Slicing Method for Curved Façade and Window Extraction from Point Clouds

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

[a] Urban Modelling Group, School of Civil Engineering, University College Dublin, Ireland
[b] Earth Institute; University College Dublin, Ireland
[c] U3D Printing Hub, University College Dublin, Ireland

Abstract:

Laser scanning technology is a fast and reliable method to survey structures. However, the automatic conversion of such data into solid models for computation remains a major challenge, especially where non-rectilinear features are present. Since, openings and the overall dimensions of the buildings are the most critical elements in the computational models, this article introduces the Slicing Method as a new, computationally-efficient method for extracting overall façade and window boundary points for reconstructing a façade into a geometry compatible for computational modelling. After finding a principal plane, the technique slices a façade into limited portions, with each slice representing a unique, imaginary section passing through a building. This is done along a façade's principal axes to segregate window and door openings from structural portions of the load-bearing masonry walls. The method detects each opening area's boundaries, as well as the overall boundary of the façade, in part, by using a one-dimensional projection to accelerate processing. Slices were optimised as 14.3 slices per vertical metre of building and 25 slices per horizontal metre of building, irrespective of building configuration or complexity. The proposed procedure was validated by its application to three highly decorative, historic brick buildings. Accuracy in excess of 93% was achieved with no manual intervention on highly complex buildings and nearly 100% on simple ones. Furthermore, computational times were less than 3 seconds for data sets up to 2.6 million points, while similar existing approaches required more than 16 hours for such datasets.

[*] Corresponding Author: School of Civil Engineering; University College Dublin, Newstead, Belfield, Dublin 4, Ireland, Tel: +353 1 716 3226

Email addresses: Debra.Laefer@ucd.ie (Debra F. Laefer), Iman.Zolanvari@ucdconnect.ie (S. M. Iman Zolanvari), Website: www.zolanvari.com

Keywords: Laser scanning, 3D computational modelling, Window detection, Feature extraction, Point cloud data, LiDAR

1. Introduction

Laser scanning, also known as Light Detection and Ranging (LiDAR), has become a common and reliable tool in applications related to Civil Engineering like urban planning (Wu et al., 2015), disaster coordination (Vetrivel et al., 2015), vegetation management (Ferraz, et al., 2016), and auto-navigation (Vu et al., 2015). Automatically transforming these largely visual models into forms readable by finite element programmes remains a challenge, unless significant manual intervention is involved. Thus, their usefulness is hampered for use in the engineering community for district-level and city-scale topics such tunnel damage prediction (Laefer et al. 2010) and microclimate modelling (Singh and Laefer, 2015). In general, having a geometrically accurate, three-dimensional (3D) building model is an essential component for computational analysis. This is complicated to achieve economically, because the geometry for the vast majority of urban structures is undocumented, and the expense of collecting that data through traditional surveying methods is likely to be prohibitive. In such cases, remote-sensing technologies can offer a cost-effective alternative (Abayowa et al., 2015). Prominent amongst these is laser scanning, which furnishes a rapid means to collect the relevant data from three-dimensional (3D) objects, including Cartesian coordinates in the x-, y- and z-directions, intensity of the reflected laser beam, plus when there is an integrated digital camera, colour values [(red-green-blue (RGB)]. These 3D points are collectively referred to as a point cloud. The challenge then becomes the extraction of geometries of interest in a way that is accurate and technically meaningful and in a format that ultimately is compatible with computational modelling software.

Extracting relevant geometrical façade components from LiDAR data is challenging, in part due to data collection imperfections including occlusions and general noise. Currently, the majority of relevant approaches, while highly effective for visualization, have acute limitations with respect to generating input for computational models. Prominent limitations are as follows: (1) having applicability only to simple, rectilinear structures; (2) requiring a priori knowledge (e.g. a reference library); (3) necessitating significant manual user involvement; and/or (4) being computationally expensive. Presently these issues are impediments to using laser scanning to generate city-scale computational models. To begin to addresses these issues, this paper introduces the Slicing Method, a new means to rapidly and automatically detect a façade's overall geometry and its openings, even in the presence of non-rectilinear shapes and significant façade complexities.

2. Related works

Building façade extraction tends to involve two major components: segmentation and feature extraction. Segmentation is the process of segregating a group of points belonging to a single surface or region. Building segmentation often separates walls from roofs and different sides of a building from each other. In contrast, feature extraction involves identifying building features (e.g. doors and windows) from patches that resulted from segmentation. Segmentation is typically a precursor to feature extraction. Both are long-standing research topics in remote sensing as described below.

2.1. Segmentation

Segmentation has an essential role in the reconstruction of 3D models from laser scanning data. While a wide variety of strategies have been developed, the majority use some form of model fitting approach (e.g. Schnabel and Klein, 2007 and Awwad et al. 2010), such as the well-established Random Sample Consensus (RANSAC) technique introduced by Fischler and Bolles (1981). As an example, Boulaassal et al. (2009) integrated the use of RANSAC for wall segmentation of planar surfaces. RANSAC iteratively calculates parameters of a potential plane through a dataset with the goal of fitting a surface. Accuracy depends upon the number of iterations and the dataset size. Since walls have much higher point densities than openings or outliers, this method can extract walls, roofs, and doors that are located on a segmented plane. The method successfully extracts planar surfaces of building façades, as long as substantial protrusions or complex details are not present. Boulaassal et al. (2009) showed that this technique can also be used for boundary detection of segmented planes.

Another highly influential technique is that proposed by Wang et al. (2004), which is based on an octree indexing structure. In that approach, planar clustering was employed for 3D segmentation by splitting points into small planes (based on characteristics such as average height, average intensity, and shape orientation) and subsequently merging neighbouring planes that have similar normal vectors. Although, the method is relatively successful for rough segmentation, it cannot be used for complicated façades. Another octree-based approach was introduced by Vu et al. (2014), in which a region-growing algorithm was introduced to segment points into large, coarse surface patches by incrementally grouping adjacent voxels (3D cells) based on them having similar normal vectors and residual values (the quadratic mean of the orthogonal distances from the points to their best-fit plane). The procedure then considered the points inside the voxels, rather than outliers or unwanted points. To accelerate the process, only points belonging to voxels adjacent to incomplete segment boundaries (around each group of voxels) were considered as candidates for merging. That segmentation method was able to quickly extract different planes of a building's façades for both terrestrial and aerial laser scanning data. However, the scalability and robustness of the method have yet to be demonstrated.

Contemporaneous to that, Chen et al. (2014) proposed an automatic and threshold-free evaluation system that offers an object-based technique for roof extraction. The enhanced algorithm performs well, even in the presence of noisy data and roofs covered in vegetation. However, the method has to date not applied to façades, which are arguably often geometrically much more complicated than roofs. Also in 2014, Lari and Habib (2014) used principal component analysis to identify the points that belong to unique planar, linear or cylindrical surfaces. Next, they automatically selected the proper representation model of the detected elements. Then, the results were corrected by implementing cylindrical neighbourhoods that utilized estimation of local point density variations along their surfaces. Finally, the outputs were improved utilizing the characteristic attributes of the segmented elements. This work introduced a robust and new approach for the identification, parameterization, and segmentation of walls, roofs, and other groups of points having the same geometrical features. However, the segmented features are too rough and general for detecting openings and overall façade boundaries with great accuracy.

2.2. Feature detection

After successful segmentation, feature extraction is often pursued. This is a popular topic. The following represents only a sampling of recent contributions. For this, Wang et al. (2004) used a simple classification method based on planar attributes, which was limited to horizontal and vertical planes (e.g. roofs, ground surfaces and walls). Later Bendels et al. (2006) attempted to detect holes in point clouds by calculating the distance of each point to all of its neighbouring points by combining an angle and a shape criterion. The proposed approach was successful for holes that were either a part of a small object or that happened due to occlusions, reflectance, or transparency during data acquisition. The method's main drawback is its computational intensity due to its need to calculate all of the points' distances to each other and to find the normal that is determined as the eigenvector corresponding to the smallest eigenvalue of the weighted covariance matrix of the points. The method also needs some human intervention for such things as window frame removal.

Simultaneously, Pu and Vosselman (2006) proposed a fully automatic approach to extract building façade features from terrestrial LiDAR, which involved first segmenting a point cloud into planar patches, then used a set of rules (i.e. ontologies) based on common construction attributes (e.g. size, position, direction and topology). After this, each segment was compared by fitting polygons to these constraints for feature classification. The method works well, except for windows, because of insufficient data. Therefore, the authors developed a subsequent hole-based window extraction method (Pu and Vosselman 2007), which is effective, as long as the buildings are relatively simple and conform to the architectural styles from which the matching rules were derived.

