table 2). Not that this same dataset was also tested and documented in detail in our prior work (Che & Olsen, 2018) for comparison.

Vo-Norvana categorizes the input point cloud into four classes: *smooth* surface, *rough* surface, *invalid* surface, and *unclassified* noise. For the TLS dataset (Figure 8), man-made objects such as road, sidewalk, and buildings are mostly classified as *smooth* surface while the tree trunks and grass categorized as *rough* surfaces. Tree branches and leaves, as well as other linear or other irregular shapes are mostly tagged as *invalid* surfaces, whereas the *unclassified* noise primarily consists of small clusters.

Close-up views of different objects further highlight the effectiveness of the proposed approach (Figure 9). For example, the architectural features (e.g., divided blocks, columns, windows, etc.) were correctly segmented for the building façade (Figure 9 (A) and (B)). Note that the façade below the balcony at the bottom of the building is a curved surface. Similarly, the curb face is effectively segmented as a single segment (Figure 9 (B)). Additionally, several moving objects were captured during the scans (e.g., vehicles, bikes, pedestrians, etc.), resulting in numerous unwanted points in the data. *Vo-Norvana* effectively segmented these points into clusters such that they can be easily removed given that these objects were grouped into segments mostly classified as invalid surfaces. For the trees, which vary in species and sizes throughout the scene (Figure 9 (A) and (C)), the tree trunks and crowns were separated into different segments because they were categorized into different classes, as discussed in the prior section.

MLS Testing

Next, we evaluated the proposed method on an MLS dataset collected along a 1.3 km stretch of road through a sub-urban area in Philomath, Oregon, United States with an average speed of 6.7 m/s and an angular resolution of 0.07°. Compared to the TLS data consisting of multiple scans to cover an area, MLS data typically has a lower point density, depending on the range and driving speed. Moreover, in addition to the ranging and angle measurement errors of the lidar sensor itself, because the GNSS receivers and Inertial Measurement Unit (IMU) provide direct georeferencing, the accuracy of MLS point cloud is typically lower than TLS. As a result, we set the σ_{local} to 0.01 m based on empirical evaluation of the data quality of several datasets with this specific system. The parameters $T_{\text{N\_Neighbors}}$ and T_{θ} were set to 3 and 150°, respectively, due to the lower point density and rare occurrence of mixed pixels in MLS data.

The segmented results (Figure 10) show that most ground points were grouped into a single segment including the road, sidewalks, and driveways except for areas further from the scanner with a local point density lower than the analysis scale (e.g., black points in Figure 10 (A, B)). Similar to the TLS data, the sidewalk and roadway were segmented together because they are smoothly connected via the curb ramps and driveways. Although most of the curbs were separated from the roadway and sidewalk, some were over-segmented into smaller sections rather than as a long stretch as in the TLS testing (Figure 10 (A, C, D, E)). The primary reason is that the point density in such areas is relatively low, and hence the normal estimation is less accurate. In addition to horizontal features, the vertical features were accurately segmented. For example, not only were the utility poles clearly distinguished from the ground, but different components (e.g., pole, ground wire, guy wire, crossarm, transformer, etc.) were also be separated into their own segments (Figure 10 (A, B, C, D)), potentially supporting detailed modeling and further analysis. Some utility poles appear to be over-segmented (e.g., Figure 10 (D)) because the secondary wires occlude the MLS system at certain angles, dividing the pole into multiple sections. If the point density is sufficient on the wires and powerlines, *Vo-Norvana* can be used to extract and segment these linear features, which are classified as *invalid* surface points (Figure 10 (C, D)). Signs of varying sizes located at different heights were accurately segmented with the poles and boards properly separated. (Figure 10 (A, B, E)). In a few cases, the sign boards were spilt into two parts at the pole because only the back of the sign was captured by the MLS system.

