Research Repository UCD

Title	Three-dimensional building façade segmentation and opening area detection from point clouds
Authors(s)	Zolanvari, S. M. Iman, Laefer, Debra F., Natanzi, Atteyeh S.
Publication date	2018-09
Publication information	Zolanvari, S. M. Iman, Debra F. Laefer, and Atteyeh S. Natanzi. "Three-Dimensional Building Façade Segmentation and Opening Area Detection from Point Clouds." Elsevier, September 2018. https://doi.org/10.1016/j.isprsjprs.2018.04.004.
Publisher	Elsevier
Item record/more information	http://hdl.handle.net/10197/9476
Publisher's statement	This is the author's version of a work that was accepted for publication in ISPRS Journal of Photogrammetry and Remote Sensing. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in ISPRS Journal of Photogrammetry and Remote Sensing (143 (2018)) https://doi.org/10.1016/j.isprsjprs.2018.04.004
Publisher's version (DOI)	10.1016/j.isprsjprs.2018.04.004

Downloaded 2026-05-02 00:26:46

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)

© Some rights reserved. For more information

Three-dimensional Building Façade Segmentation and Opening Area Detection from Point Clouds *

S. M. Iman Zolanvari [a], Debra F. Laefer [a, b]**, Atteyeh S. Natanzi [a]

[a] School of Civil Engineering, University College Dublin, Ireland

[b] Center for Urban Science and Progress and Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, New York, NY, USA

Abstract:

Laser scanning generates a point cloud from which geometries can be extracted, but most methods struggle to do this automatically, especially for the entirety of an architecturally complex building (as opposed to that of a single façade). To address this issue, this paper introduces the Improved Slicing Method (ISM), an innovative and computationally-efficient method for three-dimensional building segmentation. The method is also able to detect opening boundaries even on roofs (e.g. chimneys), as well as a building's overall outer boundaries using a local density analysis technique. The proposed procedure is validated by its application to two architecturally complex, historic brick buildings. Accuracies of at least 86% were achieved, with computational times of as little as 0.53 sec for detecting features from a data set of 5.0 million points. The accuracy rivalled the current state of the art, while being up to six times faster, and with the further advantage of requiring no manual intervention or reliance on a priori information.

Keywords: Point Cloud Segmentation, Three-Dimensional Model Reconstruction, Feature Detection, Light Detection and Ranging (LiDAR), Laser Scanning

** Corresponding Author at the Center for Urban Science and Progress and the Department of Civil and Urban Engineering, New York University, 370 Jay St., Brooklyn, NY 12011.

Email addresses: debra.laefer@nyu.edu (Debra F. Laefer), iman.zolanvari@ucdconnect.ie (S. M. Iman Zolanvari) website: https://zolanvari.com, atteyeh.natanzi@ucdconnect.ie (Atteyeh S. Natanzi).

* Please note that the presented article is a draft version. You may find the final version at https://doi.org/10.1016/j.isprsjprs.2018.04.004

1 INTRODUCTION

The ability to automatically generate three-dimensional (3D) urban façade models from point clouds has gained considerable importance across many fields including autonomous navigation (Zhang et al., 2016), vegetation management (Höfle et al., 2012), virtual reality creation (Bui et al., 2016) and environmental modelling (Singh and Laefer, 2015). Light Detection and Ranging (LiDAR) is a common remote sensing technology used to generate the point clouds that serve as input data sets for such models, as the technology has the ability to collect millions of points rapidly as x-, y- and z- positional coordinates. However, processing such a point cloud into a usable 3D solid model for computational analysis (Hinks et al., 2013) or Building Information Modelling (Laefer and Truong-Hong, 2017) continues to pose significant challenges and has largely been done only with individual facades. This is especially true if the building includes non-rectilinear features (e.g. curved windows) and complex geometric elements (e.g. cornices).

Automated, 3D building model generation is highly relevant to many civil engineering applications, since, the vast majority of existing urban structures lack measured drawings, and the cost of producing them through traditional surveying methods is cost-prohibitive when more than a handful of structures are involved (Laefer, 2016). While LiDAR offers a rapid and cost-effective alternative solution for documenting the external geometries of existing structures, the raw data are only positional in nature. Thus, an automated means for segmentation and feature extraction is required for further application-oriented usage. There are also the additional requirements of the final output needing to be both geometrically accurate and in a file format compatible with the selected modelling software (e.g. for finite element modelling in civil engineering). To automatically generate such models, the overall boundaries of each building must be established, and the opening areas across the façade must be located. These two issues are particularly challenging if: (1) a façade has protrusions; (2) there are non-rectilinear windows; (3) the roof-level features intersect non-orthogonally with the roof; (4) the use application requires multiple façades to be modelled; (5) the data set is very large; and/or (6) there is a desire to use the use model for multiple domains.

Therefore, for any solution to be viable at a city-scale, it must be both scalable and robust. As will be discussed in the next section, most current approaches struggle with these issues.

This paper is organized as follows. Section 2 describes related work that has been done for point cloud segmentation and feature extraction. Section 3 describes the theoretical framework for the study and develops the methodological pipeline. Section 4 tests the proposed method on two highly complex building and shows the experimental results. Then a comparative analysis is run to benchmark the proposed method against a highly cited new technique. Subsequently, Section 5 presents a sensitivity analysis and discusses the influence of average density and other variables on the results. Section 6 expresses the conclusions drawn from the experimental results and the comparative analysis.

2 RELATED WORKS

A raw LiDAR point cloud begins as an unclassified group of points. They are often massive in size [e.g. a quarter of a billion points per square kilometre (DataCite, 2016)]. Arguably what is needed for 3D reconstructions for engineering-based applications is some form of segmentation followed by feature extraction. While, the concepts of building segmentation and feature extraction are relatively similar, as both processes aim to find relationships among the points, they are distinctive activities. Thus, this paper considers these two concepts separately by defining segmentation as the process of clustering a group of points belonging to a single surface or region at a building scale (e.g. a single building façade), while feature extraction is herein defined as identifying specific, smaller building features (e.g. chimneys and windows) typically from previously segmented patches.

2.1 Segmentation

Segmentation is often used to help classify portions of a point cloud. The action uses one or more criteria to group points into subsets (e.g. points belonging to the same plane, having a similar density, or being in a particular orientation). As described in the

following paragraphs, three important strategies have been commonly adopted for this step: (a) geometric fitting; (b) region growing; and (c) clustering.

Geometric fitting-based approaches mostly use variants of the Random Sample Consensus (RANSAC) methods, as initially introduced by Fischler and Bolles (1981), to segment building façades by fitting planes or lines into denser point cloud areas (e.g. commonly seen with walls). As an example, Schnabel et al. (2007) employed RANSAC to fit several candidate shapes into the point cloud. For that purpose, an approximate surface normal for each point was computed, and then the number of compatible points for the candidate shape as a standard score function was counted. Although the method can detect the approximate shape of the scanned 3D object even in the presence of up to 50% outliers, the shapes are strictly limited (i.e. planes, spheres, cylinders, cones and tori), and the technique would not able to detect the shape of free-form objects. Boulaassal et al. (2009) used the fitting-based approach to extract planar clusters contours. RANSAC has been widely used in this capacity for more than a decade (e.g. Bendels et al., 2006; Awwad et al., 2010), as it is simple and applicable to many building styles and types. Yang and Förstner (2010), also applied a RANSAC-based algorithm integrated with a minimum description length to manually define the number of fitted planes in a point cloud. One of the RANSAC-based method limitations is that a tolerance threshold for the distance between the fitting plane and the searching points is always required. Defining this threshold value causes the methods to be case dependent. Another geometric fitting-based method is the Hough Transform (Hough 1962), which was introduced to recognise lines in images. Later this method was extended into 3D to identify positions of arbitrary planes (Maas and Vosselman, 1999; Vosselman and Dijkman, 2001), cylinders (Tarsha-Kurdi et al., 2007), and spheres (Rabbani and Heuvel, 2005). These methods extract surfaces with a relatively high level of success, as long as significant protrusions or intricate details are not part of a building's exterior architecture.

Region-growing is another common approach. This method divides the point cloud into large surface patches by grouping adjacent points or voxels (3D cells). Then,

coarse groups identified with similar normal vectors or other residual values (e.g. distance to neighbours) can be refined. Next, the method considers only the points inside those groups/voxels that are related to a common feature (e.g. normal vector directions) and merges them as a segmented part. An early example employing region-growing was done by Woo et al. (2002) to simplify a larger dataset into smaller voxels via an octree-based method. The result generated a 3D surface for the target object by calculating all normal vectors. The neighbour voxels were then merged into leaf nodes as seed cells, and then these seeds were grown until the deviation of the voxels' normal vectors were less than the threshold. While the result can correctly estimate the surface of objects from a point cloud, the method needs to define two manually assigned thresholds for the voxel size and for the standard deviation of the voxels' normal vectors. The process of calculating normal vectors for all cells and other additional checks for having correct normal direction can be computationally expensive. To accelerate the procedure, Vo et al. (2015) introduced an octree-based algorithm that quickly extracts different planes of a building's façades from both Terrestrial Laser Scanning (TLS) and Aerial Laser Scanning (ALS) data. However, the scalability of that method has yet to be established. In general, while region-growing has been used widely in many state-of-the-art segmentation works (e.g. Deschaud and Goulette, 2010; Wang and Tseng, 2011; Nurunnabi et al., 2012), use of the approach always depends on at least one predefined criterion, which challenges its robustness as a universal solution.

Clustering is another major segmentation strategy. Clustering is the process of grouping points with similar feature vectors into a single cluster separate from points with dissimilar feature vectors. The method has been an integral part of many algorithms [e.g. Hierarchical Clustering (Pauly et al., 2002), the k-means algorithm (Shi et al., 2011), the fuzzy C-means algorithm (Lerma and Biosca, 2005; Biosca and Lerma, 2008)]. For example, Filin and Pfeifer (2006) performed a cluster analysis in a feature space for ALS data. For that, they employed a slope adaptive neighbourhood method that is based on a distance criterion and the geometrical content of the point cloud to detect the planar surfaces in the dataset. The method could successfully

segment the coarse planes (e.g. roofs and ground plane around the building) from the airborne data, however, it is limited to segmenting only planar shapes.

Although the clustering method is similar to region growing and both are based on grouping points under common constraints, one of the major advantages of clustering over the region growing method is that no seed point(s) or regions are needed to initiate the characterization or grouping. Another benefit is that, unlike fitting-based methods, clustering can segment multi-planar and 3D façades. However, clustering is computationally expensive for 3D data sets and highly influenced by data density and quality, because the method must determine whether or not each point satisfies the clustering criteria. In addition, the approach may fail to properly segment edges, as edge points may meet the requirements of more than one cluster, especially if the criteria are based on point density or distribution (e.g. Aljumaily et al., 2015; Aljumaily et al., 2017).

2.2 Feature extraction

After coarsely segmenting a building's façade planes and roof(s), feature extraction is typically needed for generating sufficiently detailed solid models. Recent contributions in the area mostly focus on detecting different opening areas (i.e. windows and door) and determining overall façade boundaries. For that purpose, three main strategies are reviewed: (1) pattern recognition; (2) edge-based approaches; and (3) hole-based approaches.