Shortly thereafter, Becker and Haala (2007) introduced a method for solid model reconstruction, which involved integrating terrestrial images with laser scanning data. The resulting models were highly realistic returning even the crossbars of window frames. The technique, however, requires a sufficiently high density of laser scanning data to have a resolution comparable to that of a photograph. Otherwise, an unstable matching and orientation process is likely to occur.

Subsequently, Pu and Vosselman, (2009) identified façade features (e.g. doors and windows) for building façade reconstruction from terrestrial LiDAR by extending their ontological work (e.g. walls are usually large areas, which are mostly vertical planes with an intersection with a ground plane). The technique is better suited for more complicated façades than previous efforts. However, the method still relies on a priori information and user intervention to amend the final results. While visually successful, the technique is limited by (1) the devised rules having to cover all cases; (2) a heavy reliance on human intervention; (3) challenges with large protrusions; and (4) a dependency on a Delaunay Triangulation (DT) to generate a Triangular Irregular Network (TIN) for searching close neighbour points and detecting holes, which is highly time consuming for large datasets.

More recently, Truong-Hong and Laefer (2011) introduced the Façade Delaunay (FD) method in part based on DT to extract boundary points of a façade and its openings. In that method, a data point was considered a boundary point, if the point was an end point of a triangle's side having a length larger than a predefined length threshold. The study reported that a threshold by 0.15m was adequate to detect sufficient boundary points for further processing, which reconstructed the complete 2D building façade. However, since the process needs to examine all triangles in the mesh, the process is highly time-consuming with large data sets. In addition, the method may require a user-defined length threshold depending upon the data density and building form. Furthermore, Truong-Hong and Laefer (2011) also introduced a condition to eliminate boundary points of unrealistic holes. A hole was considered unrealistic, if its characteristics (height, length and a ratio of height over length) differed from those typically found in building construction based on values proposed by Becker and Haala (2007) and Pu and Vosselman (2007).

Then Truong-Hong et al. (2013) used an angle criterion to detect boundary points with a kd-tree to accelerate searching k-nearest neighbour (kNN) points of a candidate boundary point. The method also implemented a boundary coherence algorithm (Bendels et al. 2006) to eliminate incorrect boundary points. The algorithm was highly robust and accurate in extracting boundary points but requires a user-defined number of kNNs. Similar to the drawbacks of the work by Truong-Hong and Laefer (2012), the searching boundary point process is cumbersome, as it must check all points in the data set.

To reduce processing times, Truong-Hong and Laefer (2014) then introduced the Façade Voxel (FV) technique. In that method, only data points around a façade's boundaries and its openings were considered as candidate boundary points. The candidate points were extracted from full voxels around a façade's boundaries and openings. The probability of the angle at the candidate boundary point and of a triangle's perimeter, where the triangle was created from the candidate boundary point and its neighbour points and the centre point, was considered as the condition for boundary point determination. Notably, an initial voxelization bounding an input data set was subdivided into 4 smaller voxels, until the voxel size was no larger than a predefined threshold (0.2m). Then the voxel was classified as either full or empty; the centre point can be an arbitrary point within the convexhull of the candidate points. The method significantly improved processing times of previous work by the authors, while still providing sufficient data points for building reconstruction.

3. **Proposed method**

To overcome some of the above-listed deficits (i.e. handling non-rectilinear elements and reducing computational time) for masonry structures, a new method is proposed to extract façade boundaries and any openings within (e.g. window and doors). This class of buildings was selected as they represent the bulk of the building stock for which there are no as-built drawings. The method includes 3 main steps. First, each façade is roughly segmented using a planar RANSAC algorithm. Points outside of the plane are then removed; non-planar points often appear due to noise, vegetation, and interior spaces detected through windows. This segmentation may include curved and protruded features. Second, the segmented façade is sliced horizontally or vertically. Each slice passes through a portion of a wall and all of its affiliated features (e.g. decorative protrusions, windows, frames). Since glass cannot reflect the laser beams back to the scanner, if a slice encounters a window, a gap should appear. A gap may also appear if data are missing due to an occlusion; in this paper occlusions are not considered. In the proposed approach each gap causes a slice to be divided into two clusters. In general, the number of clusters in each slice is equal to the number of gaps plus one. As windows frames would cause small clusters compared to the rest of the slice, a threshold could be established for their removal. Ultimately, such a step could be extremely helpful in the removal of such non-structural elements as window crossbars. However, this has not been implemented herein. Generally, window with curtains or shades and recessed doors, are not problematic as they are on a recessed plane and not considered due to the initial planar segmentation. If the number of points on the recessed curtains is too plentiful (in comparison to neighbouring points), then the fitted RANSAC plane may unintentionally shift a few centimetres backwards behind the real façade. However, given the rest of the procedure, the impact on opening detection is likely to be negligible and could be corrected by changing the thresholding for the RANSAC step, if the problem appeared. To accelerate computation, in this step the y-axis is assigned as the new, local, vertical axis (from the ground skyward). The x-axis is along the façade (either to the left or right), and the z-axis points either inward or outward from the façade. In the third step, extreme points of each slice are extracted to create facade and opening area boundaries (Fig. 1). The entire procedure is then repeated, either horizontally or vertically depending upon the initial decision.

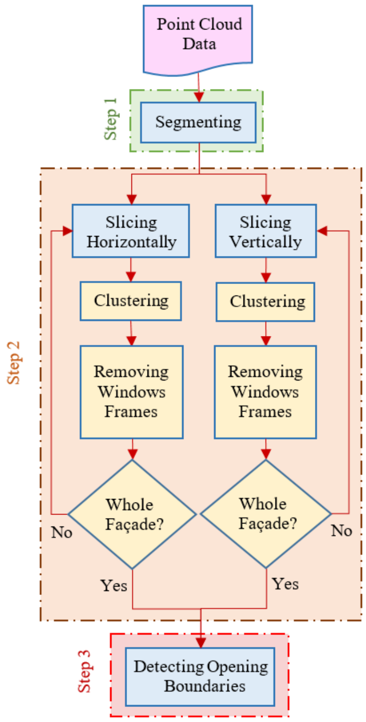


Fig. 1. Slicing Method Process

The method can be formulated by a set of points P = {p_1, p_2, p_3…, p_i . . . p_M} ⊂ $\mathbb{R}^3$ in the point cloud, which in the first step is segmented with a RANSAC algorithm from the façade into N planar or non-planar segmented parts {PL_1, PL_2 … , PL_i … PL_N}. Each PL_i corresponds to a set of 3D co-planar points. Afterwards, PL_i is divided into M number of slices S={S_1, S_2, S_3 …, S_j … , S_Q}.

$$S_j\,(P)\colon\; P \longrightarrow \{\, P_i\,(x_i, y_i, z_i) \in P \mid \text{if } j \neq h \Leftrightarrow S_j(P) \cap S_h(P) = \emptyset \,\} \qquad (1)$$

Then, each slice is projected onto a locally defined X-axis. This means that the y-value of every point of each slice would be a constant number equal to Eq. (2):

$$\forall P_i\,(x_i, y_i, z_i) \in S_j\,;\, y_i = c \qquad (2)$$

Following that, the cumulative dataset of the projected points along each line is considered. The distance of each point in the line is calculated, with respect to its two closest neighbours (1 prior and 1 subsequent) using Eqs. (3) and (4):

$$di = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2} \xrightarrow{\because\,(y_i = y_{i-1}\,\&\, z_i = z_{i-1})} \qquad (3)$$

After projecting each slice, the *y* and *z* values of those points that belong to the same slice in Eq. (3) are equal. Therefore, the required distance (*dj*) can be calculated in one-dimension in Eq.4. Also, Pk are points that are in slice Sj:

$$dj = \sqrt{(x_i - x_{i-1})^2} \mid i \geq 2, 3 \ldots, K \;\&\; P_k \in S_j \qquad (4)$$

Gaps are detected by establishing which points have affiliated distances that are greater than twice the median distance of points along that line (Eq. (5)). Each gap separates the slice into two different clusters as two groups (1 prior and 1 subsequent) {G1, G2, G3, …, Gv} (Eq. 6 and 7).

$$\xrightarrow{after\ arranging\ all\ distances\ (di)\ in\ order} Median = \begin{cases} \left(\frac{n+1}{2}\right) th\ dj, & n \text{ odd} \\ \frac{\left[\left(\frac{n}{2}\right) th\ dj + \left(\frac{n}{2}+1\right) th\ dj\right]}{2}, & n \text{ even} \end{cases} \quad (5)$$

$$if\ dj > 2 \times Median(\ d_1, d_2 \ldots, d_k\ )\ \Rightarrow Sj = G_1 \cup G_2 \quad (6)$$

$$\text{and for general case: } Sj = G_1 \cup G_2 \cup \ldots \cup G_V \quad (7)$$

Arguably, the proposed approach is applicable to all point clouds of sufficient density, irrespective of the data source (airborne or terrestrial laser scanning or imagery).