ALS Testing

Very few segmentation methods are tested on both ground-based and airborne lidar datasets. Hence, to validate the versatility and scalability of the *Vo-Norvana* segmentation, we tested it with a massive, publicly available dataset (Laefer et al., 2017) collected in Dublin, Ireland, in 2015 (Figure 11) encompassing an area of approximately 9 sq. km. More specifically, we selected all 20 flight lines on the Northeast and Southwest direction from the entire dataset, comprising nearly 640 million points. A typical ALS data processing workflow would partition the data into 2D tiles to make this immense data volume more manageable early in and throughout the processing. While this approach is effective for some processing tasks, tiling requires adding overlaps between tiles as well as additional treatments to link segments across multiple tiles to ensure the consistency of the segmentation results, particularly near the boundary of each tile. Since the *Vo-Norvana* segmentation establishes 3D tiles during voxelization as only a temporary measure, we input the individual flight lines as separate files and processed the entire test dataset directly without having to process as individual tiles. This strategy allowed us to account for offsets between flightlines in the segmentation analysis compared to the typical processing approach of merging data from all flightlines before tiling. The voxel size S_{v} was determined to be 0.35 m based on the typical point density while the local error σ_{local} and global error σ_{global} were both 0.03 m according to the data report (Laefer et al., 2017). We also set the minimum segment size to ensure each segment occupied at least 10 voxels. The other parameters $T_{\Delta\mathrm{Norm}}$, $T_{\mathrm{N\_Neighbors}}$ and T_{θ} were kept the same with prior tests to be consistent. Given the vast size and complexity of this dataset, herein we will showcase select smaller regions across the dataset to demonstrate the effectiveness of the proposed approach. To provide some reference of the actual scene, we added the corresponding satellite images along with the screenshot of the segmented point cloud.

First of all, the ground surface (mostly paved road or sidewalk surfaces) was segmented into a couple of very large segments, demonstrating that *Vo-Norvana* can serve effectively as a ground filtering approach for ALS data in an urban scene (Figures 12-14). In this case, the ground (e.g., road surface, sidewalk, etc.) did not turn out to be one segment because there are railroad tracks passing through the scene and splitting the road surface into two parts. Then along the river, the water surface was clustered given the fact that it appeared as a *smooth* or *rough* surface in the lidar data (Figures 12 and 14). In addition, the vehicles captured in the scene were not separated into individual objects (Figure 14) because *Vo-Norvana* segmented the body of a car into multiple parts based on the distinct changes in geometry.

A variety of roof structures and types are presented in this area including flat, hip, valley, dormer, dome, and others (Figure 12, 13, 15, and 16). Most of the roofs consist of planar surfaces and each face was extracted as a segment. Admittingly, these simplistic roofs are relatively easy to tackle with any segmentation approach where basic plane fitting can readily distinguish each face of the roof. However, in contrast, the courts and church (the left and center of Figure 12) both feature a dome roof, which can be very challenging to most existing methods. With *Vo-Norvana*, they were correctly segmented into a single cluster while the tips were separated from the dome. Another example shows that *Vo-Norvana* also managed to divide an octagonal roof to each planar face with other roof in different types and sizes (Figure 16). It is also worth noting that most facets of the arched roof of the train station (Figure 15) were segmented correctly with the exception of a few facets representing glass skylights that are adversely impacted by the increased lidar ranging uncertainty. Nevertheless, despite these minor issues, *Vo-Norvana* robustly copes with a variety of complex geometric surfaces throughout the scene.

The *Vo-Norvana* segmentation is proven to be capable of handling objects and features in a variety of shapes and dimensions. The ALS dataset captures several other types of assets such as streetlamps, traffic lights, and poles. Because the spatial resolution is much lower than typical TLS and MLS data, the points lying on these objects were classified as *invalid* surfaces but were still segmented properly (Figure 14). Similarly, the guard rails on the bridge were also clustered into a segment. In addition to the infrastructure, tree crowns can be also of interest in the ALS point cloud, and *Vo-Norvana* can be used to reliably distinguish individual trees within the point cloud. The points representing the tree crowns were classified as *invalid* surface points, and each was typically clustered into a single large segment; however, sometimes they were subdivided into a few smaller segments that can be grouped with further process (Figures 12 and 13).

UAS-SfM Testing

Lastly, we rigorously tested *Vo-Norvana* segmentation on a UAS-SfM point cloud, which tends to be noisier compared with the TLS data. These data were collected from a historic paper mill next to the Willamette Falls in Oregon City, Oregon, United States (Bresky, 2016) using a DJI Phantom 4 RTK UAS with approximately 10 ground control points observed with GNSS and processed against a base station. The post-processing was performed in Agisoft Metashape, providing a typical point spacing in the dense point cloud of approximately 0.05 m. It is worth noting that even though multiple flightlines with significant overlap were planned and flown to cover the area of interest, unlike the airborne lidar, the point cloud itself cannot be divided into individual flightlines given that the SfM process requires data from overlapping flightlines be combined to reconstruct the point cloud via bundle adjustment. As a result, we considered the UAS-SfM point cloud as a single, merged point cloud and did not apply σ_{global} while setting σ_{local} to 0.01 m based on the residuals at ground control points. The other parameters were again kept as consistent as possible with the other tests. *Vo-Norvana* yielded a quick but robust segmentation of various features in the scene (Figure 17). For example, the hydroelectric dam (Figure 17 (A)) was captured in the point cloud and segmented into different parts. In the paper mill, several storage tanks with spherical or cylindrical shapes of different sizes were cleanly extracted (Figure 17 (B, C)). The performance of the proposed segmentation of the roofs was robust (Figure 17 (D)) and similar to the result of ALS testing discussed in the prior section.

