

A SLIDING WINDOW METHOD FOR DETECTING CORNERS OF OPENINGS FROM TERRESTRIAL LIDAR DATA

Jiaqiang Li ^{1,*}, Biao Xiong ^{2,3}, Filip Biljecki ⁴, Gerhard Schrotter ^{1,5}

¹ ETH Zurich, Future Cities Laboratory, Singapore-ETH Centre, Singapore - jiaqiang.li@arch.ethz.ch

² School of Computer Science and Technology, Wuhan University of Technology

³ Dipper 3D, the Netherlands - b.xiong@whut.edu.cn

⁴ Department of Architecture, National University of Singapore, Singapore - filip@nus.edu.sg

⁵ Geomatics Engineering (GeoZ) City of Zurich, Switzerland - gerhard.schrotter@zuerich.ch

KEY WORDS: Sliding Window, Corners Detection, Openings Detection, LiDAR, Point Clouds, LoD3 Modelling

ABSTRACT:

Architectural building models (LoD3) consist of detailed wall and roof structures including openings, such as doors and windows. Openings are usually identified through corner and edge detection, based on terrestrial LiDAR point clouds. However, singular boundary points are mostly detected by analysing their neighbourhoods within a small search area, which is highly sensitive to noise. In this paper, we present a global-wide sliding window method on a projected façade to reduce the influence of noise. We formulate the gradient of point density for the sliding window to inspect the change of façade elements. With derived symmetry information from statistical analysis, border lines of the changes are extracted and intersected generating corner points of openings. We demonstrate the performance of the proposed approach on the static and mobile terrestrial LiDAR data with inhomogeneous point density. The algorithm detects the corners of repetitive and neatly arranged openings and also recovers angular points within slightly missing data areas. In the future we will extend the algorithm to detect disordered openings and assist to façade modelling, semantic labelling and procedural modelling.

1. INTRODUCTION

With the increase in global warming, industrialization, urbanization, etc. the planning of every detail in a city becomes highly important. Each step in urban planning has to be carefully analyzed and worked upon to derive best possible solutions among various stakeholders. In a smart city a 3D city model serves as a platform to run e.g. environmental simulations, master planning applications, disaster management applications, etc. The digital twin of a city should represent the reality as close as possible with high level of detail (LoD). City models at LoD3, as described in the CityGML 2.0 standard (Open Geospatial Consortium, 2012), are architecturally detailed models. They contain openings (windows, doors), roofs mapped in more detail (including dormers, chimneys, and roof overhangs), and contain significant façades structures (Gröger and Plümer, 2012). Besides providing a quite realistic representation for visualisation purposes (Garnett and Freeburn, 2014), such models are highly valuable for applications, like assessing energy retrofits (Previtali et al., 2014), energy simulations (Southall and Biljecki, 2017), and calculating the solar potential of walls (Catita et al., 2014). LoD3 models can be gained from architectural plans (Donkers et al., 2016) and with procedural modelling (Müller et al., 2006, Martinović et al., 2015).

Many of the existing 3D city models have a sufficient level of detail (LoD) to fulfill specific requirements, but realistic information, such as the precise geometry of façades, is missing. Light Detection and Ranging (LiDAR), as known as laser scanning, has been widely used in digitizing façades through point clouds. Those point clouds contain real 3D geometry information. Laser scanners are becoming more efficient, have a better portability,

and are improving in accuracy (Shan and Toth, 2018). With integrating a camera, the true colour value is able to be captured and registered to those points. Therefore, point clouds are an obvious choice to create 3D city models at a higher LoD (Biljecki et al., 2016).

Façade modelling from dense point clouds is usually based on boundary representation (B-Rep) that consist of vertices, edges, and faces. The relation between the three components can be automatically generated by triangulation where the representation of façades are triangular meshes. Albeit the mesh generation is efficient, the solid model is usually insufficient. Afterwards numerical optimizations are needed to mitigate noise, simplify boundaries and so on (Haala et al., 2006). Another type of representation is a polygon mesh that indicates the connected relation of points. If the point correctly represent the corner of a façade and its elements, the model becomes easy to use and edit with a commercial software like 3ds Max. Although a lot of work has been done for the boundary detection of openings, the boundary points are detected individually. They are sensitive to the noise and neighbouring points. Furthermore, there are partitioning methods that split a façade into blocks following the features of façade elements. However, the corners of openings are usually out of consideration.