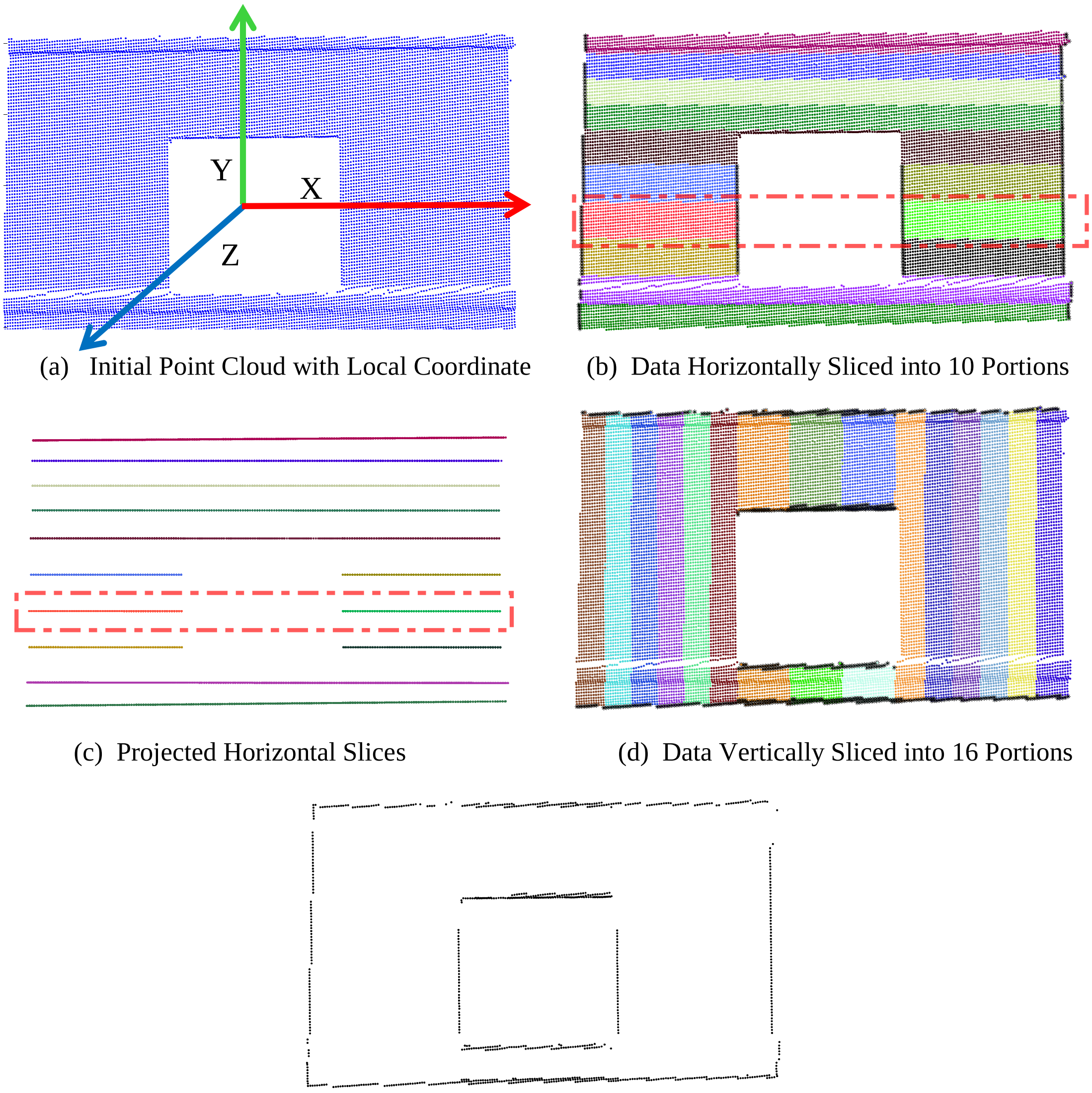
3.1. *Step-by-step implementation*

In this section, a simple rectangular façade with one opening (a window) is selected to illustrate the proposed method in a step-by-step manner. The procedure commences by segmenting all candidate points – defined as those co-located on a single surface. For most flat façades, the planar form of RANSAC is used. RANSAC's non-planar form is used for curved and heavily reticulated façades. RANSAC passes an imaginary surface through the façade. As long as RANSAC can detect the surface, the algorithm can operate correctly. This means that the façade need not be rectilinear along its boundaries nor exclusively planar. For planar RANSAC, the detected plane is considered as the façade, and all points outside a buffer zone of 20 cm are removed; the threshold can be as large as the average façade protrusion or can vary in accordance to the level of noise in the data. Since, this plane is infinite on all sides, its orientation around the z-axis is unimportant. The only crucial parameter is its offset position with regard to the origin of the z-axis.

To simplify calculations, the coordinate system is now reassigned as a local one using the lowest y-point as the origin and orienting the x-direction along the façade (Fig. 2a). Next, the façade is sliced into equal layers to distinguish solid areas (e.g. bricks, stone, or mortar) from the openings (e.g. a window) and recessed portions (e.g. a door). Theoretically, the thickness of each slice could vary. However, for simplicity sake, a uniform slice thickness is selected. More slices will capture geometrically smaller features and improve accuracy of the boundary points but at the expense of computational speed. Determining the optimum number of slices will be discussed later in the section 5.

Figure 2b shows the horizontal slicing results. For better visualisation, each cluster is portrayed with a randomly assigned colour. After slicing, each slice is projected along its lower bound x-axis to reduce computational time; data projection greatly accelerates the distance calculation step, as the problem is reduced by one dimension [Eqs. (3) and (4)]. If a distance is bigger than twice the median distance [Eqs. (5) and (6)], a gap has been detected. Since gaps are a function of openings and occlusions, their positions and sizes cannot be predetermined.

Finally, after detecting gaps and clustering all projected slices, the points are transferred back to their original positions, but each point retains its assigned cluster label. Figure 2b shows the data of each slice projected as a single horizontal line segment line positioned along the slice's lowermost boundary. Split lines are shown in unique colours according to their original slice colour. Prior to repeating the procedure with vertical slices, the final step involves identifying boundary points. This is done by picking the outermost points (minimum and maximum) of each projected cluster. In this paper, the last 1 cm of the projected line was used as a threshold for boundary point identification; threshold selection is discussed further in Section 5. Once the vertical border points (black dots in Fig. 2d) and horizontal border points (black dots in Fig. 2b) are identified, the extreme points of each cluster are assigned as border points of a specific opening or for the entire structure Fig. 2e.



(a) Initial Point Cloud with Local Coordinate

(b) Data Horizontally Sliced into 10 Portions

(c) Projected Horizontal Slices

(d) Data Vertically Sliced into 16 Portions

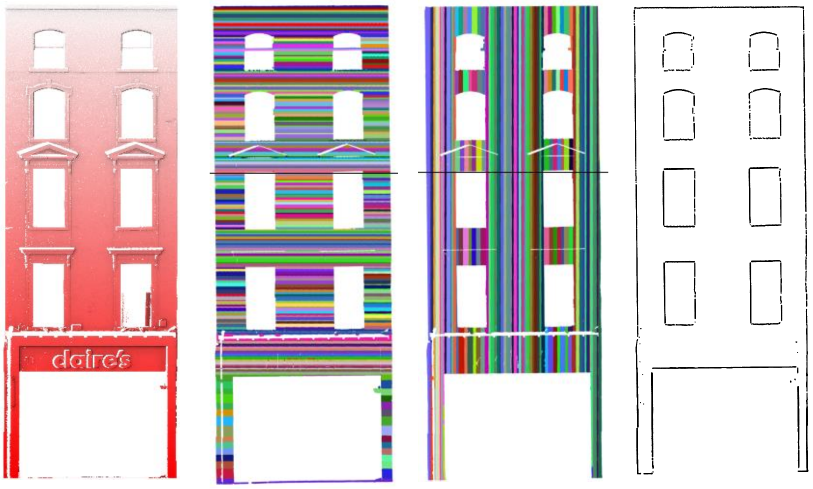
(e) Final Extracted Boundary Points

Fig. 2. Step-By-Step Illustration of the Boundary Detection Around a Single Window +Slice 4 is Marked within the Dashed Box in Fig. 2b show the input into step in Fig. 2c

4. Algorithm validation

The approach was applied to three buildings where aerial and terrestrial data were merged. The aerial data came from a RIEGL Q680i with a final projected ground density of 336 pts/m^2 (ground projection) where multiple aerial strips were merged. This was then combined with data from a terrestrial scan obtained with a Leica P20 unit. The merged dataset was used to produce a high-density data set with minimum façade occlusions. The final average density of each façade was calculated by dividing the projected area of each façade (m^2) by all collected data points on that particular façade (pts). The algorithms were implemented in MATLAB© 2015, and the data were processed using an Intel® Core™ i7-4770 CPU @ 3.4 GHz system, with an nVIDIA Riva TNT graphics board with 12.0 GB RAM under a 64-bit Windows10 configuration.