Pattern recognition involves searching for repetitive and symmetrical elements. The main concept is similar to the geometric fitting approach in segmentation. However, the main goal in segmentation is to segregate the major planes like buildings or their façades, while smaller features (e.g. windows, doors, or bricks) are not considered. For example, a shape grammar for buildings was introduced by Stiny and Mitchell (1978) and refined a few years later (Stiny, 1982) to define a set of rules for generating architectural villa ground plans. A shape grammar was also used by Koning and Eizenberg (1981) and later by Flemming (1987) to categorise a building's shape with

regard to the position of specific architectural features (e.g. the fireplace, stairs, kitchen) mostly for architectural design purposes. Subsequently, Han and Zhu (2009) used a graph grammar to extract planar rectangular features. They projected several rectangular frames of various sizes and shapes onto a given image to detect rectangular objects. More recently, Müller et al. (2007) presented a shape grammar that used image analysis techniques to identify repetitive windows from a single, high-quality photograph of a façade. Shape grammars have also been applied to LiDAR data sets (e.g. Ramamoorthi and Arvo, 1999). As part of that line of work, Mayer and Reznik (2006) proposed an automatic approach to detect windows and doors based on object orientation, with respect to coordinate axes and an implicit model shape. In contrast, Pauly et al. (2008) used a computational framework for detecting repeated 3D geometric structures. For that, they introduced an aggregation procedure that mathematically estimated rotational, cylindrical grids, complicated helices, and spirals forms. That method was also notable for its ability to estimate missing data in a 3D volumetric model. More recently, Friedman and Stamos (2013) presented an automatic, real-time registration procedure that operates online, while the scanner collects the data. For this, they employed periodically repeating features after finding the building's major planes. While successful even with low-resolution data, the robustness of the method is questionable, as it is highly reliant on there being repetitive, symmetric elements.

Another major strategy involves edge-based approaches. Gross and Thoennessen (2006), introduced a method mostly for airborne laser scanning data that established boundary points of features (particularly roofs) on the edges and corner of planes. For this purpose, they defined a sphere for each point and searched for all neighbours that could possibly be populated inside the sphere. Then, they calculated the covariance matrix (eigenvalues and the eigenvectors) of the sphere. The sphere size depended on the data density (i.e. a larger sphere is needed for sparser data). Additional features were investigated with respect to the position, scale, and rotation of the spheres. Similarly, Sotoodeh (2006), detected outliers from laser point clouds based on the local density of each feature. Furthermore, Wani and Arabnia (2003) utilised parallel

algorithms on multi-ring networks of images to detect boundary edges. The general strategy behind these techniques is highly dependent on having non-uniform data densities, non-level planes, and/or differences in contrast (in the case of image-based pixels).

A third well-established strategy uses hole-based approaches. An early example is a work by Pu and Vosselman (2006; 2009) who developed ontologies for building features with respect to size, position, direction, topology, and data density. Their method classified the elements directionally (vertical versus non-vertical); positionally (on the ground, wall, or roof); and topologically (e.g. connected to roof or wall). The approach relies heavily on adding priori information specific to each case, which limits its robustness. In addition, the technique also employs Delaunay Triangulation (DT) to generate a Triangular Irregular Network (TIN), which makes it computationally expensive for large data sets. Subsequently, Truong-Hong et al. (2011) introduced another hole-based approached in which DT generates a triangular network from which holes are detected. In that author's later work (Truong-Hong et al., 2013), an angle criterion with a k-d tree was used to detect boundary points and window openings, and then a voxel-based technique was developed with the same aim (Truong-Hong and Laefer, 2014). Both methods rely upon some semi-manual segmentation and were applied to relatively simple, rectilinear and planar facades. More recently, Zolanvari and Laefer (2016), proposed the Slicing Method (SM) method to detect openings, as well as overall boundary points of façades. This fully automatic method uses RANSAC for segmenting façades. It then slices the extracted façade into imaginary strips along its principal axes, projects the points into one dimension, and searches for gaps. Validation using independently produced measured drawings for six historic brick buildings demonstrated that, without any manual intervention, an accuracy of nearly 100% for simple buildings and in excess of 93% for highly decorative facades with curved, asymmetrical window openings with inside crossbars could be achieved. Additionally, computational times were less than 3 secs for high-density data up to 14,698 pts/m^2 (2.6 million points overall). However, the Slicing Method as presented was limited to a single, pre-segmented façade and could not incorporate sloped roofs or

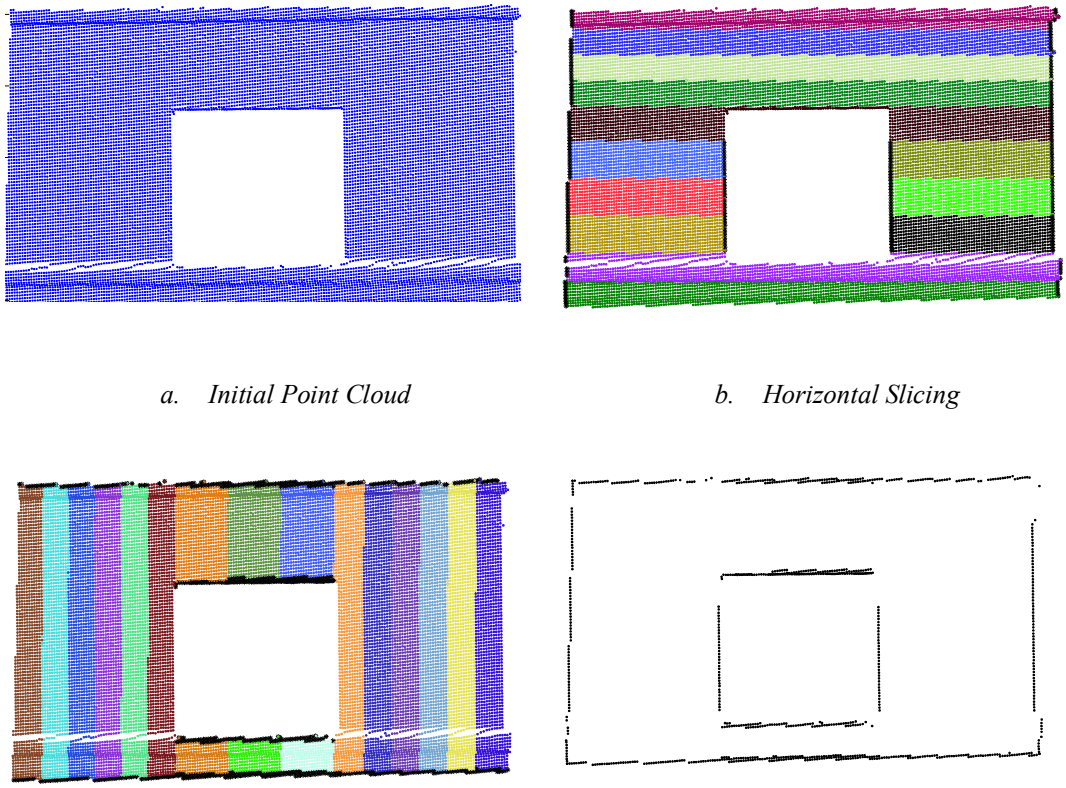
entire structures. The following work extends that method to make it fully three-dimensional and addresses complex roof/façade interfaces, something heretofore unachievable with any existing method.

3 PROPOSED METHOD

3.1 Introduction of Improved Slicing Method

Zolanvari and Laefer (2016) introduced an automatic method to extract building features with SM, which was demonstrated to be accurate and fast, but it only detected openings on a single, segmented façade. To extend those capabilities fully to 3D, including various roof types (e.g. hip, gambrel, and conical), this paper introduces the Improved Slicing Method (ISM) including a new means to segment building sides (e.g. different façades and roofs planes), to detect openings on the roofs, and to establish overall boundary points of complicated roofs including chimneys. Furthermore, this paper also presents a new strategy that would enable feature extraction without segmentation when only two sides of a building are visible in the data set (e.g. a corner building). As the ISM builds on the SM approach, a brief introduction of the latter containing the initial building blocks for the new method is provided herein. In this section, some sample data sets are selected to illustrate the proposed method in a step-by-step manner on real point cloud.

First, the planar RANSAC algorithm is applied to the roughly segmented point cloud of each planar façade, and then points that are outside of the planes (e.g. vegetation, noise, and interior objects that were acquired through the window) are removed. Next, the SM cuts the segmented façade horizontally or vertically (Figure 1). A data gap appears where the laser beam does not return a signal due to window glass or other openings. These gaps divide each slice into clusters. Thus, the number of clusters in each slice is equal to one more than the number of gaps. Notably, this step is helpful in the removal of non-structural elements such as small objects inside detected holes (e.g. window crossbars).



a. Initial Point Cloud *b. Horizontal Slicing*

c. Vertical Slicing *d. Final Extracted Boundary Points*

Figure 1. SM Feature Detecting Boundary Around a Single Window

The process's final step extracts extreme points of each cluster to generate the boundaries of the overall façade and its opening areas (Figure 1d). The end points of each cluster are harvested based on a preselected threshold with respect to the required level of output detail (e.g. 1 cm). The three steps are repeated orthogonally (i.e. horizontally or vertically based on the initial directional decision). Choosing to increase the number of slices (resulting in thinner slices) may increase the boundary points and, thus, the accuracy of the method within the constraints of data density and coverage area. However, the number of slices also influences the required computational time. This will be discussed in more detail in the discussion section.

3.2 ISM Segmentation

In this paper, segmentation of individual buildings is considered. Segregating individual buildings from the larger fabric of the cityscape remains a major

impediment to a fully automated pipeline as even the most successful techniques for segregating portions of a point cloud into individual building clusters achieve less than 70% for dense urban areas (Truong-Hong and Laefer, 2015).

The segmentation procedure of a single 3D quadrilateral building has four major steps (Figure 2). Step 1 defines local coordinates and sorts the data. This step sets the y-axis roofward and x-axis along the longest orthogonal façade; more details about defining local coordinates are provided in Zolanvari and Laefer (2016). The longest façade is the one that has the maximum magnitude of the Oriented Minimum Bounding Box (Freeman and Shapira, 1975). In this step, all points are sorted along the y-axis from the ground upward. This accelerates the next step (slicing). The longest side of each façade is identified as a pre-processing step by applying a Principal Component Analysis (PCA). The PCA procedure has five steps: (a) Step 1 finds the centre of the point cloud by calculating the mean of all X, Y and Z values of the points. Then temporarily translates all the points to the calculated centre; (b) Step 2 calculates the covariance matrix of the data set by multiplying the points location matrix in 3D by their transposed matrix and then divides by number of the points; (c) Step 3 computes the principal components by employing the singular value decomposition of the covariance matrix; (d) Step 4 transforms the original points regarding the computed principal components matrix in 3D; and (e) Step 5 performs a final check to find which dimension of the transformed data is larger and takes that as the x-axis.