Summary

In the experiment, we tested a total of five different point cloud datasets including a TLS benchmark dataset, an outdoor TLS dataset collected in the university campus, an MLS dataset captured in a suburban area, an ALS dataset covering an entire city in a high resolution, and a UAS-SfM point cloud acquired from an industrial site. We first evaluated the computational performance of the *Vo-Norvana* segmentation considering the number of points, data extent, and scene complexity (e.g., number of segments). By exploiting parallel programming (8 cores, 16 threads), the largest dataset (ALS) containing nearly 640 million points can be segmented within 50 minutes without any pre-processing. Then, we assessed the accuracy of *Vo-Norvana* quantitatively by comparing its modeling results to two existing automated methods as well as manually derived results. The average differences of the position, orientation, and dimension between the models generated from *Vo-Norvana* and manual segmentation are 0.0027 m, 0.0830°, and 0.0009 m, respectively, which indicates minimal difference between the methods. Finally, we demonstrated the versatility of the proposed framework on TLS, MLS, ALS, and UAS-SfM datasets. The results show that *Vo-Norvana* segmented these point clouds effectively and efficiently with relatively consistent parameter settings. Because each parameter has a clear physical meaning, it is straightforward for users to give proper values based on the data quality, target objects, level of detail desired, and other factors. The high computational performance also enables efficient parameter fine-tuning when needed.

Conclusion

This paper introduces a novel point cloud segmentation framework, *Vo-Norvana*, based on a specialized voxelization technique that can preserve geometric details to a large degree. *Vo-Norvana* consists of three primary steps, data organization, Normal Variation Analysis (*Norvana*), and point clustering and provides segment IDs and classes to unorganized point cloud automatically. *Vo-Norvana* was tested on a diverse range of datasets and scenes including terrestrial lidar, mobile lidar, airborne lidar, and UAS-SfM. The data volume ranges from about 1.3 million points in a laboratory setting to nearly 640 million points at a city-wide scale. The segmentation results were evaluated and discussed both qualitatively and quantitatively. Key highlights observed with the proposed approach are as follows:

1. Unlike most existing methods, *Vo-Norvana* is not limited to pre-defined geometric primitives such that it can cope with a variety of shapes in different sizes to complete the segmentation for general purposes. The automated modelling results derived from the segmented point cloud can satisfy most applications.
2. *Vo-Norvana* also can consider the positional uncertainty of the data both locally and globally, reducing improper segmentation results from noise when combining scans or flightlines from different locations.

3. By being designed to effectively take advantage of parallel programming, *Vo-Norvana* consistently achieves a computational performance on the order of hundreds of thousands of points or tens of thousands of voxels per second on a desktop computer. This efficiency holds even when processing hundreds of millions of points that cover an entire city at a time, demonstrating the outstanding scalability of the *Vo-Norvana* framework.
4. The versatility of the proposed approach was proven through extensive tests on point cloud data collected from different scenes (e.g., architecture, sub-urban, urban, industry) using a wide range of systems (e.g., ground and aerial-based, lidar and SfM-based) with relatively consistent parameter settings which does not require extensive fine-tuning.

Although not directly demonstrated in this manuscript due to scope, the point cloud segmentation results can be improved by combining multiple iterations of *Vo-Norvana* with different parameter settings to perform a multi-scale analysis considering the objects of interest and adapting to the variable point density within the scene. Such results can directly benefit semantic segmentation by feeding rich information extracted from each segment determined at several different scales. We are currently utilizing *Vo-Norvana* to enable efficient feature extraction, classification, modelling, and other applications, as well as performing additional quantitative assessments of accuracy. For example, in our recent work, we applied the proposed segmentation approach as a pre-processing tool to separate walls, floors, ceilings, furniture, and other objects for supporting the Scan-to-BIM process including the 2D floor plan generation and 3D modelling (Baru et al., 2022). In the future, we plan to leverage the *Vo-Norvana* framework in developing novel machine learning and deep learning approaches due to its efficiency and scalability.