The three buildings were from Dublin, Ireland’s city centre. Case 1 is a three-story masonry building of 42 m^2 (Fig. 3). The building has large protrusions around the openings and curved window tops on the top two floors. Case 2 (Fig. 4) is of a similar type and scale (three-story masonry building of 108 m^2) but with more complicated façade features and no vertical window alignment (thus preventing the application of many model or procedurally driven techniques). Case 3 is a much larger structure (877 m^2) with a highly complicated border shape, rounded doors, and prominent window frames (Fig. 5). This structure is one of Dublin’s oldest built in around 1700 A.D. (Christine, 2006).



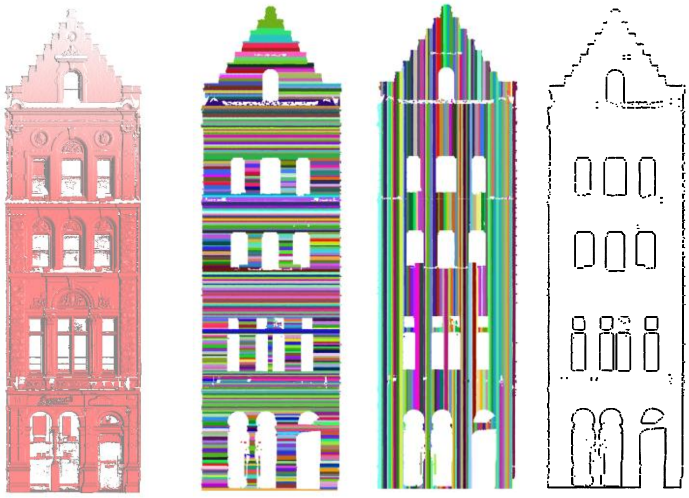
(a) Original Point Cloud

(b) Horizontal Slices

(c) Vertical Slices

(d) Extracted Boundaries

Fig. 3. Case 1, 13 Westmoreland Street



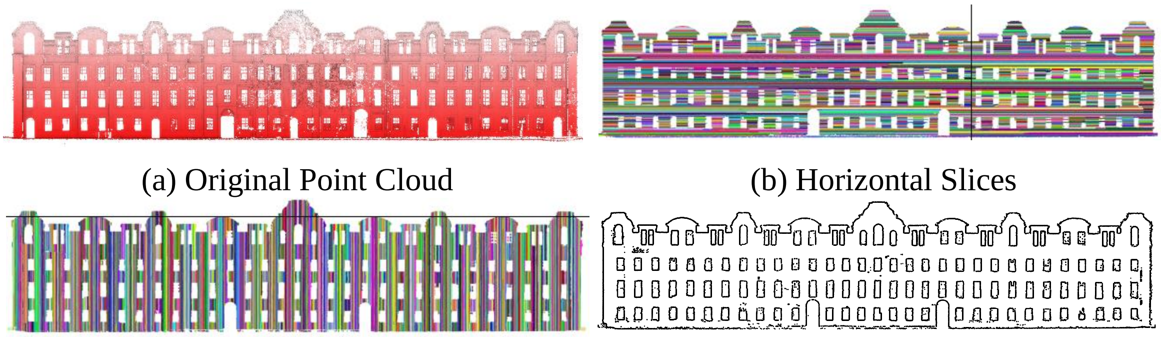
(a) Original Point Cloud

(b) Horizontal Slices

(c) Vertical Slices

(d) Extracted Boundaries

Fig. 4. Case 2, 14 Westmoreland Street


(a) Original Point Cloud

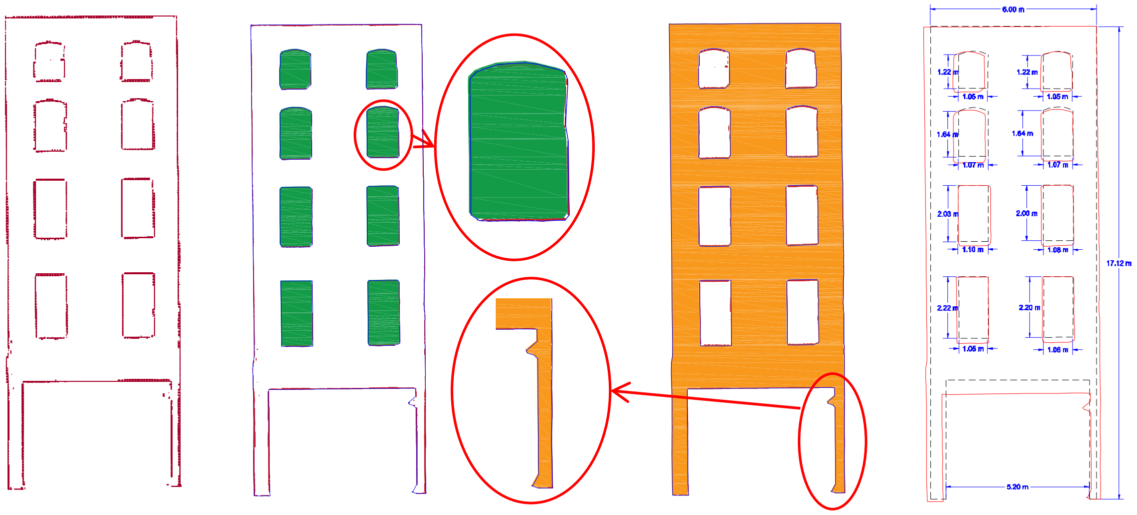
(b) Horizontal Slices

(c) Vertical Slices

(d) Extracted Boundaries

Fig. 5. Case 3, Rubrics Building

To evaluate the accuracy of the results, all of the detected boundary points were segmented using the label connected components method (LCC) proposed by Haralock and Shapiro (1991). LCC is a fast and robust method that segments point groups that are closer to each other than its other neighbours. The LCC method is applied only to the boundary points. This will separate each opening from its neighbour. For each segmented opening (windows and doors), points were substituted with circumscribed polygons that were generated with convex hulls, as shown in Fig. 6b. Overall façade boundary points were connected to each other using a concave hull algorithm, as shown in Fig. 6c. Next, all circumscribed polygons (of opening and overall boundaries) were imported into AutoCAD 2016 and overlapped onto independently derived, measured drawings (referred to herein as the reference model). The actual obtained areas were derived from the overall boundaries that are created from the concave hull minus the opening areas (orange areas in Fig. 6 (c)). Comparison of the results with measured drawings is shown in more details for a set of randomly selected windows (Fig. 7 and Table 1). Finally, the accuracy was defined by calculating the area of the substituted polygons divided by area shown on the reference model (Table 2).



(a) Detected Boundaries Points

(b) Circumscribed Polygons Inserted Using Convex Hull

(c) Circumscribed Polygon Applied Across the Façade Using Concave Hull Minus Openings