Step 2 involves slicing the building horizontally. Each slice represents a unique, imaginary band passing through a building. This is done along the façade's principal axes that serve as the local coordinates. In other words, each slice is an intersection between a horizontal plane and the building (Figure 3).

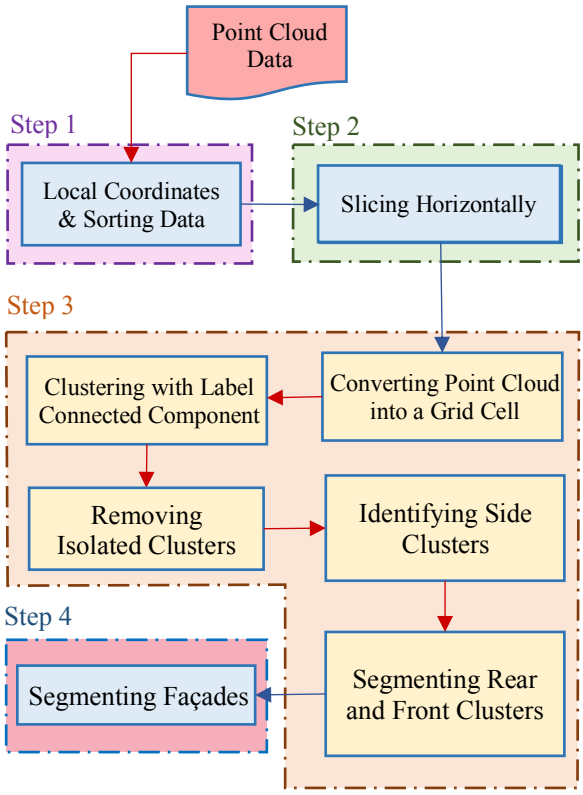


Figure 2. Flowchart of SM Segmentation Method

Step 3 transforms the point cloud into a binary format to simplify the procedure and reduce the computation time by grouping the neighbouring points. This step includes five sub-steps. First, points are converted into 10x10 cm grid cells (Figure 4a). The grid size can be varied in accordance with the required level of output detail. In this paper, a 10 cm grid cell dimension was selected, as all protrusions and details of the study case were larger than that dimension. The grid is a two-dimensional (2D) plane, and all of a slice's affiliated points are projected on it. To ensure that all points are within the defined cells (as opposed to along the grid's border), fractional coordinates are rounded. Although this temporary transformation is usually very small (and not easily illustrated in the figure), it is later discarded so as not to affect the final accuracy.

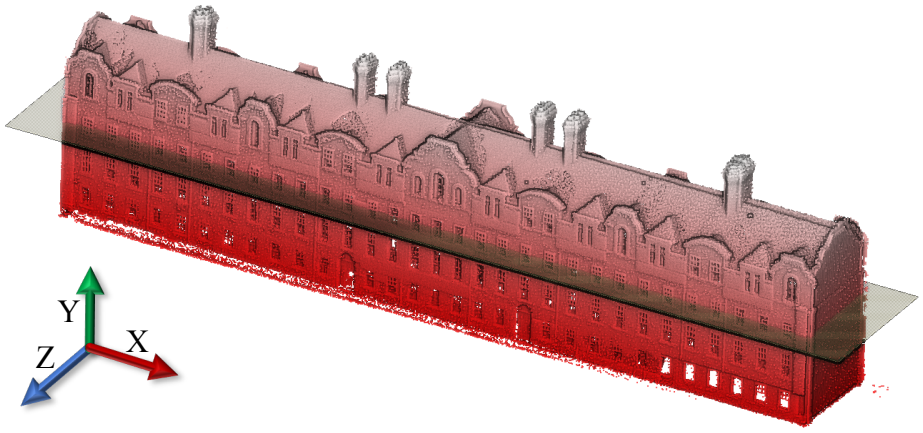
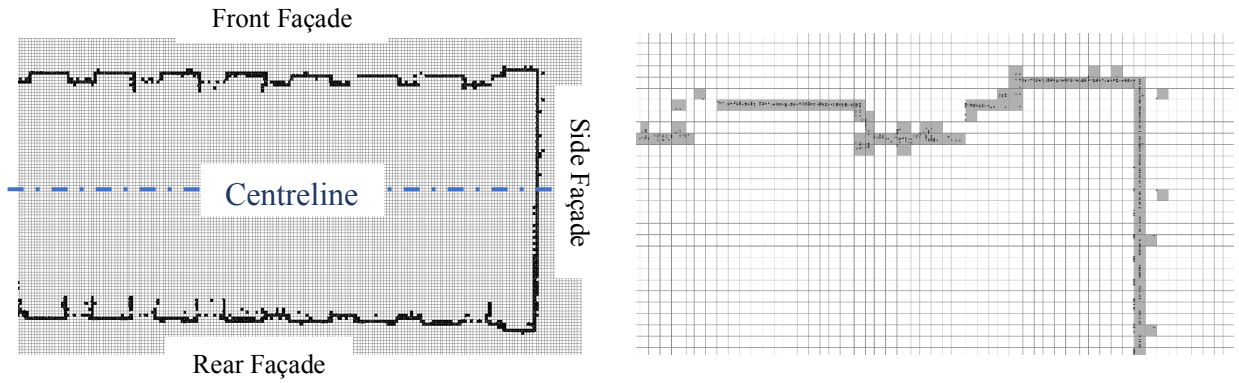


Figure 3. Rubrics Building Point Cloud with the Sample Horizontal Slice

After projection, every cell that includes at least one point is assigned a value of one and coloured grey (Figure 4b). Unpopulated cells are each assigned a zero value and coloured white (Figure 4b). In the next sub-step, each group of cells is clustered through the application of the Label Connected Components (LCC) method, as introduced by *Vincent and Soille* (1991), which discriminates between isolated and connected clusters (Figure 4c). This is followed by the third sub-step, the removal of all isolated cells, as most of them represent outliers. The clusters are then categorised as horizontal (along the x-axis) or lateral (along the depth of the z-axis) in the x-z plane by comparing the number of rows and columns of each cluster that obtained from LCC method (e.g. horizontal clusters include more rows than columns in the grid).

The fourth sub-step identifies Slender Side Clusters. This is done by designating the first and last columns of populated cells as the side clusters. Then in the fifth sub-step, the remaining clusters are split and designated as front or rear clusters with respect to an imaginary centreline (Figure 4a). In other words, after removing the two side clusters, what remains is considered as the front and rear clusters (Figure 4). Figure 4d shows the right-side cluster and the front cluster at the corner of the building.



a. Selected Slice is Brought into a Binary Grid with the Position of the Various Façades Before Segmentation

b. Top Right Corner of the Featured Slice. Grey Cells Represent One Values, and White Cells are Zero Values

c. Unconnected Clusters via LCC Shown in Different Colours

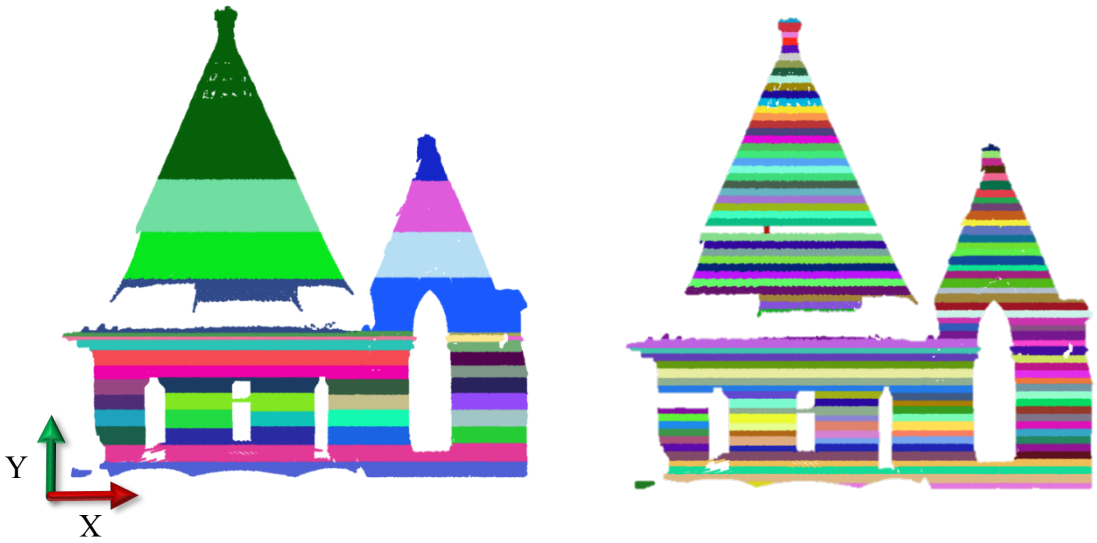
d. Segmented Front and Side Clusters

Figure 4. Transforming and Processing the Point Cloud into a Binary Format

Subsequently, in Step 4 the algorithm repeats the previous steps for all slices. This segments the building into four parts by attaching all similar clusters (i.e. sides, front and rear) together from bottom to top. Therefore, the method can successfully segment all four façades of a quadrilateral building, irrespective of roof irregularity (e.g. gabled roofs, skylights, solar panel, water tanks, and equipment rooms).

Slicing the data is a primary component of both ISM segmentation and feature extraction (as will be discussed in Section 3.3). In the original SM algorithm, the data were sliced into strips based on the projection of the data. In such an approach, if a sloped roof portion was included, the strips were no longer uniform in their coverage

area (e.g. Figure 5a). This would result in a highly uneven number of points per slice. In the ISM algorithm proposed herein, that problem is overcome through an alternative slicing procedure that sorts the data along the relevant axis (e.g. y-axis for horizontal slicing in the x-y plane) and selects an equal slice thickness along that related axis. In other words, the horizontal slicing occurs along the y-axis and the vertical slicing along the x-axis. This amended procedure provides equally thick slices for all portions of a building (Figure 5b). This accelerates the process; the previous SM method searched all points to check whether or not each appeared in a particular slice. In contrast, the new ISM employs an index to pre-locate the points.



a. Old Slicing Data in SM *b. Improved Slicing Data in ISM*

Figure 5. Slicing Data in Two Different Ways

3.3 ISM Feature Extraction

Using the proposed ISM, this paper improves the extraction of opening areas and building boundary points, while also accelerating the process. This is essential to the automatic generation of solid models for various applications.