Data Availability Statement

Some data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request (benchmark, TLS, and UAS-SfM datasets). Some data, models, or code used during the study were provided by a third party (MLS dataset). Direct requests for these materials may be made to the provider as indicated in the Acknowledgements. Some data, models, or code generated or used during the study are available in a repository online in accordance with funder data retention policies (ALS dataset, available at https://geo.nyu.edu/catalog/nyu_2451_38684).

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Nos. CMMI-1351487, OIA- 2040735, and EEC-1937070. This work is also partially supported by the Pacific Northwest Transportation Consortium (PacTrans) and University Venture Development Fund (UVDF) from Oregon State University (OSU). The authors would like to thank Oregon Department of Transportation and Chase Simpson at OSU for collecting and preparing some of the data, as well as Leica Geosystems and David Evans and Associates who provided hardware and software used in this research. CloudComapre was also used for some of the data visualization in this work. The Confederated Tribes of the Grand Ronde graciously provided access and support with the acquisition of the data at the Blue Heron Paper Mill site.

Conflict of Interest

The authors Dr. Che and Dr. Olsen have financial interests in EZDataMD LLC, a company which commercializes the technology related to this research. The conduct, outcomes, or reporting of this research could benefit EZDataMD LLC and could potentially benefit the authors.

References

American Society for Photogrammetry & Remote Sensing (ASPRS), 2019. LAS Specification 1.4-R15. http://www.asprs.org/wp-content/uploads/2019/07/LAS_1_4_r15.pdf (visited on December 30, 2021)

Baru, C., Pozmantier, M., Altintas, I., Baek, S., Cohen, J., Condon, L., Fanti, G., Fernandez, R.C., Jackson, E., Lall, U. and Landman, B., 2022. Enabling AI innovation via data and model sharing: An overview of the NSF Convergence Accelerator Track D. *AI magazine*, *43*(1), pp.93-104.

Bassier, M., Bonduel, M., Van Genechten, B. and Vergauwen, M., 2017. Segmentation of large unstructured point clouds using octree-based region growing and conditional random fields. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*(2W8), pp.25-30.

Bello, S.A., Yu, S., Wang, C., Adam, J.M. and Li, J., 2020. Deep learning on 3D point clouds. *Remote Sensing*, *12*(11), p.1729.

Bresky, R.J., 2016. *The Papermakers: More Than Run of the Mill*. CreateSpace Independent Publishing Platform, California, US.

Che, E. and Olsen, M.J., 2018. Multi-scan segmentation of terrestrial laser scanning data based on normal variation analysis. *ISPRS journal of photogrammetry and remote sensing*, *143*, pp.233-248.

Che, E. and Olsen, M.J., 2019. An efficient framework for mobile lidar trajectory reconstruction and Mo-norvana segmentation. *Remote Sensing*, *11*(7), p.836.

Che, E., Jung, J. and Olsen, M.J., 2019. Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review. *Sensors*, *19*(4), p.810.

Che, E., Olsen, M.J. and Jung, J., 2021a. Efficient segment-based ground filtering and adaptive road detection from mobile light detection and ranging (LiDAR) data. *International Journal of Remote Sensing*, *42*(10), pp.3633-3659.

Che, E., Senogles, A. and Olsen, M.J., 2021b. Vo-SmoG: A Versatile, Smooth Segment-based Ground Filter for Point Clouds via Multi-scale Voxelization. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *8*, pp.59-66.

Dal Poz, A.P. and Yano Ywata, M.S., 2020. Adaptive random sample consensus approach for segmentation of building roof in airborne laser scanning point cloud. *International Journal of Remote Sensing*, *41*(6), pp.2047-2061.

Dong, Z., Yang, B., Hu, P. and 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*, pp.112-133.

Grilli, E., Menna, F. and Remondino, F., 2017. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, p.339.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K. and Pollefeys, M., 2017. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*.

Huang, M., Wei, P. and Liu, X., 2019. An efficient encoding voxel-based segmentation (EVBS) algorithm based on fast adjacent voxel search for point cloud plane segmentation. *Remote Sensing*, *11*(23), p.2727.