(d) Measured Drawing

Fig. 6. Evaluation of Case 1

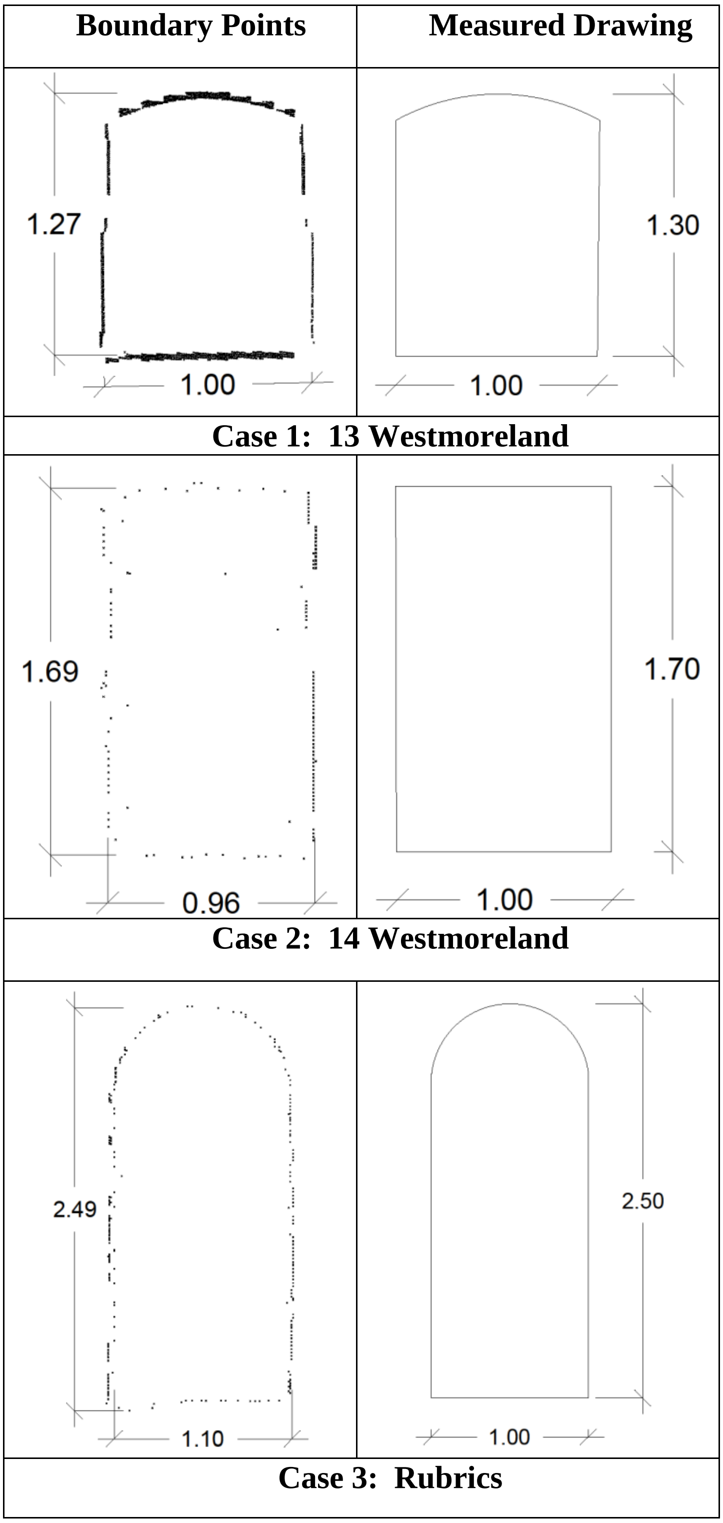


Fig. 7. Sample Boundary Points versus Reference Models

Table 1. Comparison of Extracted and Measured Areas from Reference Models

	Extracted Boundary Points			Reference Model			
Cases Name	Height x Width (m)	Area$_1$ (m^2)	Overall Height& Width(m)	Height x Width (m)	Area$_2$ (m^2)	Overall Height& Width(m)	Accuracy |(Area$_1$/Area$_2$)|
Case 1	1.27x1.00	2.62	16.88x6.1	1.30x1.00	2.40	17x6.2	91.60%
Case 2	1.69x0.69	1.60	22.19x5.69	1.70x1.00	1.70	22.5x6.2	94.11%
Case 3	2.49x1.10	1.19	15.91x70.38	2.5x1.00	1.26	16x70	94.44%

Table 2. Case Information and Slicing Method Outcomes

Data	Façade Dimensions [Height (m) x Length (m)]	Area (m2)	Number of Available Points	Average Point Density (pts / m^2)	Number of Horizontal Slices	Number of Vertical Slices	Time of Boundary Detection (Sec)	Accuracy (%)
Case 1	17x6.2	105.4	1,543,312	14,698	250	100	0.613	96.8
Case 2	22.5x6.2	139.5	1,496,958	10,730	250	80	0.879	95.3
Case 3	16x70	1,120	2,631,757	2,349	200	500	2.530	93.1

The required calculation time reported in Table 2 is only the summation of the vertical and horizontal slicing procedure plus the detection of the boundary points. The time needed for importing and reading the data into the algorithm was excluded. Since, the height of the three buildings are almost the same, the optimum number of horizontal slices for each case was highly similar (200 or 250). Because the buildings differed significantly in length, the number of vertical slices varied significantly: from 80 to 500. Accuracy ranged from nearly 93% to almost 96%. Processing times were less than 1 second for the 2 smaller buildings and only 2.5 seconds for Case 3 which was nearly 10 times larger and required a much greater number of slices and data points (almost 1.2 million more points more than the first 2 cases). While the accuracy was very high, the technique is not perfect. However, part of the loss of accuracy is due to imperfections in the initial point cloud, as will be discussed in the next section.

5. Discussion

This section will discuss three major topics related to the ultimate success of the Slicing Method procedure. The first is the optimum number of slices with respect to the accuracy of the results and the time of calculation. The second is the required data density, and the third is the data coverage.

5.1. *Optimum number of slices*

The number of slices can influence both the calculation time and the accuracy of the detected boundaries. This parameter can be illustrated by looking at a simple window (Fig. 9). This window is one of many that appears in Fig. 2. This subset of the data has 15,695 points with an average density of 2,308 pts/m^2. When a 10 slice division was compared to a 50 slice one, there was only a negligible 0.021 sec difference in processing times (0.040 sec vs 0.061 sec) for the slicing and boundary detection portion of the algorithm.

The accuracy was strongly influenced by the number of slices. The number of slices versus the output accuracy in each of the three cases is shown in Fig. 10. The optimum number of slices was taken as the maximum achievable accuracy, which was in excess of 92%. To obtain these optimal values, an initial number of vertical slices was assumed. The number of horizontal slices was then optimised. Based on those outputs, the number of vertical slices was subsequently re-optimised. The process was repeated until convergence occurred. According to Fig. 9, beyond the optimum number of slices (both horizontal and vertical), there was a noticeable loss of accuracy with the addition of further slices. This phenomenon happened because of three main reasons:

a. Since the thin slices pass through obstructions on the windows (i.e. cross bars or reflective stickers), the algorithm cannot detect where a gap starts and ends as the slice appears continuous.

b. Thin slices cause very small clusters that may be removed by the SM algorithm as outliers.

c. The small number of points will give a small value of median that is almost equal to the mean of the individual distances (due to the discrete nature of the data) and cannot show where the big gaps are located. This effect is shown in Fig. 8.

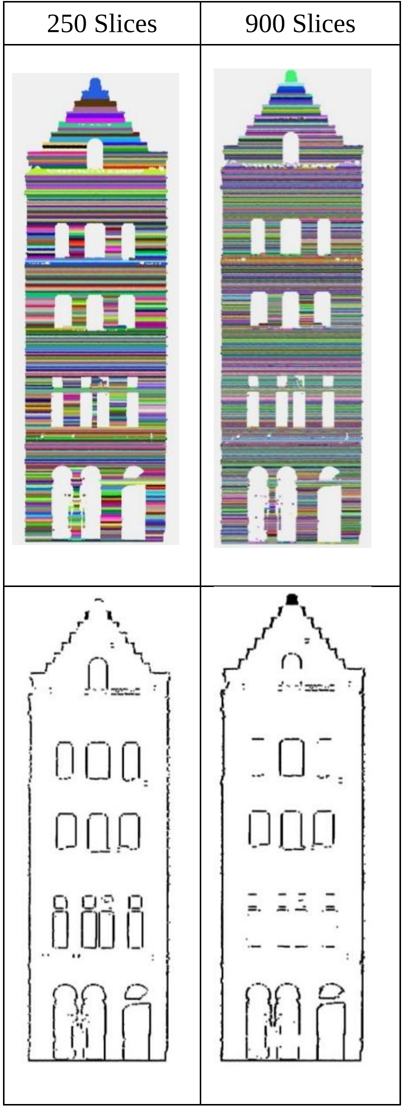
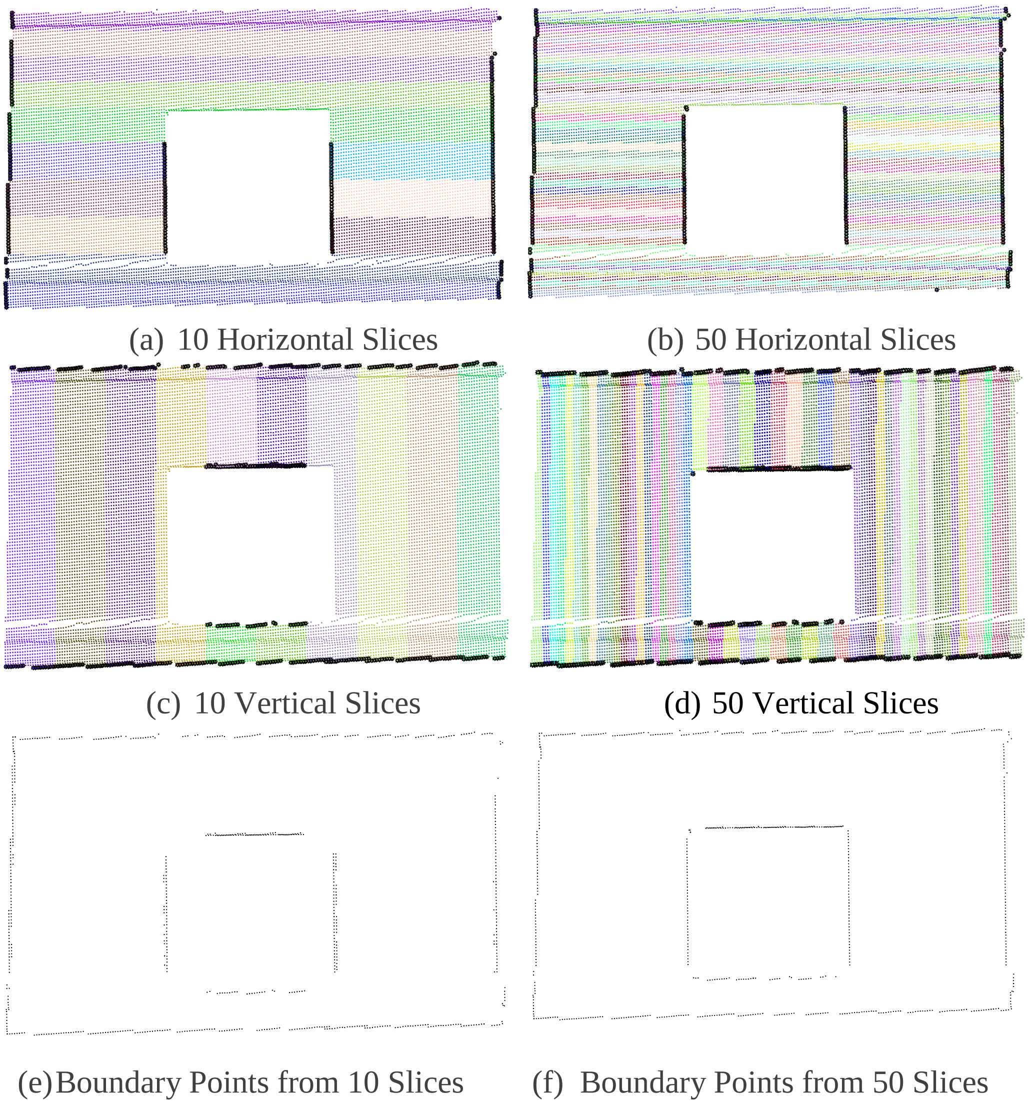