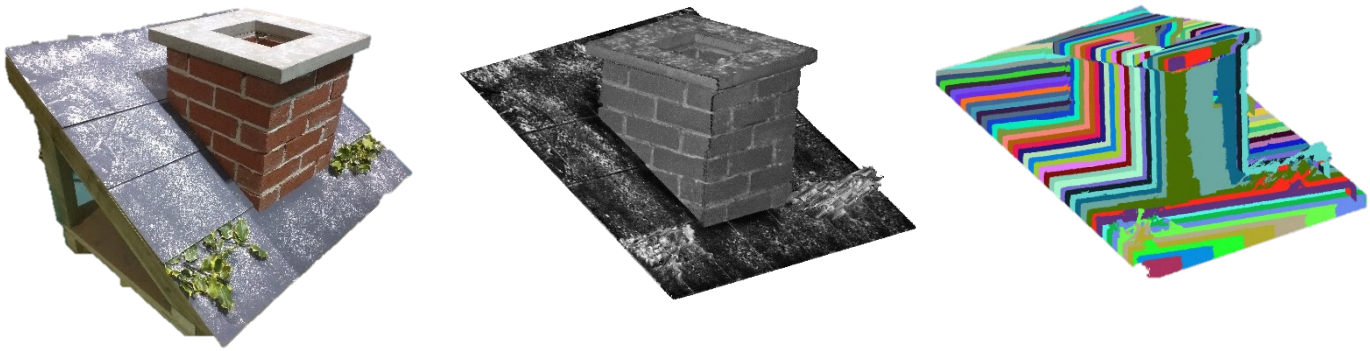
ISM feature extraction is applicable to two kinds of 3D cases: Scenario A (an entire building) and Scenario B [where only a pair of connected façades from a single building is visible (e.g. a corner building)]. Scenario A uses the output of the ISM

segmentation and sequentially processes each façade with a portion of the roof, as mentioned earlier. In Scenario B, however, the segmentation step is skipped, as will be explained below.

3.3.1 Scenario A: Entire Building

Scenario A is the general case and applicable to planar and non-planar facades segmented as described above or to any other method (e.g. clustering, region growing or geometric fitting). The feature extraction algorithm of the ISM has two main parts: (Part 1) façade and window boundary point extraction on the vertical façades, and (Part 2) roof feature (e.g. chimney) extraction.

In Part 1, while most of the desired features would be detectable with strictly horizontal and vertical slices in the x-y plane of the primary façade, not all are visible. Thus, the horizontal and vertical slices are extended to the top of the roof to detect any sloped features or items embedded in the sloped portion of the roof. This requires a third set of slices — this time along the z-axis in the x-z plane. These additional slices can detect any sloped opening located on the roof that is not visible in the x-y view. For example, the slices on the x-z plane can extract the chimneys including their interior openings (Figure 6); such features are generally only available via aerial (as opposed to ground-based) data capture.

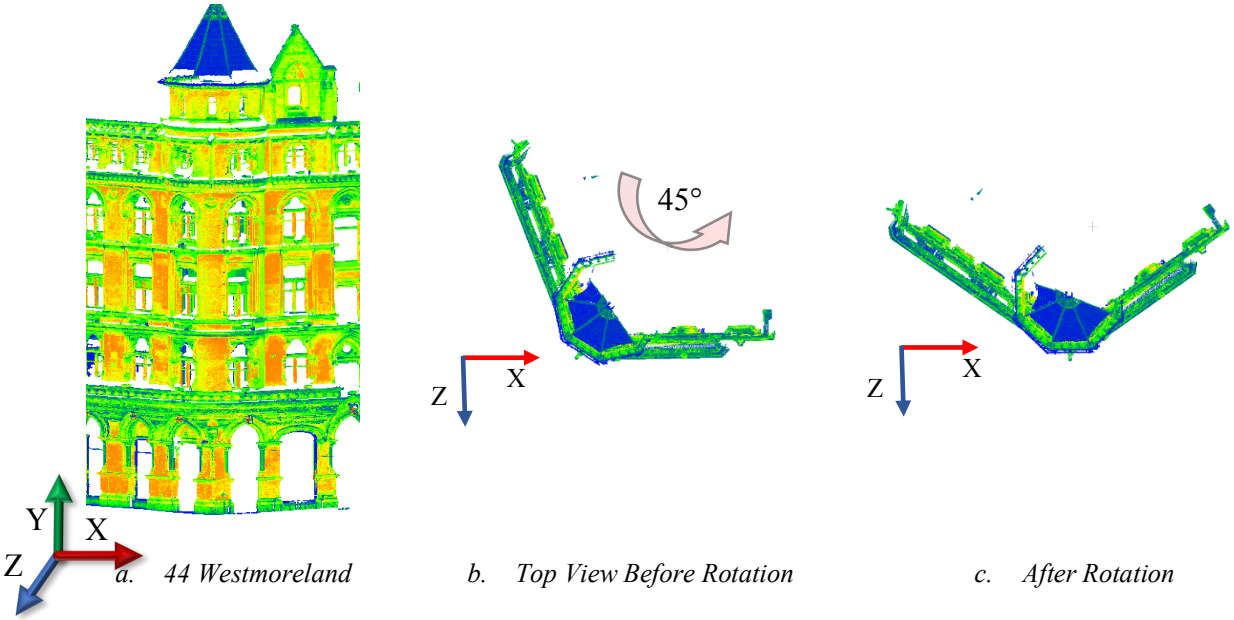


a. Roof Features Image *b. Raw Point Cloud* *c. Sliced on x-z Plane*

Figure 6. Slicing the Roof Features on x-z Plane along the z-axis

3.3.2 *Scenario B: Connected Façade*

Scenario B is a special case and applicable when there are only two visible facades or when a building's curvature does not enable a clear orthogonal cutting of the façade into two planes (e.g. Figure 7a). For those cases, this paper introduces an alternative strategy in which the ISM directly detects openings and overall boundary points of a building from the raw point cloud without a segmentation step. In this case, instead of defining a local x-axis along one façade (Figure 7b), the x-axis is rotated ±45° around the y-axis. This enables the horizontal and vertical slices to be projected onto the x- and y-axis. From this, the ISM detects the extreme points of each projected slice (Figure 7c). This strategy is applicable to all kinds of connected, adjacent façades that meet the ISM feature detection criteria after projection of the façade on its front plane (e.g. x-y plane), as long as the openings are not be covered. If those criteria are met, the rotation strategy can be employed, even if the angle of the two adjacent façades is not exactly 90°, which may occur due to the presence architectural features (e.g. drainage pipelines or other protrusions). As long as all holes on the two façades are visible from the centre of the angle, the ISM feature detection can extract the openings. Otherwise, each façade should first be segmented (e.g. with planar RANSAC), with the ISM feature detection applied subsequently.



a. 44 Westmoreland *b. Top View Before Rotation* *c. After Rotation*

Figure 7. The Raw Point Cloud (Case B) Before and After Rotation

4 EXPERIMENTAL RESULTS

4.1 DATA SPECIFICATION

To show the effectiveness of the ISM, it is applied to the Rubrics (Case A), a large masonry building in Dublin, Ireland. Case A has an area of ~877 m^2 along its west and east façades. The building has a highly complex roof shape, rounded doors, and prominent window frames. This structure is located at Trinity College Dublin. Built around 1700 A.D., the Rubrics is one of Dublin's oldest buildings (Casey, 2005). The final data set includes 20,880,021 points derived by merging an aerial and a terrestrial LiDAR scan. The aerial data (Laefer et al., 2017) were collected with a RIEGL Q680i scanner using the multiple overlap strategy as outlined in Hinks et al. (2009), which resulted in a final projected ground projection density of 225 pts/m^2. The terrestrial scan was obtained with a Leica P20 unit. The merged data set for the Rubrics building was used to produce a high-density data set with minimum façade occlusions. While the nature of ALS and TLS data are similar, the TLS more easily provides points on vertical planes (e.g. façade of buildings) from the street level. In contrast, ALS data mostly collect data of the horizontal planes (e.g. building roofs), with lower density on the vertical facades (compared to TLS data). The method was implemented with a MacBook Pro® computer, 2.9GHz dual-core, Intel® Core™ i5 processor, 8GB 2133MHz memory with a 512GB SSD1 hard disk drive on MATLAB® R2017a and employed general Math and Image Processing Toolboxes; all reported performance metrics are in part influenced by the specific hardware configuration on which the algorithm is run.

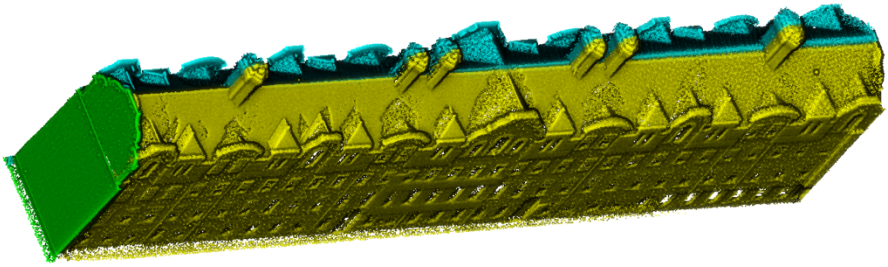


Figure 8. ISM Segmentation on the Rubrics Building (Case A) with Each Colour Representing a Segmented Part

Figure 8 shows the segmented parts of the Rubrics building with ISM. In this case, the building was segmented into four parts. Each segmented part was then used separately as input for the ISM feature extraction portion (Table 1). Since, local point density influences computation time and accuracy, each segmented part is shown separately. However, identifying density for such complex parts is challenging, as the data are highly heterogeneous in both the individual and merged data sets. Thus, in this paper, density values were obtained by a planar projection of the data onto the face of a bounding box obtained as the minimum-volume enclosing cuboid oriented to align with the main planes of the building. Therefore, the surrounding area of the data was roughly achieved by calculating the rectangular area of each face of the bounding box, and the average data density was obtained by dividing the average number of points of each segmented part by the projected rectangular area. The density ranged from 7,000 to 188,000 pts/m^2.

Table 1. Data Input for ISM Segmentation for Case A

Segmented Parts	Projected Area (m^2)			Number of Points	Average Density (k-pts/m^2)		
	x-y	x-z	y-z		x-y	x-z	y-z
West	320.03	1333.30	87.93	9,629,725	30	7	110
East	312.91	1336.00	86.03	9,271,381	30	7	108
North	6.23	10.87	142.07	981,332	158	90	7
South	5.32	9.32	142.43	997,583	188	107	7

Figure 9 displays one segmented part from the results that were shown from the Figure 8 model (either the east or west part that is coloured in yellow and blue, respectively) after the ISM feature extraction was applied in only 1 directions. Specifically, slices through the roof area become quite wide, which may preclude finding smaller openings and features. By also slicing the building along the z-x plane, this problem is avoided, as shown in Figure 10.

