POINT CLOUD DATA CONVERSION INTO SOLID MODELS VIA POINT-BASED VOXELIZATION

Tommy Hinks[(1)], Hamish Carr[(2)], Linh Truong-Hong [(3)], and Debra F. Laefer[(4)*]

[(1)] PhD, School of Computer Science & Informatics, University College Dublin (UCD), CSI/A0.09, Belfield, Dublin 4, Ireland. Email: tommy.hinks@ucd.ie

[(2)] Senior Lecturer, School of Computing, Faculty of Engineering, University of Leeds, EC Stoner Building 6.06, UK. Email: h.carr@leeds.ac.uk

[(3)] PhD, Urban Modelling Group (UMG), School of Civil, Structural and Environmental Engineering (SCSEE), UCD, Newstead G67, Belfield, Dublin 4, Ireland. Email: linh.truonghong@gmail.com

[(4)]* Associate Professor, Lead PI, UMG, SCSEE, UCD, Newstead G25, Belfield, Dublin 4, Ireland. Email: debra.laefer@ucd.ie, corresponding author

ABSTRACT

Automated conversion of point cloud data from laser scanning into formats appropriate for structural engineering holds great promise for exploiting increasingly available aerially- and terrestrially-based pixelized data for a wide range of surveying related applications from environmental modeling to disaster management. This paper introduces a point-based voxelization method to automatically transform point cloud data into solid models for computational modeling. The fundamental viability of the technique is visually demonstrated for both aerial and terrestrial data. For aerial and terrestrial data, this was achieved in less than 30 seconds for data sets up to 650k points. In all cases, the solid models converged without any user intervention when processed in a commercial Finite Element Method program.

KEYWORDS: Terrestrial laser scanning aerial laser scanning, LiDAR, voxelization, computational modeling, solid models, Finite Element

INTRODUCTION

Laser scanning has achieved great prominence within the Civil Engineering community in recent years for topics as divergent as coastline monitoring (Olsen et al. 2009 and 2011), airport layout optimization (Parrish and Nowak 2009), and ground displacement identification for water system risk assessment (Stewart et al. 2009). Additionally, there has been strong motivation to obtain further functionality from laser scanning and other remote sensing data including three-dimensional (3D) volume estimation for mining (Mukherji 2011), road documentation (Dong et al. 2007) structural identification (Shan and Lee 2005; Zhang et al. 2011), and emergency planning (Laefer and Pradhan 2006). Furthermore, computational responses of city-scale building groups are increasingly in demand for heightened urbanization, disaster management, and microclimate modeling, but input data are typically too expensive due to the need for manual surveying. Furthermore, current software tools for transforming remote sensing data into computational models have one or more of following problems: a low degree of reliability; an inability to capture potentially critical details; and/or a need for a high degree of human interaction. To date, a seamless, automated, and robust transformation pipeline from remote sensing data into city-scale computational models does not exist. This paper lays the groundwork for key advancements in such a pipeline. Herein, the proposed procedure reconstructs building façades from point clouds, which is a fundamental step for generating city-scale, computational models.

FAÇADE RECONSTRUCTION

In recent years, developments in laser scanning technology and flight path planning allow

ALS to acquire point cloud data quickly and accurately at a city-scale, thereby having the potential for reconstructing 3D building surfaces across an entire city in nearly real-time. A number of approaches based on semi-automatic (e.g. Lang and Forstner 1996) and automatic (e.g. Henricsson et al. 1996) approaches have been proposed to reconstruct building models from such data sets, but automatically extracting highly detailed, accurate, and complex buildings still remains a challenge (Haala and Kada 2010). The semi-automatic procedures need human operator intelligence. The automatic visual modeling of urban areas from ALS data tends to extract sample points for an individual building by applying segmentation techniques and then reconstructs each building individually. In such cases, vertical façade surfaces are not portrayed in detail, and outlines may be of relatively low accuracy, unless ground planes are integrated, which requires either a priori information or manual intervention. Unfortunately, the effectiveness of engineering modeling often depends largely on the geometric accuracy and details of the building models, thus the current mismatch.

Presently, commercial products are generally semi-automatic (Laefer et al. 2011), while in the computer graphics and photogrammetry communities, research has focused on automated surface reconstruction from dense and regular sample points [e.g. (Hoppe 1994; Kazhdan et al. 2006)]. Unfortunately, ALS data are often sparse and irregular and may contain major occlusions on vertical surfaces due to street- and self-shadowing (Hinks et al. 2009). Dedicated urban modeling, surface reconstruction approaches generally use the major building planes (Chen and Chen 2008) and can be described as either model-driven or data-driven. Model-driven techniques use a fixed set of geometric primitives, which are fitted to the point data. Such techniques can be effective when a data set is sparse, since the fitting of geometric primitives does not require complete data. In contrast, data-driven techniques derive surfaces directly from the point data and are capable of modeling arbitrarily shaped buildings. Generally, data-driven approaches are more flexible than model-driven approaches

but are often sensitive to noise in the input data.

For strictly visual representation, model-driven approaches can be effective. For example, Haala et al. (1998) proposed four different primitives and their combinations to derive automatically 3D building geometry of houses from ALS and existing ground planes. Similarly, Maas and Vosselman (1999) introduced an invariant moment based algorithm for the parameters of a standard, gabled-roof house type, which allowed for modeling asymmetric elements such as dormers. However, these efforts assume homogenous point distributions, which is unrealistic. You et al. (2003) also adapted a set of geometric primitives and fitting strategies to model complex buildings with irregular shapes, but the approach required user intervention and generated only limited wall details. Hu et al. (2004) also used a combination of linear and non-linear fitting primitives to reconstruct a complex building, in which aerial imagery was used to refine the models.