We aim for automatically detecting the corner of openings on façade point clouds in this work. Generally, the façade elements are following architectural rules representing rectilinear and repetitive structures. We are taking advantage of these traits to detect the corner points of openings. However, the point cloud data contains certain unexpected points nearby the boundary of openings. This is possibly caused by objects, such as window bars or decals, curtains, plants and so on. It increases the difficulty of point cloud processing, in particular analysing points

*Corresponding author

within their neighbourhoods. Instead of searching out individual points of boundaries, we propose to inspect the change between walls and openings to directly locate the border lines. As the boundary points and lines are sought out according to the wide strip-shape areas of the façade, the influence of discrete noise points will be mitigated.

In this paper, we describe a method for detecting opening corners based on a global-wide sliding window. This window performs horizontal and vertical retrieval of the façade separately and calculates the gradient of the number of points in each search area. By statistically analysing the gradient, the area corresponding to the change of the façade element can be located. From these areas, intersections of the façade element transitions are extracted. Finally, by intersecting these boundaries, corners of openings are generated.

The contributions of this paper are as follows. First, we propose a robust algorithm to locate corner of façade openings from terrestrial LiDAR points. The algorithm is robust to noise and outliers, which are very common in terrestrial LiDAR data. Second, we introduce a novel algorithm to recover façade openings from occlusion areas based on global and symmetry information.

The paper is structured as follows. Requirements and challenges of opening detection are discussed in Section 2. We introduce our solution for detecting corners of openings by a sliding window method in Section 3. The experimental results with mobile and terrestrial LiDAR data are presented and analysed in Section 4. The conclusion and future work are given in Section 5.

2. RELATED WORK

A considerable amount of work has been done on detecting openings using point clouds from images (Müller et al., 2007, Ceylan et al., 2012) and LiDAR (Kelly et al., 2017). Our work is based on LiDAR point clouds, including static terrestrial and mobile LiDAR data. Usually there are no points in openings areas which represent holes. This characteristic has been used in the openings detection, such as extracting edge points by detecting long TIN edges and corrected the shape by concave hull (Pu and Vosselman, 2009). Additionally, the combination of α -shape and delaunay triangulation method is used (Arikan et al., 2013). Furthermore, symmetry information has been used to detect openings with descriptors (Kerber et al., 2013). Moreover, a semi-automatic reconstruction of openings is done mainly by snapping with energy functions, symmetry information and human interaction (Nan et al., 2010). In this paper, we focus on detecting corners of openings from point clouds, the modelling part is not considered.

Cell decomposition is a typical method for automatically detecting openings on a façade (Becker and Haala, 2007). It searches the boundary points of openings depend on their borders. For instance, if there is no point on the right side within a radius search area, then this point belongs to the left edge points. Similarly, the other edge points (right, top and bottom) can be searched out. By using these edge points, horizontal and vertical edge lines are estimated to partition the façade into 3D cells which are classified with a binary map afterwards. The classification facilitates the understanding of façade grammar (Becker, 2009). However, the initial step of searching boundary points is sensitive to noises, which requires a façade point cloud with high quality (Zolanvari and Laefer, 2016). Different from this method, our approach directly inspects the change of façade elements to locate the edge

lines that mitigate the influence of noise points and partly occluded areas.

An adaptive partitioning method for façade elements is proposed that roughly segment a façade and boundary of openings by a penalty function with RANSAC approach. Then, the similarity of each nearby slice is measured to identify façade elements. Same clusters are consistently split and rectified until the evaluation of the result meets predefined thresholds. Each sub-divided block is recursive partitioned to extract more façade elements (Shen et al., 2011). The method is adaptive to many types of façade (e.g. with balcony, with protrusion and conical). However, it is sensitive to patterns (e.g. non-rectilinear, irregular repetitive, etc.) on a façade. Although it partitioned the façade, a further estimation to the angular point of façade elements is still necessary. Later, as an extension of this method but most focus on façade elements, the local lattice is used (Mesolongitis and Stamos, 2012). The deviation of normal by each two neighbouring points is computed for lattice. Then, boundary lines are roughly fit by using Hough Transform with the grid. With a defined energy function, repetitive structures are detected and labelled (Wang et al., 2016). However, depending on grid fitting and penalty functions, the method needs tremendous empirical thresholds for the lattice hypothesis and weighting parameters. In contrast, our approach extracts the boundaries by analysing the gradient of changes of points. These changes are displayable and therefore help to determine the parameters.