Isenburg, M., 2013. LASzip: lossless compression of LiDAR data. *Photogrammetric engineering and remote sensing*, *79*(2), pp.209-217.

Laefer, D.F., Abuwarda, S., Vo, A.V., Truong-Hong, L. and Gharibi, H., 2017. 2015 aerial laser and photogrammetry survey of Dublin city collection record.

Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M.A., Cao, D. and Li, J., 2020. Deep learning for LiDAR point clouds in autonomous driving: a review. *IEEE Transactions on Neural Networks and Learning Systems*.

Lin, Y., Wang, C., Chen, B., Zai, D. and Li, J., 2017. Facet segmentation-based line segment extraction for large-scale point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(9), pp.4839-4854.

Maalek, R., Lichti, D.D. and Ruwanpura, J.Y., 2018. Robust segmentation of planar and linear features of terrestrial laser scanner point clouds acquired from construction sites. *Sensors*, *18*(3), p.819.

Mahmoudabadi, H., Shoaf, T. and Olsen, M.J., 2013, July. Superpixel clustering and planar fit segmentation of 3D LIDAR point clouds. In *2013 Fourth International Conference on Computing for Geospatial Research and Application* (pp. 1-7). IEEE.Matrone, F., Grilli, E., Martini, M., Paolanti, M., Pierdicca, R. and Remondino, F., 2020. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS International Journal of Geo-Information*, *9*(9), p.535.

Ni, H., Lin, X. and Zhang, J., 2017. Classification of ALS point cloud with improved point cloud segmentation and random forests. *Remote Sensing*, *9*(3), p.288.

Olsen, M.J., Roe, G.V., Glennie, C., Persi, F., Reedy, M., Hurwitz, D., Williams, K., Tuss, H., Squellati, A. and Knodler, M., 2013. *Guidelines for the Use of Mobile LIDAR in Transportation Applications* (No. Project 15-44).

Olsen, M.J., Wartman, J., McAlister, M., Mahmoudabadi, H., O'Banion, M.S., Dunham, L. and Cunningham, K., 2015. To fill or not to fill: sensitivity analysis of the influence of resolution and hole filling on point cloud surface modeling and individual rockfall event detection. *Remote Sensing*, *7*(9), pp.12103-12134.

Poux, F. and Billen, R., 2019. Voxel-based 3D point cloud semantic segmentation: Unsupervised geometric and relationship featuring vs deep learning methods. *ISPRS International Journal of Geo-Information*, *8*(5), p.213.

Poux, F., Mattes, C. and Kobbelt, L., 2020. Unsupervised segmentation of indoor 3D point cloud: Application to object-based classification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *44*(W1-2020), pp.111-118.

Vo, A.V., Laefer, D.F. and Byrne, J., 2021. Optimizing Urban LiDAR Flight Path Planning Using a Genetic Algorithm and a Dual Parallel Computing Framework. *Remote Sensing*, *13*(21), p.4437.

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.

Vosselman, G., 2013. Point cloud segmentation for urban scene classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci*, *1*(257-262), p.1.

Wang, K., Wang, T. and Liu, X., 2019. A review: Individual tree species classification using integrated airborne LiDAR and optical imagery with a focus on the urban environment. *Forests*, *10*(1), p.1.

Wang, Q. and Kim, M.K., 2019. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced Engineering Informatics*, *39*, pp.306-319.

Wang, R., 2013. 3D building modeling using images and LiDAR: A review. *International Journal of Image and Data Fusion*, *4*(4), pp.273-292.

Wang, R., Peethambaran, J. and Chen, D., 2018. Lidar point clouds to 3-D urban models $: $ A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(2), pp.606-627.

Wang, X., Chan, T.O., Liu, K., Pan, J., Luo, M., Li, W. and Wei, C., 2020. A robust segmentation framework for closely packed buildings from airborne LiDAR point clouds. *International Journal of Remote Sensing*, *41*(14), pp.5147-5165.

Weinmann, M., Jutzi, B., Hinz, S. and Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, pp.286-304.

Xia, S., Chen, D., Wang, R., Li, J. and Zhang, X., 2020. Geometric primitives in LiDAR point clouds: A review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, pp.685-707.

Xie, Y., Tian, J. and Zhu, X.X., 2020. Linking points with labels in 3D: A review of point cloud semantic segmentation. *IEEE Geoscience and Remote Sensing Magazine*, *8*(4), pp.38-59.