Fig. 8. Comparison of results of 250 and 900 Slices

For horizontal slices, the optimum number was in the range of 200 to 250 slices. Beyond that point there was a loss of accuracy, despite an allocation of additional computational resources in the form of processing additional slices. The similarity in the optimal number of horizontal slices across the 3 cases was due to the fact that all of the buildings were about the same height (17 m - 22.5 m). When the number of slices was considered with respect to the building heights, this value was calculated from Eq. 8. The result was 7cm for the horizontal slices was 4cm for the vertical ones.


(a) 10 Horizontal Slices

(b) 50 Horizontal Slices

(c) 10 Vertical Slices

(d) 50 Vertical Slices

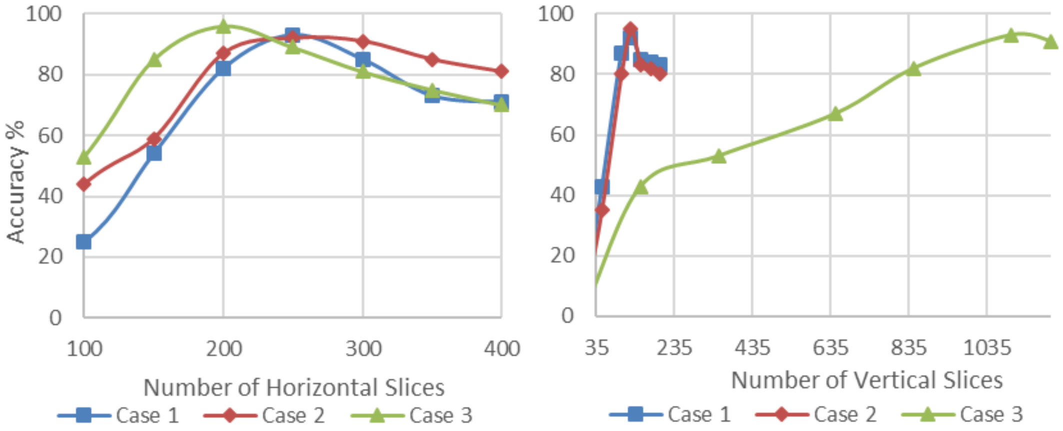
(e)Boundary Points from 10 Slices

(f) Boundary Points from 50 Slices

Fig. 9. Comparative Application of Slicing Method on a Simple Window with 10 and 50 Slices

In contrast to the high similarity in the number of horizontal slices, the optimal number of vertical slices varied greatly – from 80 to 1,100, as the buildings differed significantly in width (from 6.2 to 70 m), but when considered as a function of building width, the results from the 3 cases were highly similar around 4 cm or 25 slices per horizontal meter of building (Table 2). In addition, the point cloud density does not influence the optimum slice thickness, as it is the function of overall dimension of façade. Density only affects the median of the points' distance. Therefore, if the openings (gaps) are bigger than the median, the SM may fail to detect the opening, but the number of slices does not change.

Figure 11 shows the number of slices versus time of calculation for the slicing and boundary point identification portion of the algorithm for each case. For the horizontal slices, the number of slices in Cases 1 and 2 were relatively independent of time. In contrast, Case 3 had a clear time-based dependency. However, as the actual expenditure in computation was only a further 0.4 sec between the most extreme cases (10 slices versus 600), the dependency can be considered as negligible. Therefore, only output accuracy should be a consideration for selecting the number of slices.

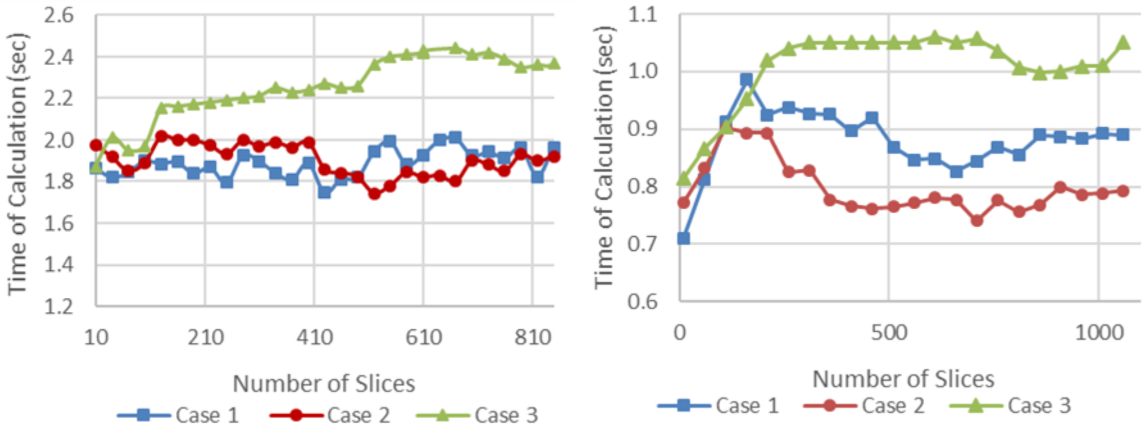


(a) Horizontal Slices

(b) Vertical Slices

Fig. 10. Accuracy of Detecting an Opening with Respect to Number Slices

$$Optimum\ thickness\ of\ slices\ = \frac{Width\ of\ the\ facade}{Optimum\ number\ of\ all\ slices} = \begin{cases} (1700\ cm)/(250\ ) \ \ \cong 7\ cm, & for\ Horizontal\ Slices \\ (620\ cm)/(150\ ) \ \ \cong 4\ cm, & for\ Vertical\ Slices \end{cases} \tag{8}$$