Recently, a slicing method is proposed for façade and window extraction from point clouds (Zolanvari and Laefer, 2016). The method firstly segments the primitive of a façade by RANSAC. It slices the segmented façade into numbers of portions depending on the width of the façade and an empirical value. Each slice represents one cluster unless a gap breaks the cluster into two. Then, a projection within a local coordinate system is applied to the points in each cluster, where the endpoints of the projection line are extracted as boundary points. This method detects openings in one direction that reduce the computation cost and provide relatively high accuracy with less manipulate. However, it requires sufficient density of point clouds, which is hard to be ensured when using mobile LiDAR data. In addition, occlusions are not under consideration in this method. Instead, the approach we propose considers occlusion conditions in the sliding window. Consequently, it is advisable in partly occlusive point clouds.

There are some challenges in opening detection, which we seek to overcome in this paper:

1. Occlusion: Depending on different measurements and measuring angles, the point density of the boundary of openings is unstable. For example, when using mobile LiDAR, the upper boundary of a window has higher point density and reliability than that of the lower part. This is caused by a mobile laser beam emitted from relatively 2 meters above the ground could be occluded by windowsills where has less return from the bottom of the window.
2. Inhomogeneous and low point density: The point density of terrestrial LiDAR data is dependent on beam divergence, the range to targets, and other system dependent factors. Typically, the higher the part of a façade, the lower the point density.
3. Outliers and noise: Even extracted the primitive of a façade, there are a certain number of non-interested points locate

nearby openings. Such points are summarized as noise which is hardly distinguished from the point cloud aspect.

3. PROPOSED METHODOLOGY

Our work aims at detecting corner points of openings on a façade. The input of our process is a raw terrestrial LiDAR data, including static and mobile laser point clouds. Mostly, the quality of raw point clouds are inconsistent, depending on the environment complexity and measurements. The collected points are unstructured and contain a certain degree of noise, outliers and other influences as mentioned in Section 2. Therefore, the first and necessary step is to analyse the raw data (i.e. Which attribution is available? How is the distribution of points? Where is the research area of interest? Etc.). In some of our cases, the true colour is registered to the point cloud, but the variation of colour representing façade elements is not sharp enough to be differentiated. In addition, different buildings have diverse colours that increase the unavailability of the RGB values. Hence, we only use the geometry information. Others, such as colour values, are not under consideration in our case.

3.1 Overview of the method

Our approach of detecting openings corners is based on LiDAR point clouds of urban building façades. The facade point cloud collected by static or mobile LiDAR contains uncertain depths. This uncertainty will increase the difficulty for processing. For ease of the difficulty, we clip a rough primitive of the facade from the raw data. It is then projected to the local Y-Z coordinate plane. The later process consists of three main steps, which are elaborated on in the continuation.

1. Sliding Window Search

We propose a sliding window to inspect the façade point cloud. The search window is defined in the shape of a wide stripe containing two equal parts. While searching the whole façade by sliding the window, the index of each part and its interior points will be recorded. Since the point cloud in the sliding window already covers the occlusion, the analysis of global information in the striped window can effectively reduce the influence of occlusion and density inconsistency.

2. Gradient Generation and Change Detection

To detect the change between façade elements, we calculate the gradient of change based on the number of points in the first and second half of the sliding window. For example, if it slides from the wall to openings where the first part has higher density than that of the second part, the gradient is basically negative. Peaks of the gradient values are selected, representing the position of the change. Sometimes, outliers and noise affect the gradient value. We mitigate these effects by adaptively counting the position separations to derive symmetry information.

3. Angular Point Intersection

After selecting the gradient peak, the corresponding sliding window can be located. As gradients are calculated through two parts of the sliding window, we use the intersection of these two parts to approximate the boundary between openings and the wall. Meanwhile, angular points can intersect from the horizontal and vertical borderlines.

3.2 Pre-processing

In the most of cases, not all the collecting points are coplanar. The laser beam goes through windows or return from protrusions, which leads points to attach (or with a certain level of distance) to the primitive of a façade. These points may influence the later process. For example, if a projection is made when there is an angle between the point and the projecting direction, non-interest points will be projected onto the reference plane that alters the original geometry of the façade. Several works have been done on automatically extracting primitive from an interested research area (Mesolongitis and Stamos, 2012, Xiong et al., 2015, Zolanvari and Laefer, 2016). For ease of the issue, we clip the primitive of façade points from the raw data using CloudCompare (<http://www.cloudcompare.org>). It is then projected to the local Y-Z coordinate plane by using Hessian normal form (Rusu and Cousins, 2011). The input point cloud is unstructured, even it has been projected to a 2D plane. For detecting features, we structure façade points with the octree data structure (Meagher, 1982).