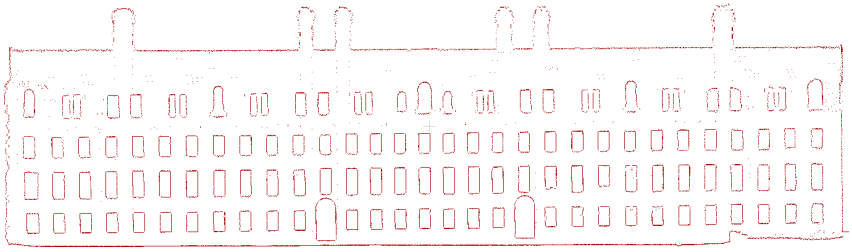


Figure 9. Detected Boundary Points without Slices in z-x Plane (Case A)

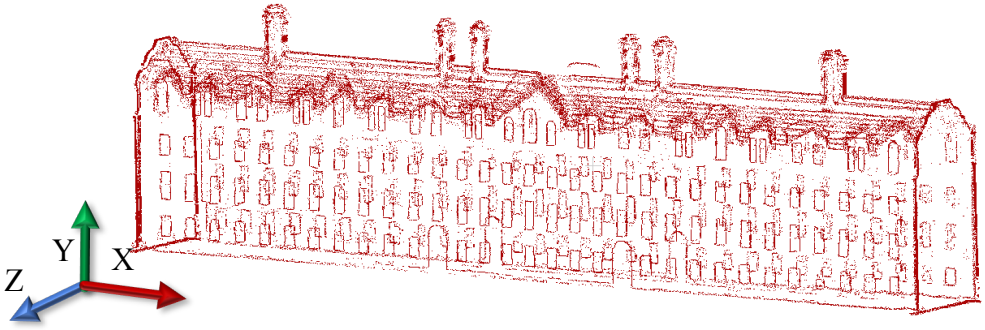


Figure 10. Boundary Points of all Windows of the Rubrics Building (Case A), as well as the Edges

Table 2. Data Input for Case B

Data	Area			Number of Points	Average Density (k-pts/m^2)		
	x-y	x-z	y-z		x-y	x-z	y-z
Case B	88.91	258.03	211.64	5,731,729	65	22	27

To demonstrate how the ISM feature extraction algorithm works in Case B with two connected façades, a very complex building in Dublin Ireland's city centre was selected (Case B, 44 Westmoreland) The building has a conical roof, rounded corners, unsymmetrical windows and highly detailed architectural features (Figure 11). The acquired point cloud contained 5,731,739 points and was obtained exclusively from a Leica P-20 terrestrial laser scanner.

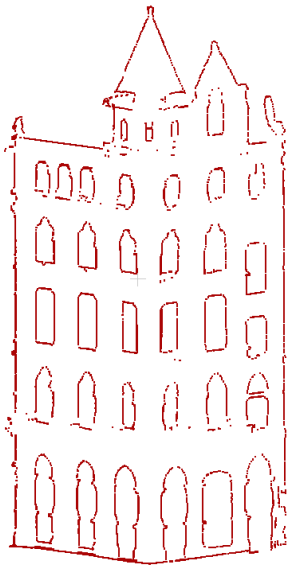


Figure 11. Extracted Boundary Points for Case B

4.2 EVALUATION RESULT ACCURACY

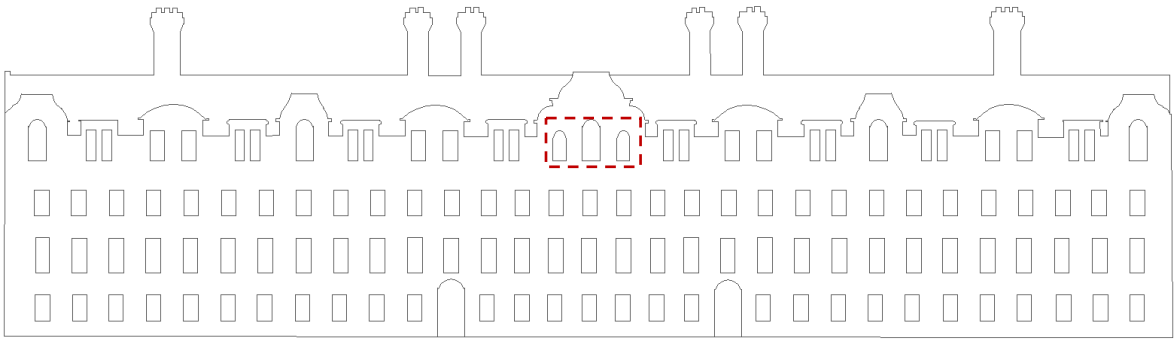
4.2.1 Boundary Detection Evaluation

To evaluate the accuracy of the proposed procedure, all boundary points were overlaid onto an independently derived set of measured drawings (referred to herein as the reference model). However, as the detected boundary points are in three-dimensions, they needed to be projected onto 2D planes for calculating differences in areas. For that purpose, in Case A, all detected boundary points of each segmented part were projected onto the largest face of their bounding box. As such, the west and east sides of the building were projected on the x-y plane (Figure 13a), and the south and north sides were projected onto the y-z plane.

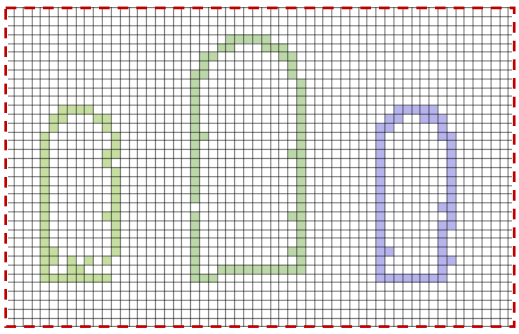
After projecting the results onto the 2D planes, boundary points were clustered using the LCC method (Figure 12b). For each group of clustered points (e.g. windows and doors), a circumscribed polygon was substituted as generated with convex hulls (Figure 12c). The overall façade boundary points were connected to each other using a concave hull algorithm. Next, all circumscribed polygons (of openings and the overall boundaries) were imported into AutoCAD 2016 and overlaid onto the reference model. The actual areas obtained were derived from the overall boundaries that were created

from the concave hull minus the opening areas. Finally, the accuracy was defined by calculating the area of the substituted polygons divided by the area shown on the reference model. More details of this evaluation procedure are available elsewhere (Zolanvari and Laefer, 2016). In the non-segmented Case B, the results and measured drawing were cut in half and projected only onto the largest bounding box faces (Figure 13). The Feature Extraction Accuracy (σ) is formulated in Eq. (1) where n is number of detected clustered opening and a_i is the area of the i_{th} detected clustered circumscribed polygons and the a'_i is the corresponding area at the reference measured drawing.

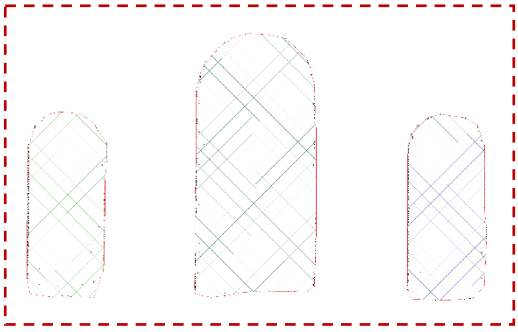
$$\sigma = \left| \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} a'_i} \right| \times 100$$



a. Rubrics Building CAD Drawing

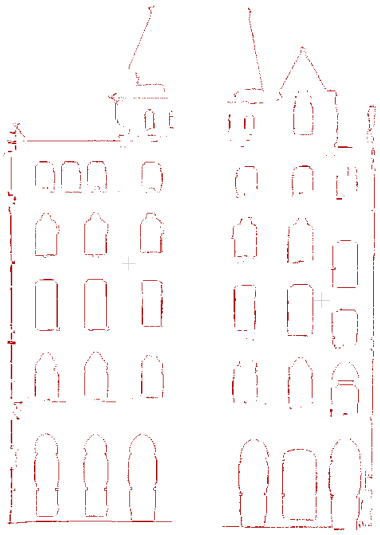


b. Clustered Boundaries with LCC



c. Surrounded Area of Openings

Figure 12. Case A (Rubrics Bldg.)



a. Left Side b. Right Side

Figure 13. Case B Projected Boundary Points

Table 3 reports the segmentation time for Case A and for the ISM feature extraction for both Cases A and B. The time needed for importing, reading, and plotting the data into the algorithm was excluded. Accuracy ranged from 86% to 93% and averaged 88% across the two cases. Feature extraction times were less than 1 sec for each of the 4 parts of Case A and 0.53 sec for all of Case B. Thus, the technique was able to process between 413k points per second (pts/sec) for segmentation and 8,773K to 10,814 pts/sec for extracting features. While the accuracy was quite high for such complicated and decorative buildings, the technique is not perfect. However, part of the accuracy loss is due to imperfections in the initial data (e.g. self-shadow occlusions), as previously discussed in Zolanvari and Laefer (2016).

Table 3. Computation Time for ISM Segmentation and Feature Extraction

Case	Total Available Points	Feature Extraction Time (sec)				Segmentation Time (sec)	Total Time (sec)	Feature Extraction Accuracy (%)			
		West	East	North	South			West	East	North	South
A	20,880,021	0.94	0.91	0.27	0.26	48.14	50.52	88.2	86.9	93.1	91.4
B	5,731,739	0.53				-	0.53	86.1			

In Figure 14, the processing time of the feature extraction algorithm is presented with respect to the number of points to be processed. The exponential fitted regression curve shows the best equation to the processing time as the coefficient of determination (denoted by R-squared) value which is explained variation / total variation, is nearly 1. The trendline equation in this figure shows the stability of the algorithm. Therefore, the processing time is expected to increase exponentially in a larger dataset.

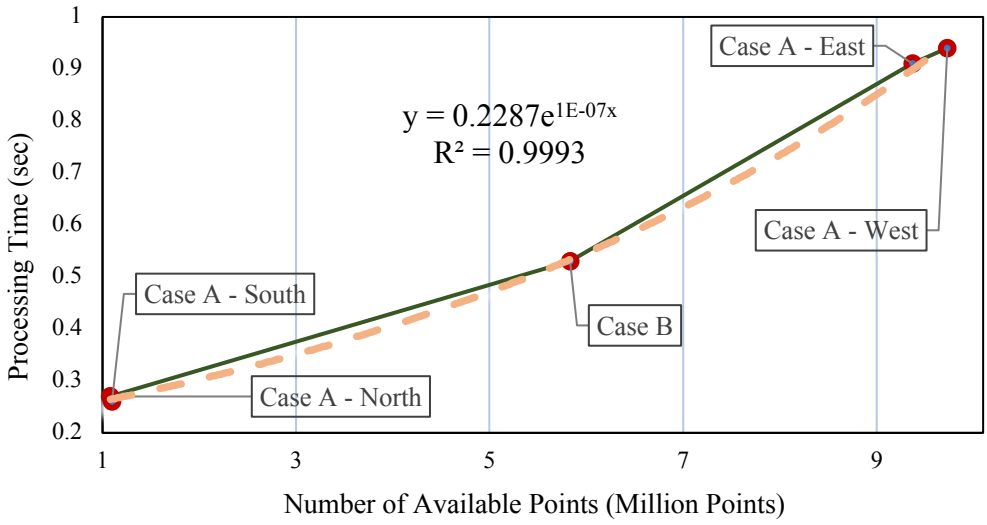


Figure 14. Feature Detection Processing Time with Respect to the Number of Available Slices.

4.3 Comparative Analysis

To demonstrate the value of the proposed ISM approach, it is compared to a novel and highly cited region-growing algorithm that employs an octree (Vo et al., 2015). That region-growing method segments three-dimensional point clouds in a series of coarse-to-fine patches. That algorithm applies an octree-based voxelization process to extract coarse patches by defining neighbour groups to compute the local surface properties (e.g. voxel's normal vector). The identified patches are then passed through a refinement step that incorporates semantic-based feature criteria. This requires manual selection of the voxel size, angular divergence level, and residual value threshold. The input parameters and processing requirements are shown in Table 5. The angular value and the residual value are considered as 0.25 radians and 5 cm, respectively. The voxel size is 80 cm and 20 cm for the Cases A and B, respectively.