Xu, Y., Tong, X. and Stilla, U., 2021. Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry. *Automation in Construction*, *126*, p.103675.

Yang, J., Kang, Z., Cheng, S., Yang, Z. and Akwensi, P.H., 2020. An individual tree segmentation method based on watershed algorithm and three-dimensional spatial distribution analysis from airborne LiDAR point clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, pp.1055-1067.

Zhang, Y., Yang, W., Liu, X., Wan, Y., Zhu, X. and Tan, Y., 2021. Unsupervised Building Instance Segmentation of Airborne LiDAR Point Clouds for Parallel Reconstruction Analysis. *Remote Sensing*, *13*(6), p.1136.

Zhao, C., Guo, H., Lu, J., Yu, D., Zhou, X. and Lin, Y., 2021. A new approach for roof segmentation from airborne LiDAR point clouds. *Remote Sensing Letters*, *12*(4), pp.377-386.

Zhu, X., Liu, X., Zhang, Y., Wan, Y. and Duan, Y., 2021. Robust 3-D plane segmentation from airborne point clouds based on quasi-a-contrario theory. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *14*, pp.7133-7147.

Table 1. Summary of the proposed point clustering approach for different types of surfaces.

Surface Class	**Neighbor Searching**	**Member *Core* Points**	**Clustering Criteria / Thresholds**	**Segment Criteria**
Smooth	*Core* points from 26 neighboring voxels	*Smooth*	Normal gradient (3 estimations) $T_{\Delta \mathrm{Norm}}$	Number of *core* members $T_{\mathrm{N\_Cores}}$
Rough		*Unclustered Smooth* + *Rough*	Normal gradient (1 estimation) $T_{\Delta \mathrm{Norm}}$	
Invalid		*All remaining*	3D distance	

Table 2. Key information (e.g., dimensions, errors), parameters, and processing efficiency for the datasets used in the experimentation. The efficiency is reported in four metrics: processing time, point-based, voxel-based, and segment-based performance.

Key Information and Parameters	Benchmark	TLS	MLS	ALS	UAS-SfM
Number of scans/flight lines	1	8	1	20	1
Number of points	1,205,600	391,764,131	75,657,595	639,309,547	79,541,367
Number of valid voxels	96,655	18,688,805	23,933,223	105,012,969	17,596,719
Adjusted X extent (m)	3.0	289	1,367	2,818	344
Adjusted Y extent (m)	4.0	290	486	3,095	704
Adjusted Z extent (m)	3.0	44	153	518	60
Voxel size S_v (m)	0.010	0.050	0.050	0.350	0.100
Local error σ_{local} (m)	0.003	0.003	0.010	0.030	0.010
Global error σ_{global} (m)	-	0.006	-	0.030	-
Min segment size T_{N_Cores}	100	50	50	10	10
Max normal gradient $T_{\Delta\text{Norm}}$ (degree)	15	15	15	15	15
Min neighbor $T_{\text{N\_Neighbors}}$	8	8	3	3	3
Max neighbor angle T_{θ} (degree)	90	90	150	150	150
Number of segments	56	16,435	28,490	979,668	18,762
Processing time (s)	3	1,030	633	2,929	347
Point-based performance (million points per second)	0.402	0.380	0.120	0.218	0.229
Voxel-based performance (million valid voxels per second)	0.032	0.018	0.038	0.036	0.051
Segment-based performance (segments per second)	19	16	45	334	156

Table 3. Summary of the quality of the model fitting statistics from the Manual, RANSAC, Norvana, Vo-Norvana (Core) and Vo-Norvana (Full). Note that the Manual and Norvana results were reported in Che & Olsen (2018) while RANSAC modeling was conducted in CloudCompare.