3.3 Sliding Window Search

Normally, façade elements are aligned following architectural rules. For instance, a slab is located between two rows or columns of openings when looking from the outside of a building. In another word, starting from an opening (e.g. a window), the area above belongs to a wall. Such rule is applicable in general standard buildings. To exploit this rule in the structured point cloud, we develop a sliding window method to inspect the façade horizontally from the bottom to the top and vertically from the left to the right (see Figure 1).

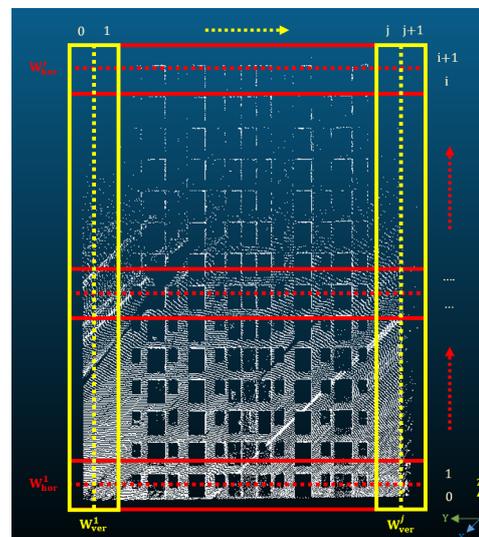


Figure 1. Sliding Window Search. Red frames show the sliding window search in the horizontal direction. The yellow frame represents the vertical search window. Note that, for the sake of a clear visual representation, the size of the sliding window is enlarged in the figure.

Taking the horizontal direction as an example, we first set up an initial window (W_{hor}^1 as shown in Figure 1). The size of the sliding window depends on its depth, width and height. Since the window search is performed on the projected two-dimensional plane, in order to reduce the computational cost, we set the depth to 0 cm. The height is determined according to the pattern of the façade. As the prior knowledge of the architecture aspect, the slab between each floor is around 50 cm. Thus we assume

that a striped window with the height of 40 cm (inner parts of the sliding window are equally 20 cm) is suitable to inspect the change of façade structures. In order to globally analyse the façade point cloud to reduce the noise effect, the width of the sliding window should be sufficient to cover the width of the façade data $Width_f = |Y_{max} - Y_{min}|$. To ensure all points can be counted, we slightly enlarge the width of the sliding window. The increase is half the height of the window. Therefore, the width of the horizontal sliding window ($Width_w$) is defined as $Width_w = Width_f + \frac{1}{2}Height_w$

After constructing the initial frame of the window, we start to define the sliding of the window. There are four key parameters need to be defined, including the starting and ending position, as well as the direction and distance of the slide. The position of the lower left corner of the window is marked as the starting position $(X_{hor}, Y_{hor}, Z_{hor})_{start}$. In order to cover the entire area, the window based on this initial position needs to include the lowest and most marginal point $(X, Y, Z)_{min}$ in the façade data. Therefore, we add a tolerance to this starting position, which is express as $(X_{hor}, Y_{hor}, Z_{hor})_{start} = (X, Y, Z)_{min} - \frac{1}{4}(Depth, Height, Height)_w$. Then, we give the direction of movement of the horizontal sliding window to the Z axis. It slides at half the height of the window each time ($dist_{move} = \frac{1}{2}Height_w$). When it moves once, the initial position is updated according to the moving distance. The index of each point in the upper and lower parts of the window is stored for subsequent feature extraction and global information analysis. After recording all the information, the window will automatically slide along the direction and distance until it reaches the highest point (Z_{max}) of the façade data.

Similarly, for vertical search, the depth is still set to 0 cm. We set the width of the vertical sliding window to 40 cm. As half the width of the window, the sliding distance is 20 cm. The height of the vertical sliding window is defined as $Height_w = Height_f + \frac{1}{2}Width_w$. Assigning Y axis as the moving path, the vertical sliding window inspects the façade starting at the same location as the horizontal search until it traverses completely the façade.

3.4 Gradient Generation and Change Detection

Feature extractions from point clouds often rely on the relationship between a single point and its neighbourhood. This is sensitive to the noise surrounding the search point. Our proposed solution is to use a change of the number of points (see equation 1) to establish a relationship to the overall area in the sliding window. Features, i.e. the change of façade elements, can be extracted by these gradient values and the corresponding window positions.

Following the mentioned architectural rule, it is assumed that the number of points within the sliding window will be reasonably large when it moves to the slab or other place that belongs to the wall. On the other hand, when the area inside the sliding window belongs to openings where basically no representing points, the amount of point will be relatively small. We compare the points within the upper and lower parts (left and right in the vertical direction) of the sliding window. If the sliding window arrives at the boundary between slab and openings, the number of points in the two parts of the window will change greatly. Therefore, based on the number of points acquired in the previous search, we extract features by calculating the gradient of the number of points in the sliding window.