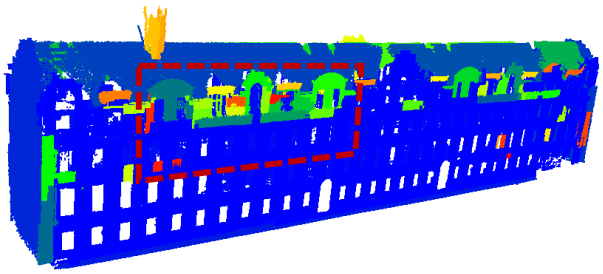
Table 5. Input Parameters and Output Results

Data	Region-growing Algorithm						Proposed ISM Feature Extraction Processing Time (sec)
	Angular Value (radian)	Residual Value (cm)	Voxel Size (cm)	Number of Points	Number of Detected Patches	Processing Time (sec)	
Case A	0.25	5	80	4,587,949	57	15.6	2.38
Case B	0.25	5	20	5,099,442	255	9.45	0.53

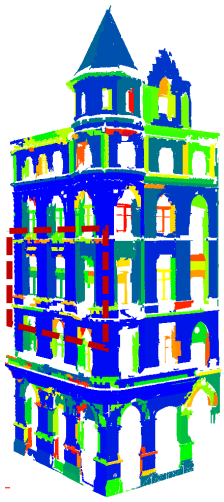
The reported computation time includes only the main segmentation and was based on processing on a computer with the specifications described in section 4.1. To accelerate the process, the point cloud was downsampled to 4.3 million points for Case A, and 1.5 million points for Case B. This resulted in a 1 cm distance between the points in each case. While ISM considers the original data set without downsampling and the processing time is recorded for the initial data. The results are shown in Figure 15. In Case A, the region-growing algorithm succeeded in segmenting the largest patches on the façade and the roof (shown in blue in Figure 15a). However, the total number of segmented patches was 57 and required 15.6 sec. All patches were segmented according to their normal vector orientation and the other aforementioned parameters as per Vo et al. (2015). The region-growing algorithm could not segment windows and doors (Figure 15b). In contrast, the ISM could detect all openings with a minimum accuracy of 86.9% in only 2.38 sec (Figure 15c).

In Case B (Figure 15d), the region-growing algorithm completed segmentation in 9.45 sec. However, the magnified portion of Case B (shown in the dashed box in Figure 15d and 17e) demonstrates that segmenting a point cloud using a cutting-edge region-growing method cannot represent the correct location of the windows or their surrounding frames. In contrast, the ISM (see Figure 15f) detected the boundary points of the features with an 86.1% accuracy in only 0.53 sec.

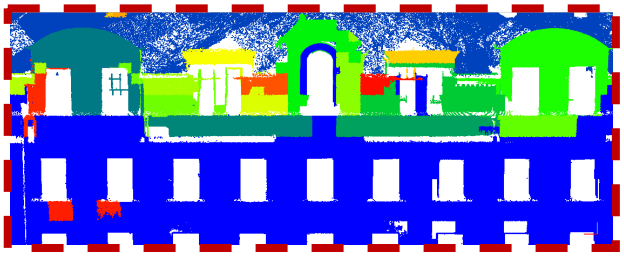
The primary purpose of the region-growing algorithm was segmenting the façade. The input parameters were manually selected by the algorithm's authors (Vo et al., 2015); further optimization of the method and thresholds may be possible but is outside the scope of this paper. Figure 15 shows that segmenting such an architecturally complex



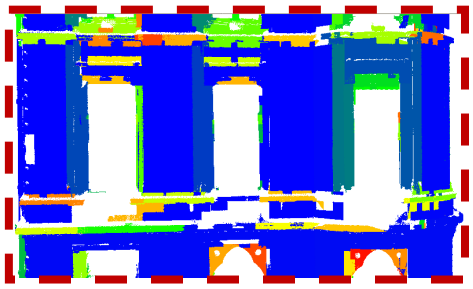
a. Case A: Segmented Result Using the Region-Growing Algorithm



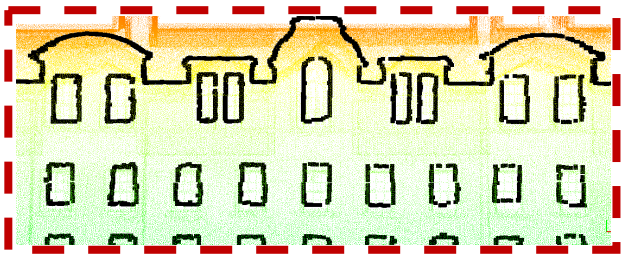
d. Case B: Segmented Result Using the Region-Growing Algorithm



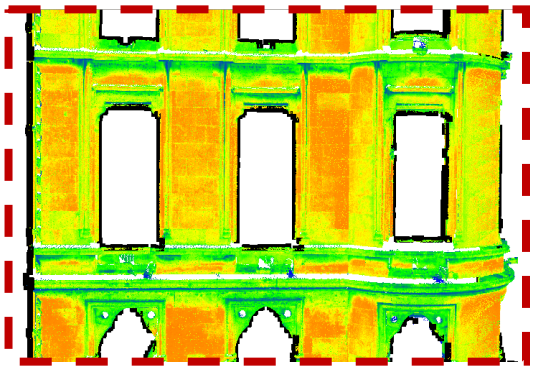
b. Segmented Case A Using the Region-Growing Algorithm (the Magnified Dashed Box in Figure 15a)



e. Segmented Case B Using the Region-Growing Algorithm (the Magnified Dashed Box in Figure 15b)



c. Detected Boundary Using ISM Algorithm in Case A



f. Detected Boundary Using ISM Algorithm on Case B

Figure 15. Octree-Based Region-growing Segmentation and the Corresponding ISM Result

façade can be a challenging task, particularly for detecting openings. All colours in the Figure 15a, b, d and e were randomly selected to show the individual, segmented patches. For example, the region-growing algorithm incorrectly detects patches at the top of Case A (Figures 15a and 15b) and produces several extra patches at the protrusions around the windows in Case B (Figures 15d and 15e). Furthermore, in order to evaluate the region-growing algorithm, manual segmentation is needed [as was done in the method's initial presentation in Vo et al. (2015)]. Finally, the processing time is up to 17 times longer than the proposed ISM method (depending upon the data set). In the Figure 15c and 15f, the raw data colour is only for visualisation and they show elevation and intensity of points. Also, the opening points are shown in black.

In general, the ISM offers three main advantages compared to methods that use a hierarchical volume approach (e.g. voxel model). First, all hierarchical methods may lose some level of accuracy depending upon their voxel size. Second, processing data for conversion into a hierarchical form may need a substantial amount of memory depending upon the point cloud size. Third, reverting the hierarchical volume back into the initial point format cannot be done in a manner to obtain an exact similar variant model as the original topology, thereby losing the surface information. In contrast, the Slicing Method uses the simplification strategies from 3D to 1D that minimise the required computational resources without losing any accuracy of the initial data set.

5 DISCUSSION

This section provides a sensitivity study for the ISM algorithm and then benchmarks it against a popular new technique by Vo et al. (2015) based on a voxelized region growing approach.

5.1 Sensitivity Study

5.1.1 Slice Selection

Selecting the ideal number of slices must be based on a combination of (a) data density and (b) building dimensions. Specifically, larger façades and high-density data both

require extra slices. In addition, the number of slices depends on the relative location of the scanner to the building and the building's shape and size (especially corners). While defining a unique global formula for the ideal number of slices is not currently possible, in the original SM algorithm a sensitivity study showed that with an average data density of 2-14 k-pts/m^2 a slice thickness of 7 cm for horizontal and 4 cm for vertical slices was the most accurate in detecting the boundary points in the minimum amount of time. This was deemed as the optimum thickness. Although the optimum slice thickness can be varied based on the required level of detail, the sensitivity analysis of the original SM algorithm also showed that the slice thickness is not the most influential factor in SM. Therefore, this paper uses a similar configuration as was proposed in the SM where the number of slices is based on the aforementioned spacing thickness. As the buildings in Cases A and B were of similar heights, they have a similar number of horizontal slices (280 and 345 of 7 cm, respectively). However, as the buildings differed greatly in length, Case A had 1800 vertical slices for the east and west facades and 220 for the north and south ones, while Case B had 300 vertical slices (all based on a 4 cm thickness).

Table 4. Number of Utilised Slices in ISM Algorithm

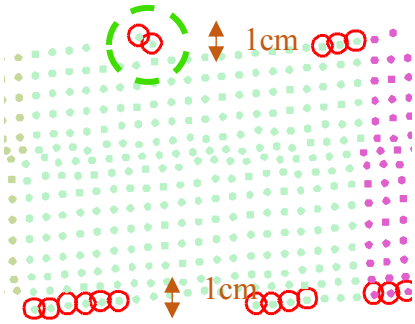
Data	Horizontal Slice Thickness (cm)	Number of Horizontal Slices	Vertical Slice Thickness (cm)	Number of Vertical Slices	
				West and East	North and South
Case A	7	285	4	1800	220
Case B	7	345	4	300	

5.1.2 Boundary Selection

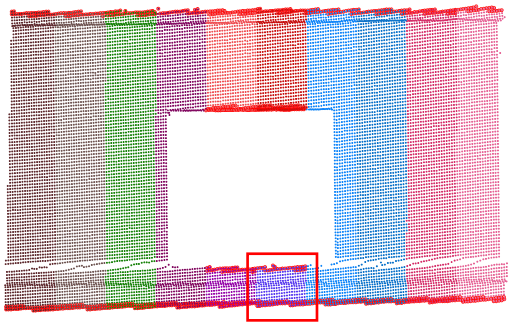
An important input parameter for the ISM is the Extreme Selected Points Width (ESPW). The original SM algorithm employed a 1 cm ESPW to harvest the clusters of the extreme points (on the edges of both sides of each cluster). A larger ESPW value provides a thicker border line of boundary points, while a smaller number may result in the loss of some boundary points and may have a negative impact on accuracy. An example of an insufficiently large ESPW (e.g. 1 cm) is shown in Figure 16a, with a close-up in Figure 16b, where the inclined orientation of the data points can cause

points to be out of alignment with other border points, such as those denoted within the green dashed circle (Figure 16). These can cause false results, if the algorithm selects them instead of the border points. To overcome this issue, one solution is an overly large ESPW (e.g. 3.5 cm) as shown in Figure 16c and d, which extracts more points along the boundary. However, some of the ESPW results can be detected from the inner layer of the boundary and not necessarily the outermost portion of an outline's points. Another solution is reducing the slice thickness where these non-aligned points occur (Fig 16e and f).