Object	Approach	Error Statistics (m)				# of Points
		Mean	Std. Dev.	Abs. Mean	Abs. Max.	
Plane	*Manual*	0.0000	0.0013	0.0011	0.0047	132,983
	RANSAC	0.0000	0.0014	0.0011	0.0069	133,485
	Norvana	0.0000	0.0013	0.0010	0.0044	126,435
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0014**	**0.0011**	**0.0092**	**6,989**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0014**	**0.0011**	**0.0102**	**132,836**
Sphere1	*Manual*	0.0000	0.0022	0.0018	0.0091	119,872
	RANSAC	0.0000	0.0023	0.0019	0.0089	123,019
	Norvana	0.0000	0.0022	0.0018	0.0091	117,257
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0025**	**0.0021**	**0.0080**	**9,696**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0023**	**0.0019**	**0.0092**	**122,153**
Sphere2	*Manual*	0.0000	0.0010	0.0008	0.0048	40,109
	RANSAC	0.0000	0.0011	0.0008	0.0091	42,132
	Norvana	0.0000	0.0010	0.0007	0.0060	39,200
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0013**	**0.0010**	**0.0056**	**3,660**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0010**	**0.0008**	**0.0055**	**41,897**
Cylinder1	*Manual*	0.0000	0.0002	0.0002	0.0018	4,661
	RANSAC	0.0000	0.0003	0.0002	0.0017	4,625
	Norvana	0.0000	0.0004	0.0003	0.0014	3,595
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0003**	**0.0003**	**0.0015**	**265**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0003**	**0.0002**	**0.0016**	**4,614**
Cylinder2	*Manual*	0.0000	0.0004	0.0003	0.0017	11,890
	RANSAC	0.0000	0.0005	0.0004	0.0027	12.285
	Norvana	0.0000	0.0004	0.0003	0.0017	5,668
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0012**	**0.0006**	**0.0131**	**1,145**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0010**	**0.0005**	**0.0173**	**12,719**
Cone1	*Manual*	0.0000	0.0006	0.0004	0.0028	34,298
	RANSAC	0.0000	0.0006	0.0005	0.0048	35,144
	Norvana	0.0000	0.0005	0.0004	0.0025	30,881
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0007**	**0.0006**	**0.0044**	**2,192**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0006**	**0.0005**	**0.0053**	**35,161**
Cone2	*Manual*	0.0000	0.0006	0.0005	0.0026	15,428
	RANSAC	0.0000	0.0007	0.0005	0.0062	16,108
	Norvana	0.0000	0.0005	0.0004	0.0062	13,329
	***Vo-Norvana* (*Core*)**	**0.0000**	**0.0008**	**0.0006**	**0.0045**	**1,062**
	***Vo-Norvana* (Full)**	**0.0000**	**0.0006**	**0.0005**	**0.0059**	**15,445**

Table 4. Comparison of the modeling results using the RANSAC, Norvana, Vo-Norvana (Core) and Vo-Norvana (Full) compared with the manually extracted results.

Object	Approach	Difference against manual modeling results		
		Position (m)	Orientation (°)	Shape (m)
Plane	*RANSAC*	0.0027	0.0090	-
	Norvana	0.0031	0.0069	-
	***Vo-Norvana* (*Core*)**	**0.0101**	**0.0148**	**-**
	***Vo-Norvana* (Full)**	**0.0036**	**0.0123**	**-**
Sphere1	*RANSAC*	0.0002	-	0.0003
	Norvana	0.0001	-	0.0000
	***Vo-Norvana* (*Core*)**	**0.0014**	**-**	**0.0015**
	***Vo-Norvana* (Full)**	**0.0001**	**-**	**0.0001**
Sphere2	*RANSAC*	0.0002	-	0.0002
	Norvana	0.0004	-	0.0008
	***Vo-Norvana* (*Core*)**	**0.0006**		**-0.0006**
	***Vo-Norvana* (Full)**	**0.0001**	**-**	**-0.0001**
Cylinder1	*RANSAC*	0.0003	0.1365	-0.0005
	Norvana	0.0015	0.7468	-0.0006
	***Vo-Norvana* (*Core*)**	**0.0019**	**0.1146**	**-0.0006**
	***Vo-Norvana* (Full)**	**0.0022**	**0.0845**	**-0.0004**
Cylinder2	*RANSAC*	0.0005	0.0044	0.0000
	Norvana	0.0039	0.0152	0.0021
	***Vo-Norvana* (*Core*)**	**0.0023**	**0.0179**	**0.0009**
	***Vo-Norvana* (Full)**	**0.0023**	**0.0227**	**0.0009**
Cone1	*RANSAC*	0.0093	0.1308	0.0010
	Norvana	0.0105	0.1776	-0.0031
	***Vo-Norvana* (*Core*)**	**0.0246**	**0.3797**	**0.0025**
	***Vo-Norvana* (Full)**	**0.0061**	**0.0889**	**0.0011**
Cone2	*RANSAC*	0.0060	0.2528	0.0042
	Norvana	0.0090	0.1509	0.0000
	***Vo-Norvana* (*Core*)**	**0.0078**	**0.4410**	**0.0025**
	***Vo-Norvana* (Full)**	**0.0045**	**0.2066**	**0.0030**

Figure 1. Workflow of *Vo-Norvana* framework for point cloud segmentation.