However, most façade point clouds have inhomogeneous density. For example, due to the scanning distance and occlusions, the density of points in the upper part of the façade is less than that in the lower part. This challenge will increase the uncertainty of the range of gradient changes. To alleviate this problem, we assume that the property of points located in the current sliding window is similar, such as they have approximate point distances. Therefore, we can effectively compress the gradient values to the 0 to 1 interval by dividing the difference between the points on the upper and lower parts of the sliding window by the number of points, which is express as

$$\mathbf{G} = \frac{\mathbf{N}_{i+1} - \mathbf{N}_i}{\mathbf{N}_{i+1} + \mathbf{N}_i} \rightarrow \begin{cases} 1 & \mathbf{N}_i \ll \mathbf{N}_{i+1} \\ 0 & \mathbf{N}_i \approx \mathbf{N}_{i+1} \\ -1 & \mathbf{N}_i \gg \mathbf{N}_{i+1} \end{cases} \quad (1)$$

Where \mathbf{N}_i means the number of points in the upper (left) part of the sliding window; \mathbf{N}_{i+1} is the number of points in the lower (right) part. These two numbers are positive. When \mathbf{G} approaches to 1 that means the number of points in the latter area is much larger than the former one. Conversely, a negative value means that there are more points in the first half. When \mathbf{G} approaches 0, it means that the points of the two parts of the sliding window are similar.

The sign of the gradient \mathbf{G} indicates the variation trend of the façade elements. Since the number of points must be non-negative, the sign of the gradient depends on the difference between the first and second parts of the sliding window. When the search window slides from wall to openings, the gradient of the number of points will have a change from positive to negative. However, this process is subject to a certain degree of noise, which caused by the points from non-interest objects. For analysing the change position and noise influences, we plot the gradient values as shown in Figure 2.

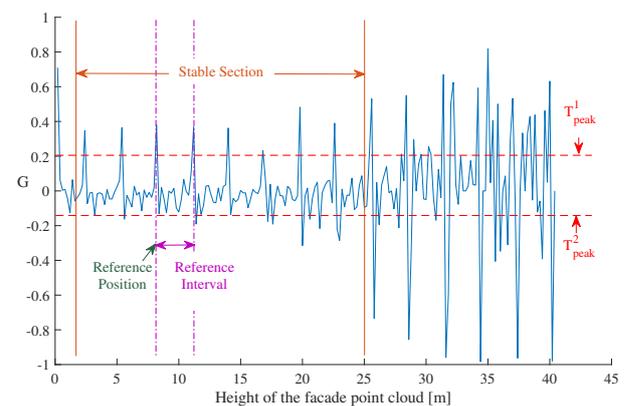


Figure 2. Statistical analysis of gradients \mathbf{G} in sliding windows

By the assumption, the peaks of the gradient in the statistic figure generally represent the change between openings and wall. Such peak values can be extracted by two filters. \mathbf{T}^1_{peak} restricts the positive values (e.g. a window to a slab) that values below this threshold will be filtered out. In contrast, the gradients staying over the constraint will be treated as peak values. Similarly, we keep values which are less than the \mathbf{T}^2_{peak} constraint (e.g. a wall to an opening). Each change should perform a sharp rising or down trend following a regular pattern.

However, due to the low point density and other effects, the performance of gradients becomes unstable in the higher part of the façade. To reduce the impact, we pre-defined a stable section to extract symmetric information to constrain the overall region. First, we calculate the interval between all extracted peaks and sort them in descending order. Then, the mean \bar{I}_{peak} and median \tilde{I}_{peak} of the peaks in the given stable section are calculated. By comparing these two numbers, the smaller one is taken as the symmetric information representing the reference interval in Figure 2. The definition is expressed as

$$I_{ref} := \min(\tilde{I}_{peak}, \bar{I}_{peak}) \quad (2)$$

For each neighbouring peaks, the separation distance should be larger than the reference interval I_{ref} (see equation 2). By this symmetry information, the pseudo peaks appearing in the extracted peaks can be filtered out. Thus positions of the sliding window representing the change of façade elements are detected, which are represented by the remaining peaks.

3.5 Angular Point Intersection

The final step in the approach is to intersect the corners of openings by extracting the horizontal and vertical intersection of the wall and the openings. From the position of sliding window given in the previous step, we can locate the boundary between the wall and openings. Since each gradient value is obtained from the points corresponding to the upper and lower parts of the sliding window, in order to obtain robust borderlines, we use the intersection line of the two parts to approximate the boundary between different façade elements.