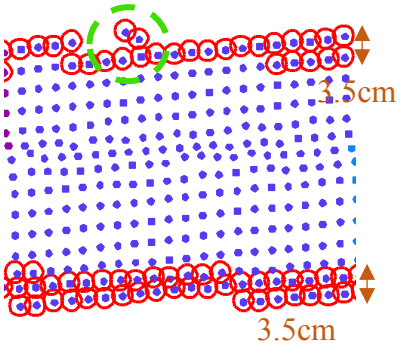
(a) 10 Slices with 1 cm EPSW



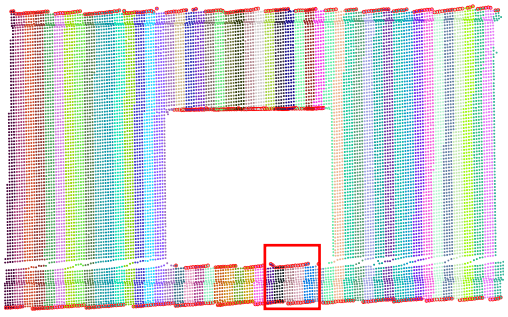
(b) Close-up of sub-image (a)



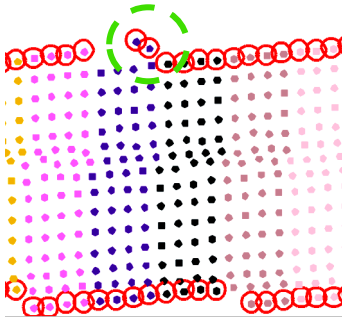
(d) 10 Slices with 3.5 cm EPSW



(e) Close-up of sub-image (d)



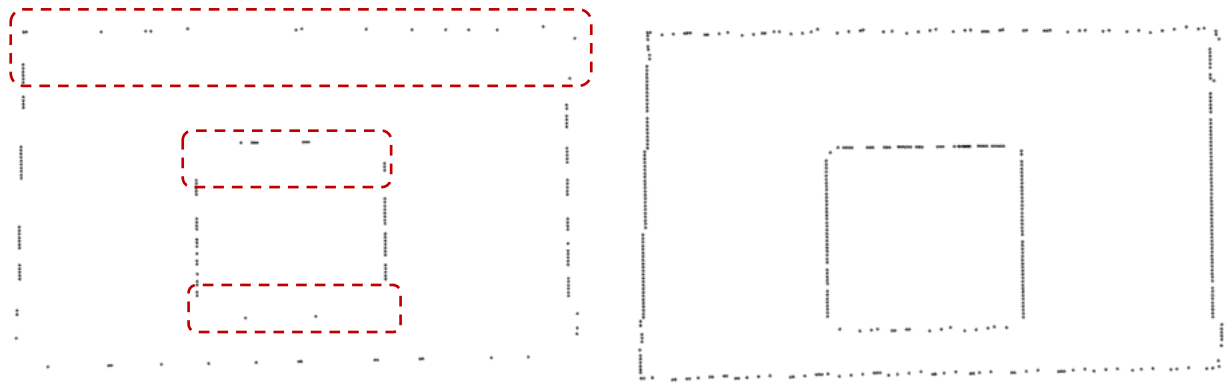
(f) 50 Slices with 1 cm ESPW



(g) Close-up of sub-image (f)

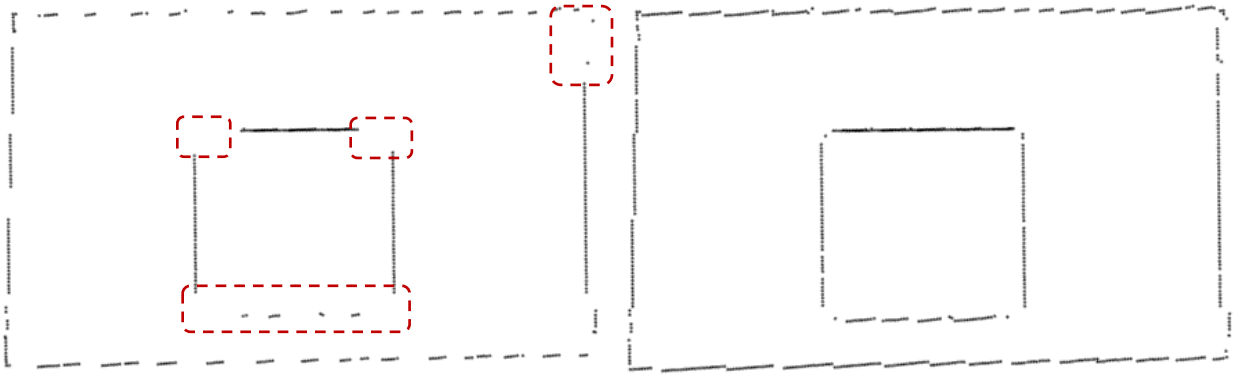
Figure 16. Detected Edge Points with Various Slice Thickness and an ESPW of an Isolated Window

To discuss the effect of selecting different ESPW ranges and the numbers of slices on the results, several configurations were chosen for the simple window in Figure 16. ESPW thicknesses of 0.5 cm, 1 cm, and 4.5 cm were selected with either 10 or 50 slices as shown in Figure 17. Figure 17a shows the poor capture of the windows' sides due to choosing an overly small ESPW (0.5 cm for 10 slices), especially where there is a higher potential to include extruded points (as shown in the dashed boxes).



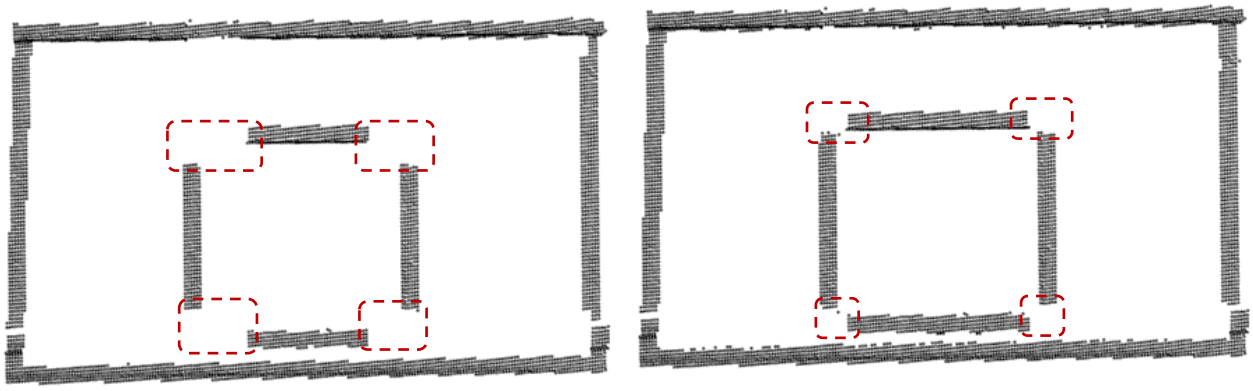
a. 10 Slices with 0.5 cm ESPW

b. 50 Slices with 0.5 cm ESPW



c. 10 Slices with 1 cm ESPW

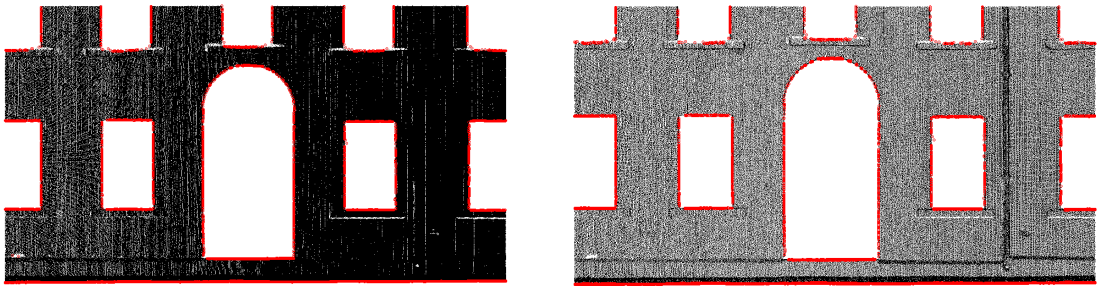
d. 50 Slices with 1cm ESPW



e. 10 Slices with 4.5 cm ESPW *f. 50 Slices with 4.5 cm ESPW*

Figure 17. Extracted Boundary Points of Sample Window with Various Slice Numbers and ESPW

Although Figure 17b shows that increasing the number of slices to 50 can slightly improve the results, incorrect boundary selection remains as an issue, even with a thicker ESPW (Figure 17c). Additionally, Figure 17d shows that choosing 1cm ESPW with 50 slices was the best configuration and provided the clearest results. Furthermore, increasing the ESPW thickness to 4.5 cm did not necessarily improve the results and again included incorrect boundary points especially at the corners (Figures 17e and 17f, for 10 and 50 slices, respectively). Therefore, while a minimum ESPW (e.g. 1 cm for most data sets) was needed, the algorithm detected more precise boundary corner points by increasing number of slices, as opposed to increasing the ESPW thickness.



a. 100%, 6940 pts/m^2 with Minimum Distance of 5 mm

b. ~50%, 3297 pts/m^2 with Minimum Distance of 10 mm

c. ~5%, 420 pts/m^2 with Minimum Distance of 50 mm

d. ~1%, 83 pts/m^2 with Minimum Distance of 100 mm

Figure 18. Extracted Boundaries with Different Average Densities

5.1.2.1 Point Density Effect

To investigate the point density effect, the average data density was reduced from the maximum captured density. For the east facade of Case A, this was done by reducing the original point density of 6,940 pts/m^2 to 50%, 5%, and 1% (3297, 420, and 83 pts/m^2, respectively). This changed the average spacing from 5 mm in the original to 10 mm, 50 mm, and 100 mm, respectively. This resulted in a proportional decrease in the number of extracted points (marked in red) [Figure 18]. The two-order of magnitude reduction in data density did not, however, significantly compromise the quality of the core output, as demonstrated by the fact that none of the results contained incorrect boundary points, however, lower density would potentially compromise the final boundary detection accuracy, as fewer points would be available for boundary edge and corners detection, which is likely to lead to less accurate feature identification in the form of less sharp edges. Therefore, the final boundary resolution is, of course, related to the average density, as a higher point density coupled with a larger number of

slices can provide the maximum resolution of boundary and its corner. Obviously, If the average density of point cloud is too low (e.g. the 83 pts/m^2 shown in Figure 18d), simply increasing the number of slices does not improve the boundary resolution.

6 CONCLUSIONS

The paper introduced the Improved Slicing Method (ISM) as an extension to the initial Slicing Method (SM) for fully 3D processing point cloud and was developed to segment quadrilateral buildings into their main façades and detect complicated, free-form openings and overall boundaries including out-of-plane protrusions (e.g. roof features) for complex urban structures with non-trivial roof shapes.

This paper also introduced a strategy for applying the ISM feature extraction component without segmentation where only two façades are visible, as occurs on corner lots. The accompanying sensitivity study showed that by increasing number of slices, as opposed to increasing the collection thickness at the slice ends, better corner boundaries can be obtained. The method is notable for its ability to work at relatively sparse data densities, in this case as little as 87 pts/m^2 or about 40%, thus showing the potentially high impact of this work for city-scale modelling. While only laser scanning point cloud data has been used for developing the technique, it is potentially applicable to imagery-based point clouds, as well.