(a) Horizontal Slices (b) Vertical Slices

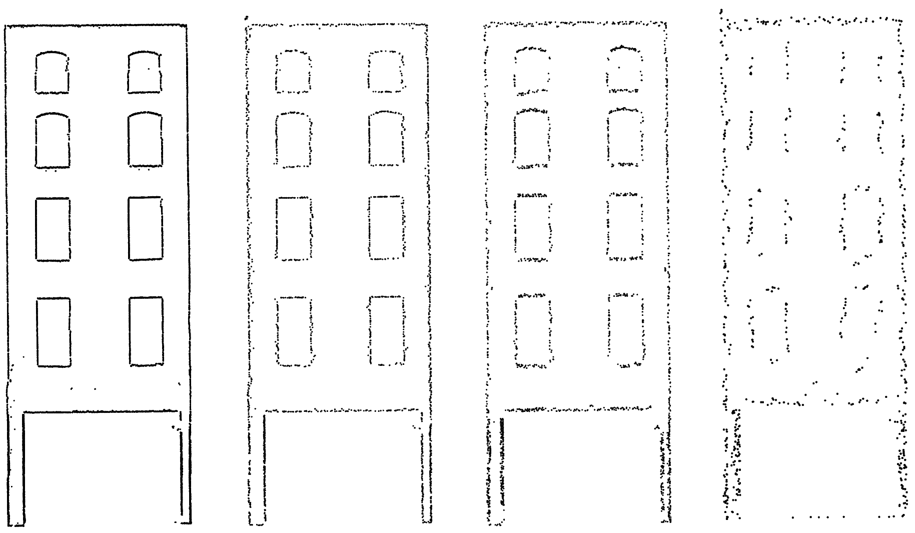
Fig. 11. Accuracy of Detecting an Opening with Respect to Number Slices

5.2. *Data density*

The second important input parameter for the Slicing Method is data density of the point cloud. To establish a minimum viable data density, a sensitivity study was undertaken where the data were randomly degraded, and the outputs were compared (Fig. 12). While, this does not show a required minimum density, boundary detection was still largely achievable (with accuracy of 84.2%) at 175 pts/m^2. At 31 pts/m^2 an accuracy of 43% was achieved. With decreasing densities degradation first appears around the windows and ultimately around the façade boundaries. The SM may fail with very low density (lower than 31 pts/m^2). An absolute minimum cannot be stated, because it depends on require accuracy, shape of openings (rectangular of irregular curved), size of opening and windows cross bars inside of openings.

To benchmark the Slicing Method directly against other window and façade detection methods, the SM method was tested against 3 similar methods identified in the literature: the Façade Delaunay (FD) [Truong-Hong, et al. 2011], the Façade Angle (FA) [Truong-Hong, et al. 2013], and the Façade Voxel (FV) [Truong-Hong and Laefer, 2014]. The datasets used by the authors of those methods were significantly simpler, as the building boundaries and openings were strictly rectilinear (Fig. 13). The outputs of those three methods were reported based on running on a Precision Workstation T5400 Intel(R) Pentium (R) Xeon (8CPU) 2GHz with 8190Mb RAM. While not identical to the authors' machine. the processing capacities are not wholly dissimilar. The comparative results are shown in Table 3. In each case, the SM approach was at a minimum four orders of magnitude faster, while achieving at least equivalent accuracy. The vast improvement in speed is because each of the three comparative methods consider each data point with respect to all of its neighbours in 2D, while the SM procedure considers only a point's two closest neighbours in one dimension.

The lowest density used with the FA, FD and FV algorithms by the original authors was 175 pts/m2 and generated accuracies of 92% to 98% when applied to Cases A, B and C in figure 13. In contrast, at that density the Slicing Method generated accuracies of 97% to 99%. The SM method was found to be able to extract boundary points at a 43.4% accuracy level with densities as little as 31 pts/m^2 for even more complicated planar masonry buildings. In summary, since the required quality of the outcomes will dictate the needed input densities, there is no absolute minimum density that can be stated as a general rule. However, a density of 130 pts/m^2 is recommended to achieve at least a 90% accuracy for the kinds of building presented herein.



(a) 2500 pts/m^2 Accuracy 96.2% Time 0.25 sec

(b) 400 pts/m^2 Accuracy 91.0% Time 0.09 sec

(c) 175 pts/m^2 Accuracy 84.2% Time 0.08 sec

(d) 31 pts/m^2 Accuracy 43.4% Time 0.08 sec

Fig. 12. Results from Various Average Density

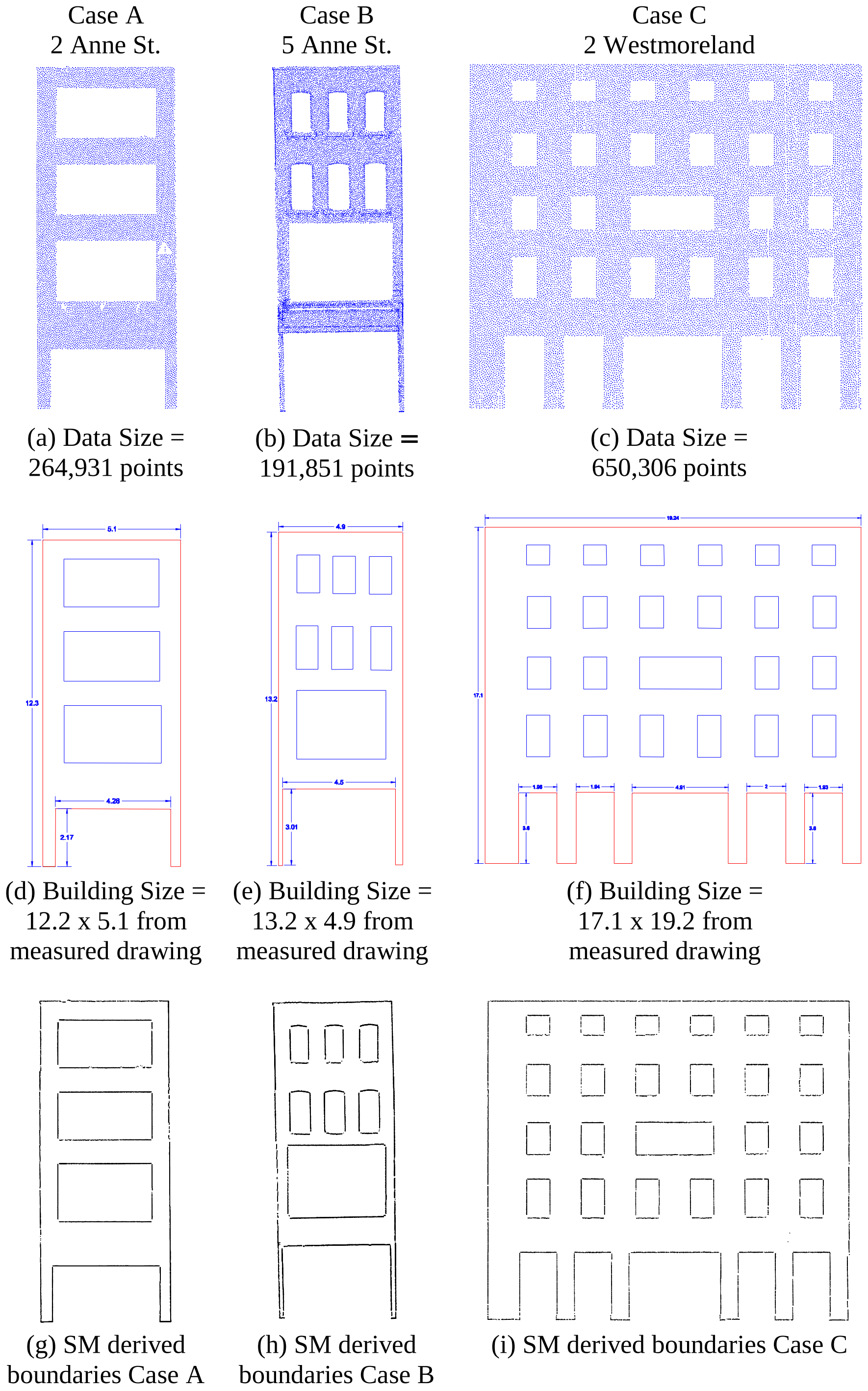
Table 3. Comparative Data Outputs from Façade Angle (FA), Façade Delaunay (FD), Facade Voxel (FV) and the Slicing Method (SM)

Data	FA Accuracy	FD Accuracy	FV Accuracy	SM Accuracy	FA Time	FD Time	FV Time	SM Time
2 Anne St.	100%	99.6%	99.6%	99.5%	2.4 hr	2.5 hr	8 min	0.38 sec
5 Anne St.	94.4%	94.4%	96.7%	98.1%	50 min	1 hr	10 min	0.50 sec
2 Westmoreland	97.1%	98.5%	97%	97.8%	16 hr	16.6 hr	42 min	0.61 sec

5.3. *Data coverage*

A third crucial characteristic of data acquisition that can affect the quality of the results is data coverage. This includes protrusion-based shadows (unexpected holes) on the façade. To show the impact of this phenomenon on the Slicing Method, close ups are provided for Case 2 in Fig. 14. The occlusions at the uppermost and first level due to the angle of data were repaired and populated with synthetic points in Fig. 14b. The accuracy went from 94.8% to 98.1%.