Empirically, the bottom and the top of the façade are not in the stable section, due to occlusions and noise effects. It will cause misleading boundaries if we extract them following the search order of the sliding window. Thereby, we assign a reference peak position. The extraction of horizontal (vertical) borderlines is executed from the referenced position to the bottom (left) and to the top (right) separately under the constraint of the derived reference interval I_{ref} . Finally, the corners of openings can be detected by intersecting these horizontal and vertical boundary lines.

4. EXPERIMENTS

The approach has been tested on static terrestrial LiDAR data (Figure 3) and mobile LiDAR point clouds (Figure 4, 5). As mentioned in Section 3, we set the height of the horizontal sliding window and the width of that in vertical direction to 40 cm for all the cases as shown in this paper. The reference position is pre-defined at 40 which can be tweaked depending on different cases. Empirically, we find that the thresholds of peak selection set to $T_{peak}^2 < -0.15$ and $T_{peak}^1 > 0.15$ are adaptable to our cases in static and mobile terrestrial point clouds.

Figure 3 shows the sliding window method for corners detection from a static terrestrial LiDAR point cloud and the gradient in the horizontal sliding window. The corners of openings have been generally detected with the settings. Since we projected the 3D point cloud onto a 2D plane, the points with certain depth which do not belong to this plane are also counted after projection. It is sensitive to irregular structures when using sliding window method. An example is shown in Figure 3 (left) where the top of the façade has a modern design, the gap within that area also

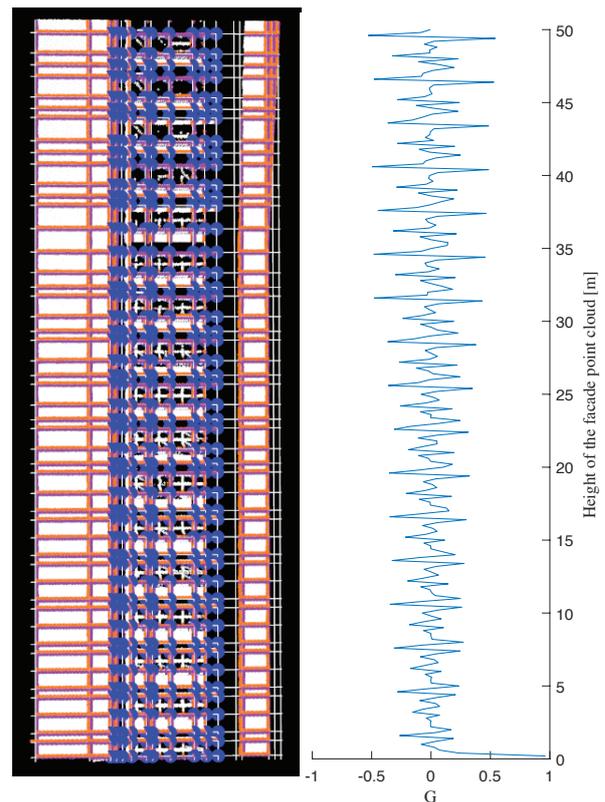


Figure 3. Result of detecting opening corners by the sliding window method from the static terrestrial LiDAR data. The height and width of the façade point cloud (left) are 50.00 m and 16.81 m respectively. The plot on the right shows the gradient value in the horizontal sliding window.

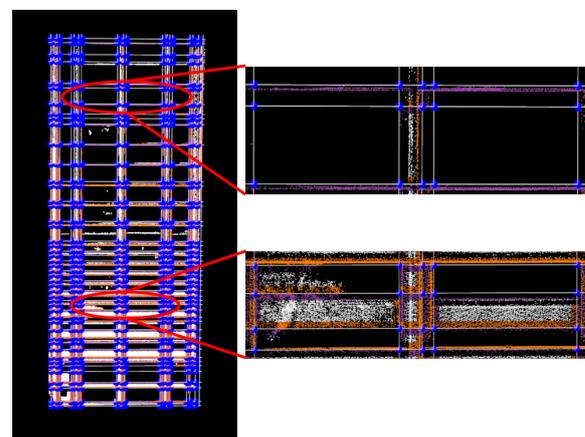


Figure 4. Recovery of openings within low point density (upper right) and noise areas (lower right). The height and width of the façade point cloud (left) from mobile LiDAR are 52.52 m and 21.61 m respectively.

has a sharp gradient in the sliding window. Therefore, an unexpected vertical line is extracted. Focusing on openings detection, this case can be adjusted with pre-processing or post-processing by limiting the range of intersection.