When compared to a novel and highly cited new region-growing algorithm (Vo et al., 2015), the results were at least as accurate (at 86% accuracy) with no a priori information nor manual intervention and as much as 6 times faster for point clouds containing approximately 5 million points, thereby overcoming the incorrect segmentation that occurs when geometric criteria (e.g. the normal vector, a residual value) are used for architecturally complex buildings.

ACKNOWLEDGMENTS

This work was sponsored by European Research Council grant ERC-2012-StG_20111012" RETURN – Rethinking Tunnelling in Urban Neighbourhoods" Project 30786. The authors thank Anh-Vu Vo for applying the Octree-based region-growing (Vo et al., 2015) on Cases A and B in the discussion.

REFERENCES

Aljumaily, H. Laefer, D. Cuadra, L., 2015. A Big Data Approach for 3D Building Extraction from Aerial Laser Scanning. J. Computing in Civil Eng. 30 (3), 04015049. http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000524.

Aljumaily, H., Laefer, D., Cuadra, D., 2017, in press. Spatial Big Data Mining Based on Clustering Approach. Journal of Computing in Civil Engineering CPENG-2067.

Awwad, T.M., Zhu, Q., Du, Z., Zhang, Y., 2010. An improved segmentation approach for planar surfaces from unstructured 3D point clouds. The Photogrammetric Record 25 (129), 5-23.

Bendels, G. H., Schnabel, R., Klein, R., 2006. Detecting Holes in Point Set Surfaces. Journal of WSCG 14, 89-96.

Biosca, J.M., Lerma, J.L., 2008. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. ISPRS Journal of Photogrammetry and Remote Sensing 63 (1), 84-98.

Boulaassal, H., Landes, T., Grussenmeyer, P., 2009. Automatic extraction of planar clusters and their contours on building façades recorded by terrestrial laser scanner. International Journal of Architectural Computing 7 (1), 1-20.

Bui, G., Morago, B., Le, T., Karsch, K., Lu, Z., Duan, Y., 2016. Integrating videos with LIDAR scans for virtual reality. In: Höllerer, T., Interrante, V., Lécuyer, A., Suma, E. (Eds.), Proceedings of Virtual Reality (VR) 2016, IEEE, Greenville, SC, 19-23 March, pp. 161-162.

Casey, C., 2005. Dublin: The City Within the Grand and Royal Canals and the Circular Road with the Phoenix Park, vol. 3. Yale University Press, New Haven.

DataCite, 2016. Sample Lidar data, https://goo.gl/BvfNp1. (Accessed 20 December, 2016)

Deschaud, J.-E., Goulette., F., 2010. A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing. In:

Proceedings of 3D Processing Visualization and Transmission Conference (3DPVT2010), 3DV, Paris, France, 17-20 May.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24 (6), 381-395.

Filin, S. and Pfeifer, N., 2006. Segmentation of airborne laser scanning data using a slope adaptive neighborhood. ISPRS journal of Photogrammetry and Remote Sensing, 60(2), pp.71-80.

Flemming, U., 1987. More than the sum of parts: the grammar of Queen Anne houses. Environment and Planning B: Planning and Design 14 (3), 323-350.

Freeman, H. and Shapira, R., 1975. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, *18*(7), pp.409-413.

Friedman, S., Stamos, I., 2013. Online Detection of Repeated Structures in Point Clouds of Urban Scenes for Compression and Registration. International Journal of Computer Vision 102, 112-128.

Gross, H., Thoennessen, U., 2006. Extraction of lines from laser point clouds. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3), 86-91.

Han, F., Zhu, S.C., 2009. Bottom-up/top-down image parsing with attribute grammar. IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (1), 59-73.

Hinks, T., Carr, H., Laefer, D., 2009. Flight optimization algorithms for aerial LiDAR capture for urban infrastructure model eneration. Journal of Computing in Civil Engineering 23 (6), 330-339. http://dx.doi.org/10.1061/(ASCE)0887-3801(2009)23:6(330).

Hinks, T., Carr, H., Truong-Hong, L., Laefer, D.F., 2013. Point cloud data conversion into solid models via point-based voxelization. Journal of Surveying Engineering 139 (2), 72-83. http://dx.doi.org/10.1061/(ASCE)SU.1943-5428.0000097.

Höfle, B., Hollaus, M., Hagenauer, J., 2012. Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne LiDAR data.

ISPRS Journal of Photogrammetry and Remote Sensing 67, 134-147. http://dx.doi.org/10.1016/j.isprsjprs.2011.12.003

Hough P.V.C., 1962. Method and means for recognizing complex patterns. U.S. Patent 3069654.

Koning, H., Eizenberg, J., 1981. The language of the prairie: Frank Lloyd Wright's prairie houses. Environment and planning B: planning and design 8 (3), 295-323.

Laefer, D. (2016). New City-Scale Damage Prediction Methods during Urban Tunnelling. In: Van Balen, K., Verstrynge, E. (Eds.), Proceedings of Structural Analysis of Historical Constructions, SAHC, Leuven, Belgium, 13-15 September, pp. 15-22.

[data set] Laefer, D.F., Abuwarda, S., Vo, A.-V., Truong-Hong, L., Gharibi, H., 2017. 2015 Aerial Laser and Photogrammetry Survey of Dublin City Collection Record. NYU Spatial Data Repository, http://dx.doi.org/10.17609/N8MQ0N

Laefer, D.F., Truong-Hong, L., 2017. Toward automatic generation of 3D steel structures for building information modelling. Automation in Construction 74, 66-77.

Lerma, J.L., Biosca, J.M. 2005. Segmentation and filtering of laser scanner data for cultural heritage. In: Proceedings of CIPA 2005 XX International Symposium, ICOMOS/ISPRS, 26 September-01 October, Torino, Italy.

Maas, H.G., Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data, ISPRS Journal of Photogrammetry and Remote Sensing, 54 (2-3), 153-163. http://dx.doi.org/10.1016/S0924-2716(99)00004-0

Mayer, H., Reznik, S., 2006. Building facade interpretation from uncalibrated wide-baseline image sequences. ISPRS Journal of Photogrammetry and Remote Sensing 61 (6), 371-380.

Müller, P., Zeng, G., Wonka, P., Van Gool, L., 2007. Image-based procedural modeling of facades. ACM Transactions on Graphics (TOG) 26 (3), 85-95.

Nurunnabi, A., Belton, D., West, G. 2012. Robust segmentation in laser scanning 3D point cloud data. In: Proceedings of 2012 International Conference on Digital

Image Computing Techniques and Applications (DICTA), IEEE, 3-5 December Fremantle, Western Australia, pp. 1-8.

Pauly, M., Gross, M., Kobbelt, L.P., 2002. Efficient simplification of point-sampled surfaces. In: Proceedings of the conference on Visualization '02 (VIS ’02), IEEE, Boston, MA, 27 October – 1 November, pp. 163-170.

Pauly, M., Mitra, N.J., Wallner, J., Pottmann, H., Guibas, L.J., 2008. Discovering structural regularity in 3D geometry. ACM Transactions on Graphics (TOG) 27 (3), Article No. 43

Pu, S., Vosselman, G., 2006. Automatic extraction of building features from terrestrial laser scanning. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (5), 25-27.

Pu, S., Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing 64 (6), 575-584.

Rabbani, T., Van Den Heuvel, F., 2005. Efficient Hough transform for automatic detection of cylinders in point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3), 60-65.

Ramamoorthi, R., Arvo., J. Creating generative models from range images. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH '99), ACM, Los Angeles, CA, 8-13 August, pp. 195-204.

Schnabel, R., Wahl, R. and Klein, R., 2007, June. Efficient RANSAC for point- cloud shape detection. In Computer graphics forum (Vol. 26, No. 2, pp. 214-226). Blackwell Publishing Ltd.

Shi, B.Q., Liang, J., Liu, Q., 2011. Adaptive simplification of point cloud using k-means clustering. Computer-Aided Design 43 (8), 910-922.

Singh, M., Laefer**,** D., 2015. Recent Trends and Remaining Limitations in Urban Microclimate Models. Open Urban Studies and Demographics Journal 3, 1-12. http://dx.doi.org/10.2174/2352631901401010001.

Sotoodeh, S., 2006. Outlier detection in laser scanner point clouds. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 36 (5), 297-302.

Stiny, G., 1982. Spatial relations and grammars. Environment and Planning B: Planning and Design, 9 (1), 113-114.

Stiny, G., Mitchell, W.J., 1978. The Palladian Grammar. Environment and planning B: Planning and design, 5 (1), 5-18.

Tarsha-Kurdi, F., Landes, T., Grussenmeyer, P., 2007. Hough-transform and extended ransac algorithms for automatic detection of 3d building roof planes from LiDAR data. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 36 (Part 3), 407-412.

Truong-Hong, L., Laefer, D.F., Hinks, T., Carr, H., 2011. Flying voxel method with Delaunay triangulation criterion for façade/feature detection for computation. Journal of Computing in Civil Engineering 26 (6), 691-707.

Truong-Hong, L. and Laefer, D.F., 2015. Quantitative evaluation strategies for urban 3D model generation from remote sensing data. Computers & Graphics, 49, pp.82-91. doi.org/10.1016/j.cag.2015.03.001i.

Truong-Hong, L., Laefer, D.F., 2014. Octree-based, automatic building facade generation from LiDAR data. Computer-Aided Design 53, 46-61.

Truong-Hong, L., Laefer, D.F., Hinks, T., Carr, H., 2013. Combining an angle criterion with voxelization and the flying voxel method in reconstructing building models from LiDAR data. Computer-Aided Civil and Infrastructure Engineering 28 (2), 112-129. http://dx.doi.org/10.1111/j.1467-8667.2012.00761.x.

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

Vosselman, G., Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 34, 37–44.

Vincent, L. and Soille, P., 1991. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis & Machine Intelligence, (6), pp.583-598.

Wang, M., Tseng, Y.H., 2011. Incremental Segmentation of lidar point clouds with an octree- structured voxel space. The Photogrammetric Record 26 (133), 32-57.

Wani, M.A., Arabnia, H.R., 2003. Parallel edge-region-based segmentation algorithm targeted at reconfigurable multiring network. The Journal of Supercomputing 25 (1), 43-62.

Woo, H., Kang, E., Wang, S. and Lee, K.H., 2002. A new segmentation method for point cloud data. International Journal of Machine Tools and Manufacture, 42(2), pp.167-178.

Yang, M.Y. and Förstner, W., 2010, January. Plane detection in point cloud data. In Proceedings of the 2nd int conf on machine control guidance, Bonn (Vol. 1, pp. 95-104).

Zhang, H., Li, J., Cheng, M., Wang, C., 2016. Rapid inspection of pavement markings using mobile lidar point clouds. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 41 (Part B1), 259-264. http://dx.doi.org/10.5194/isprs-archives-XLI-B1-717-2016

Zolanvari, S.M.I., Laefer, D.F., 2016. Slicing Method for curved façade and window extraction from point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 119, 334-346. doi.org/10.1016/j.isprsjprs.2016.06.011