Figure 2. Example of the proposed PCA alignment and 3D tiling approach to optimize the partitioning of mobile lidar data

Figure 3. Simplified 2D schematic of the proposed voxelization and subsampling approach.

Figure 4. Schematic illustrating proposed core point Norvana where point #0 (yellow) is the point under analysis while the others are its 8 neighboring core points. The two graphs in the center show the side view and top view of the points in the projected coordinate system of point #0. In the side view, parameter σ (i.e., σ_{local} or σ_{global} depending whether the neighboring point and point #0 are from the same source or not) represents the positioning uncertainty of a point with respect to point #0. From the top view, point #8 is removed from the analysis to reduce the sensitivity of the proposed analysis to the noise. The graph on the right shows the process of generating a mesh where the angle θ at point #0 for each triangle cannot be larger than T_θ.

Figure 5. Growing process for smooth surface and rough surface where point A and B are both classified as smooth surface core points while point D, E, F are classified as rough surface points. In the process of growing from point A to B, two of the three estimations of the normal difference between these two points are shown in the figure where point B' is the adjusted position for point B with the given parameter σ (i.e., σ_{local} or σ_{global}) while the third estimation is obtained in a similar way with $\Delta\mathrm{Norm}'$ by adjusting point A along the normal of point B. When clustering points on smooth and rough surfaces, the estimations of normal differences are compared against the user parameter $T_{\Delta\mathrm{Norm}}$.

Figure 6. Overview of the lidar datasets used in the experiment, including the Benchmark, TLS, MLS, ALS, and UAS SfM/MVS data.

Figure 7. Segmentation results from the benchmark data with RANSAC, QTPS, core points only and full dataset with Vo-Norvana where each randomly assigned color represents a unique segment and unclassified points are colored in black.

Figure 8. Feature classification and segmentation result for TLS data where each distinct color (randomized) represents a unique segment. Unclassified points are colored in black. Subplots show smooth, rough, invalid and unclassified points.

Figure 9. Close-up views of the segmentation result on the TLS dataset where each color represents a unique distinct segment. Unclassified points are colored in black. Close-ups are shown for the building façade (A) to highlight the ability to segment complex architectural details, a busy street (B) with noise from pedestrians and vehicles to highlight the usefulness for noise removal, and segmentation of individual tree canopies (C).

Figure 10. Segmentation result of the MLS testing dataset where each color represents a unique distinct segment. Unclassified points are colored in black. Close-ups show the detailed segmentation of a variety of urban objects including buildings (A and C), poles (A, B, C, and D), powerlines (C and D), signs (A, B, and E), curbs (C, D, and E), and the road surface (A, B, C, D, and E).

Figure 11. Segmentation results for the ALS test dataset where each distinct color represents a unique segment. Unclassified points are colored in black. Locations of closeup views for details shown in Figure 12 - 16 are identified.

Figure 12. Close-up view of the segmented point cloud and reference satellite images near Four Courts (left) and Adam & Eve's Church and St. Audoen's Church (right) where multiple types of roofs are successfully segmented. Numbers show common points between the photograph and point cloud for reference.

Figure 13. Close-up view of the segmented point cloud and reference satellite image near Communications Workers' Union consisting of trees, road, and buildings (roof).

Figure 14. Close-up view of the segmented point cloud and reference street view images at Talbot Memorial Bridge where the pole-like objects in different sizes are segmented into individual objects. The objects highlighted include street lamps (a, b, e) and traffic lights (c, d).

Figure 15. Close-up view of the segmented point cloud and reference satellite image at Pearse Station and St Andrew's Roman Catholic Church where a variety of complex buildings present in the scene.

Figure 16. Close-up view of the segmented point cloud and reference satellite image at Technological University Dublin and Seetec jobpath Bishops Square where the roofs consist of a variety of shapes.

Figure 17. Segmentation result of the UAS-SfM testing data where each distinct color represents a unique segment. Unclassified points are colored in black. The close-up views include: (A) a hydroelectric dam (the photograph is from the source UAS imagery dataset); (B) cylindrical and spherical storage tanks in different sizes; (C) a cylindrical storage tank; (D) various types of building roofs. Numbers show common points between the photograph and point cloud for reference.