We also experiment the sliding window method on mobile LiDAR data. Although the lower part of the façade point cloud from mobile LiDAR has higher point density, the area has more possi-

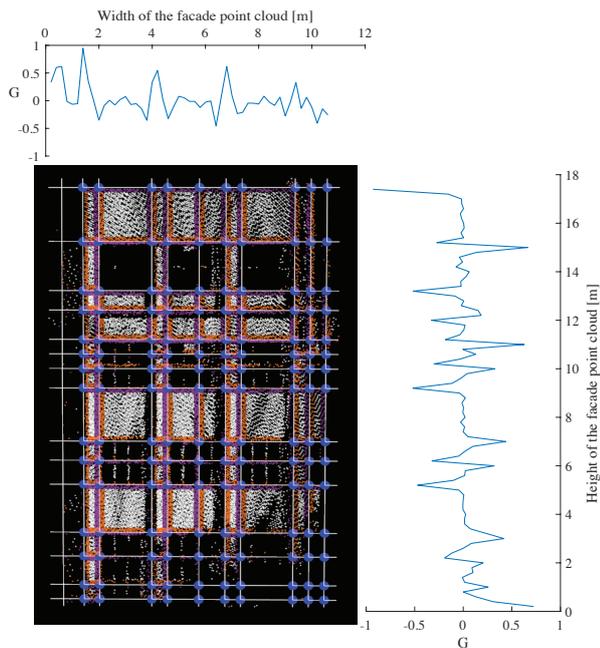


Figure 5. Window detection by the sliding window method from the mobile LiDAR data. The height and width of the façade point cloud are 17.66 m and 10.97 m respectively. The plot of vertical gradient values is on the top of the figure and that of the horizontal gradients is placed on the right-hand side.

bility to contain non-interested points. As shown in Figure 4 (low right), an opening is occluded that increased noise points occupied nearly a half of the open area irregularly. Our method analysed the stripe-wide information and detected the corner points within that area. Additionally, due to the character of mobile laser scanning, the upper site of the façade lacks information. The algorithm is able to estimate the boundary of openings in this site based on the existing information from the lower part. For providing a robust detection, the estimation contains symmetry information, that is, the reference spacing is used to constrain the selection of peaks.

Figure 5 is another sample of mobile LiDAR data in our study. Peaks in the plots represent the sharp variations of point density between inner parts of the sliding window. Based on the global and symmetry information, the sliding window method refined the façade structure of the right lower part where has a certain level of occlusions. Comparing to the statistical figure of static terrestrial laser scanning, that of the mobile LiDAR has less distinct identity of patterns. In such case, our method is still valid.

However, there are still some limitations should be improved in our method. Since the sliding window approach is mainly based on the gradient of the number of points in globally wide windows, it is subject to abnormally aligned structures. If the openings are not regularly arranged, there may increase many lines (or no lines) depends on the gradient and intersect unexpected points. Although our method can recover corners of openings within partly noise and missing data areas, it is ineffective in those mostly or fully occluded areas (e.g. the area with no points) where has no sharp change of gradients. Because of the inherent nature of the method, closed doors and windows with blinds are hardly detected when full of points are on a flat surface.

5. CONCLUSIONS

We propose an approach to detect corner points of repetitive openings from terrestrial LiDAR point clouds. The approach searches and analyses the façade with a globally wide sliding window that mitigates the noise effect in horizontal and vertical directions based on octree structure. Through computed gradients in each sliding window, we apply statistical analysis and derive the symmetry information to inspect changes of façade elements. The approach detects corners of openings and recovers corner points in a partly missing data area. However, mostly occluded areas and not aligned openings are the limitations.

In the future we plan to improve our approach to be adaptive to abnormally arranged façade elements. To improve the accuracy of the detection, the depth information and integration of image features will be considered. The detected corner points will be used to classify façade structures and serve as an initial input for feature extraction from images and semantic labelling by machine learning.

ACKNOWLEDGEMENTS

This material is based on research/work supported by the National Research Foundation under Virtual Singapore Award No. NRF2015VSG-AA3DCM001-024. We would like to thank the Singapore Land Authority (SLA) and Philipp URECH from Ecosystem Services in Urban Landscapes Group at the Future Cities Laboratory, for providing the terrestrial point cloud data.