The FA, FD and FV methods were applied previously to three cases to which the Slicing Method has now been applied to enable direct comparison (Fig 13). When comparing the Slicing Method to the current state of the art for data driven techniques, three things become apparent. The first relates to the extended capabilities of this technique over other published techniques with respect to the tremendous speed of the procedure coupled with the capacity to detect curved window and façade boundaries. Lastly no manual preparation, a priori information, or user-selected thresholds are needed.



Case A
2 Anne St.

Case B
5 Anne St.

Case C
2 Westmoreland

(a) Data Size = 264,931 points

(b) Data Size = 191,851 points

(c) Data Size = 650,306 points

(d) Building Size = 12.2 x 5.1 from measured drawing

(e) Building Size = 13.2 x 4.9 from measured drawing

(f) Building Size = 17.1 x 19.2 from measured drawing

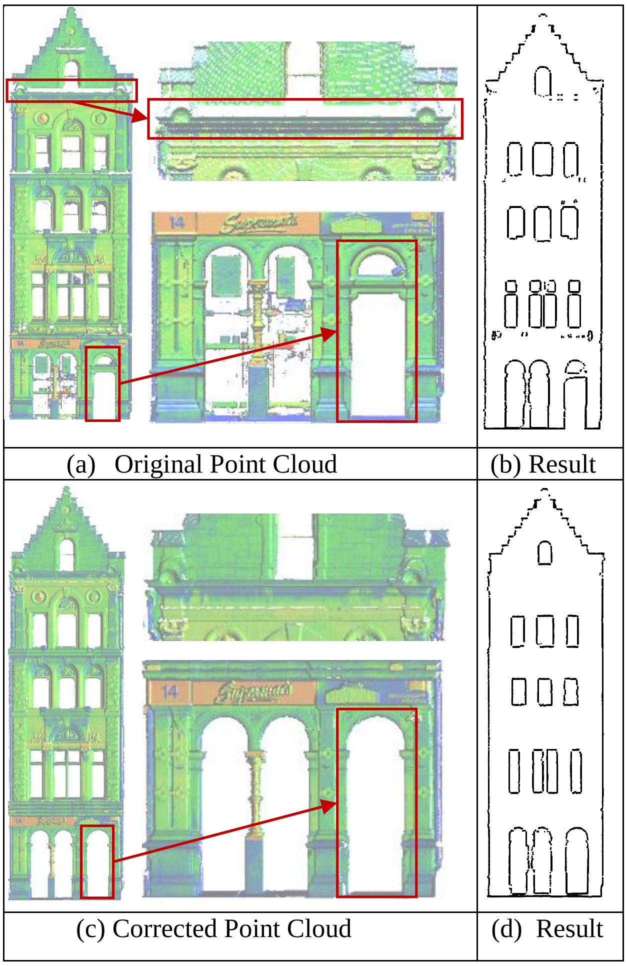
(g) SM derived boundaries Case A

(h) SM derived boundaries Case B

(i) SM derived boundaries Case C

Fig. 13. Comparative Data Output

This is not say that the method is without limitations. First, occlusions interfere with proper boundary detection, and if an occlusion is as large as the smallest real opening (i.e. window), problems are likely to emerge (an issue previously identified by Truong-Hong and Laefer, 2013). Second, if cross bars on the windows are sufficiently thick to be co-located on the façade plane, the algorithm may not be able to distinguish them from the facade, instead of as unwanted points inside of openings. Most likely, this latter point can be addressed with small modifications in the feature detection portion of the algorithm. Finally, as demonstrated in Fig. 14a, even with very high density data, there are still further opportunities for boundary detection refinement through the introduction of a further post-processing step, which could greatly facilitate the use of much lower density datasets. This is important, because although the speed of the technique is not negatively impacted by very dense datasets, such datasets may not always be available in mobile and/or aerial laser scanning. Furthermore, a client may not want to expend the resources for acquiring such density either with respect to the acquisition time or in regard to the burden of data storage and management.


(a) Original Point Cloud

(b) Result

(c) Corrected Point Cloud

(d) Result

Fig. 14. Original Point Cloud versus Repaired Data Set Outcomes for Case 2

6. CONCLUSION

This paper introduced the Slicing Method that was developed to quickly detect free-form openings and overall boundaries from structures with out-of-plane protrusions and complex façades. The method involves three main steps. First, each façade is roughly segmented using a RANSAC-based algorithm. Second, the segmented façade is sliced horizontally or vertically, with each slice passing through a portion of a wall and all of its affiliated features. If a slice encounters a window other discontinuity, the slice is divided into two clusters. Third, the data from each slice is projected onto a line from which the extreme points are harvested to create facade and window boundaries. The Slicing Method represents a major breakthrough in the following: (1) identifying the actual geometries for non-rectilinear openings; (2) accurately extracting complicated facade boundaries; and (3) minimizing the computational resources needed. This was achieved with no reliance on a priori information, user-selected threshold, or manual user involvement. When benchmarked against three highly ornate masonry structures, the method consistently achieved at least 93% accuracy in only 2.5 seconds with datasets of 2.6 million points. When the Slicing Method was then tested against three other recent methods using only rectilinear structures, the computation time was at least four orders of magnitude faster.

ACKNOWLEDGMENTS

This work was sponsored by European Research Council grant ERC-2012-StG_20111012" RETURN – Rethinking Tunnelling in Urban Neighbourhoods" Project 30786.

REFERENCES

[1] Wang, M. and Tseng, Y.H., 2004. Lidar data segmentation and classification based on octree structure. *parameters*, *1*, p.5.

[2] Pu, S. and Vosselman, G., 2007. Extracting windows from terrestrial laser scanning. *Intl Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*, pp.12-14.

[3] Boulaassal, H., Landes, T. and 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), pp.1-20.

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

[5] Fischler, M.A. and 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), pp.381-395.

[6] Bendels, G.H., Schnabel, R. and Klein, R., 2006. Detecting holes in point set surfaces.

[7] 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.

[8] Vosselman, G. and Maas, H.G. eds., 2010. *Airborne and terrestrial laser scanning*. Whittles Publishing.

[9] Biosca, J.M. and 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), pp.84-98.

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

[11] Becker, S. and Haala, N., 2007, September. Combined feature extraction for façade reconstruction. In *Proceedings of the ISPRS Workshop Laser Scanning* (pp. 241-247).

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

[13] 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.

[14] 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.

[15] Haralock, R.M. and Shapiro, L.G., 1991. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., pp. 28-48.

[16] Abayowa, B.O., Yilmaz, A. and Hardie, R.C., 2015. Automatic registration of optical aerial imagery to a LiDAR point cloud for generation of city models.*ISPRS Journal of Photogrammetry and Remote Sensing*, *106*, pp.68-81.

[17] Ferraz, A., Mallet, C. and Chehata, N., 2016. Large-scale road detection in forested mountainous areas using airborne topographic lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *112*, pp.23-36.

[18] Wu, B., Tang, S., Zhu, Q., Tong, K.Y., Hu, H. and Li, G., 2015. Geometric integration of high-resolution satellite imagery and airborne LiDAR data for improved geopositioning accuracy in metropolitan areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, *109*, pp.139-151.

[19] Vetrivel, A., Gerke, M., Kerle, N. and Vosselman, G., 2015. Identification of damage in buildings based on gaps in 3D point clouds from very high resolution oblique airborne images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, pp.61-78.

[20] Chen, D., Zhang, L., Mathiopoulos, P.T. and Huang, X., 2014. A Methodology for Automated Segmentation and Reconstruction of Urban 3-D Buildings from ALS Point Clouds. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, *7*(10), pp.4199-4217.

[21] Lari, Z. and Habib, A., 2014. An adaptive approach for the segmentation and extraction of planar and linear/cylindrical features from laser scanning data.*ISPRS Journal of Photogrammetry and Remote Sensing*, *93*, pp.192-212.

[22] Truong-Hong, L., 2011. *Automatic generation of solid models of building Façades from LiDAR data for computational modelling*. University College Dublin.

[23] Truong‐Hong, L., Laefer, D.F., Hinks, T. and 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), pp.112-129.

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

[25] Laefer, D.F., Hinks, T. and Carr, H., 2010. New possibilities for damage prediction from tunnel subsidence using aerial LiDAR data. In *Presented at Geotechnical Challenges in Megacities, ISSMGE International Geotechnical conference, June 7-10, 2010, Moscow, Russia*.Vol. 2, 622-629.

[26] Singh, M. and Laefer, D.F., 2015. Recent Trends and Remaining Limitations in Urban Microclimate Models. *Open Urban Studies and Demography Journal*, *1*(1).

[27] Vo, A.V., Truong-Hong, L. and Laefer, D.F., 2015, July. Aerial laser scanning and imagery data fusion for road detection in city scale. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International* (pp. 4177-4180). IEEE.