REFERENCES

- Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M. and Maierhofer, S., 2013. O-snap: Optimization-based snapping for modeling architecture. *ACM Transactions on Graphics (TOG)* 32(1), pp. 6.
- Becker, S., 2009. Generation and application of rules for quality dependent façade reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(6), pp. 640–653.
- Becker, S. and Haala, N., 2007. Refinement of building facades by integrated processing of lidar and image data. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science* 36, pp. 7–12.
- Biljecki, F., Ledoux, H. and Stoter, J., 2016. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems* 59, pp. 25–37.
- Catita, C., Redweik, P., Pereira, J. and Brito, M. C., 2014. Extending solar potential analysis in buildings to vertical facades. 66, pp. 1–12.
- Ceylan, D., Mitra, N. J., Li, H., Weise, T. and Pauly, M., 2012. Factored facade acquisition using symmetric line arrangements. In: *Computer Graphics Forum*, Vol. 31 number 2pt3, Wiley Online Library, pp. 671–680.
- Donkers, S., Ledoux, H., Zhao, J. and Stoter, J., 2016. Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS* 20(4), pp. 547–569.
- Garnett, R. and Freeburn, J. T., 2014. Visual Acceptance of Library-Generated CityGML LOD3 Building Models. *Cartographica* 49(4), pp. 218–224.

- Gröger, G. and Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing* 71, pp. 12–33.
- Haala, N., Becker, S. and Kada, M., 2006. Cell decomposition for the generation of building models at multiple scales. *Photogrammetric Computer Vision*.
- Kelly, T., Femiani, J., Wonka, P. and Mitra, N. J., 2017. Bigsur: Large-scale structured urban reconstruction. *ACM Transactions on Graphics*.
- Kerber, J., Bokeloh, M., Wand, M. and Seidel, H.-P., 2013. Scalable symmetry detection for urban scenes. In: *Computer Graphics Forum*, Vol. 32number 1, Wiley Online Library, pp. 3–15.
- Martinović, A., Knopp, J., Riemenschneider, H. and van Gool, L., 2015. 3D All The Way: Semantic Segmentation of Urban Scenes from Start to End in 3D. In: *CVPR 2015: The 28th IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Boston, United States, pp. 4456–4465.
- Meagher, D., 1982. Geometric modeling using octree encoding. *Computer graphics and image processing* 19(2), pp. 129–147.
- Mesolongitis, A. and Stamos, I., 2012. Detection of windows in point clouds of urban scenes. In: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference On*, IEEE, pp. 17–24.
- Müller, P., Wonka, P., Haegler, S., Ulmer, A. and van Gool, L., 2006. Procedural modeling of buildings. *ACM Transactions on Graphics* 25(3), pp. 614–623.
- Müller, P., Zeng, G., Wonka, P. and Van Gool, L., 2007. Image-based procedural modeling of facades. In: *ACM Transactions on Graphics (TOG)*, Vol. 26number 3, ACM, p. 85.
- Nan, L., Sharf, A., Zhang, H., Cohen-Or, D. and Chen, B., 2010. Smartboxes for interactive urban reconstruction. *ACM Transactions on Graphics (TOG)* 29(4), pp. 93.
- Open Geospatial Consortium, 2012. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Technical report.
- Previtali, M., Barazzetti, L., Brumana, R., Cuca, B., Oreni, D., Roncoroni, F. and Scaioni, M., 2014. Automatic façade modelling using point cloud data for energy-efficient retrofitting. *Applied Geomatics* 6(2), pp. 95–113.
- Pu, S. and Vosselman, G., 2009. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing* 64(6), pp. 575–584.
- Rusu, R. B. and Cousins, S., 2011. 3d is here: Point cloud library (pcl). In: *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Shanghai, China.
- Shan, J. and Toth, C. K., 2018. *Topographic laser ranging and scanning: principles and processing*. CRC press.
- Shen, C.-H., Huang, S.-S., Fu, H. and Hu, S.-M., 2011. Adaptive partitioning of urban facades. In: *ACM Transactions on Graphics (TOG)*, Vol. 30number 6, ACM, p. 184.
- Southall, R. and Biljecki, F., 2017. The VI-Suite: a set of environmental analysis tools with geospatial data applications. *Open Geospatial Data, Software and Standards* 2, pp. 23.
- Wang, J., Xu, Y., Remil, O., Xie, X., Ye, N., Yi, C. and Wei, M., 2016. Automatic modeling of urban facades from raw lidar point data. In: *Computer Graphics Forum*, Vol. 35number 7, Wiley Online Library, pp. 269–278.
- Xiong, B., Jancosek, M., Elberink, S. O. and Vosselman, G., 2015. Flexible building primitives for 3d building modeling. *ISPRS Journal of Photogrammetry and Remote Sensing* 101, pp. 275–290.
- Zolanvari, S. I. and Laefer, D. F., 2016. Slicing method for curved façade and window extraction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* 119, pp. 334–346.