In contrast, many data-driven techniques operating on ALS data reconstruct roof shapes directly from sample points of roof planes. Subsequently, the remainder of the building is simply extruded to the ground level from the roof shape outlines. Vosselman and Dijkman (2001) used a Hough transform for extraction of plane faces (roof planes) from the ALS data, and then 3D building models were reconstructed by combining ground planes and the detected roof planes. Hofmann et al. (2003) introduced a method to extract planar roof faces by analyzing triangle mesh slopes and orientations from a Triangular Irregular Network (TIN) structure generated from ALS data. More recently, Dorninger and Pfeifer (2008) used an α-shape approach to determine a roof outline from point clouds of the roof projected onto a horizontal plane. Also, Zhou and Neumann (2010) created impressive buildings for a large urban area by using a volumetric modeling approach, in which roof planes were determined based on a normal vector obtained from analysis of grid cells belonging to roof layers.

However, these models are also extruded and lack vertical wall details.

Therefore, this paper presents an automated approach to convert point clouds of individual buildings into solid models for structural analysis by means of computational analysis, in which the point clouds that were semi-automatically segmented from LiDAR data become the input (Figure 1). Notably, this proposed approach focuses on reconstructing solid models by using voxel grids with the critical parameter as either the voxel size or the number of voxel grids; for more details on collecting aerial and terrestrial laser scanning and on segmenting point clouds see Truong-Hong (2011) and Hinks (2011).

SOLID MODELING

To generate building models directly from point cloud data for engineering simulations [(e.g. finite element method (FEM)] there are three dominant methods: (i) Constructive Solid Geometry (CSG), where objects are represented using Boolean combinations of simpler objects; (ii) Boundary Representations (B-Reps), where object surfaces are represented either explicitly or implicitly, and (iii) spatial sub-division representations, where an object domain is decomposed into cells with simple topological and geometric structure, such as regular grids and octrees (e.g. Goldman 2009; Hoffmann and Rossignac 1996); there are many extensive treatises available for in-depth considerations of this topic (see e.g. Böhm et al. 1984; Rossignac and Requicha 1984,1999).

Generating solid models automatically from point cloud data is particularly important because the cost of manually creating solid models of existing objects is far greater than the associated hardware, software, and training costs. As such, spatial sub-division representations are used extensively for creating solid models of buildings, in which regular grids or octrees are employed to decompose an entire object into non-overlapping 3D

regions, commonly referred to as voxels. Voxels are usually connected and describe a simple topological and geometric structure. In grids, a volume is sub-divided into smaller regions by appropriate planes parallel to the coordinate system axes, typically using a Cartesian coordinate system. An initial voxel bounding all point data recursively divides a volume into eight sub-voxels, organized in a hierarchical structure (Samet 1989). Voxels may be labeled white, black, or gray based on their positions (Figure 2). Black voxels are completely inside the solid, while white voxels are completely outside. Voxels with both black and white children are gray (Hoffmann and Rossignac 1996).

In an application of spatial sub-division for surface reconstruction, Curless and Levoy (1996) presented a volumetric method for integrating range images to reconstruct an object's surface based on a cumulative, weighted, signed-distance function. Unfortunately, the approach is not suited for arbitrary objects. In related work, Guarnieri et al. (2005) built a triangulated mesh of an object's surface by combining a consensus surface (as proposed by Wheeler et al. 1998), an octree representation, and the marching cubes algorithm (Lorensen and Cline 1987). This multi-faceted algorithm can reduce the effect of the noise due to surface sampling, sensor measurements and registration errors. However, for optimal results, the method requires modification of parameters that heavily depend on input data characteristics, such as the voxel size or the threshold value for the angle or the distance between two consecutive neighbor range viewpoints.

VOXELIZATION

Critical to octree/quadree representations for further processing is voxelization. This term describes the conversion of any type of geometric or volumetric object such as curve, surface, solids or computed tomographic data into volumetric data stored in a 3D array of voxels

(Karabassi et al. 1999). Initially, a voxel grid divides a bounded, 3D region into a set of cells, which are referred to as voxels. The division is typically conducted in the axial directions of a Cartesian coordinate system. Before voxelization, three pairs of coordinate values ([x_{min}, x_{max}], [y_{min}, y_{max}], [z_{min}, z_{max}]) are created along the three axes (X, Y and Z), defining a global system (Figure 3). The basic idea of voxelization algorithm is to examine whether or not voxels belong to the object of interest, and to assign a value of 1 or 0, respectively (Karabassi et al. 1999); a further description of voxel grids is available in Cohen and Kaufman (1990).

An initial voxel bounding all point cloud data in 3D Euclidean space R^3 are sub-divided into subset voxels by grids along x-, y-, and z- coordinates in a Cartesian coordinate system. Each voxel in the subset is represented by an index v(i, j, k), where $i \in [0;N_x\text{-}1]$, $j \in [0;N_y\text{-}1]$, and $k \in [0;N_z\text{-}1]$ (Figure 3). With the dimensions of individual voxels (Δx, Δy, Δz), a number of voxels (N_x,N_y, N_z) along each direction are given in Equations 1-3:

$$N_x = \frac{(x_{max} - x_{min})}{\Delta x} + 1 \quad (1)$$

$$N_y = \frac{(y_{max} - y_{min})}{\Delta y} + 1 \quad (2)$$

$$N_z = \frac{(z_{max} - z_{min})}{\Delta z} + 1 \quad (3)$$

The voxel has eight associated lattice vertices associated with six rectangular faces (Figure 3). Each interior voxel has 26 neighboring voxels with 8 sharing a vertex, 12 sharing an edge, and 6 sharing a face. Conversely, an exterior or interior voxel on a hole's boundary often has only 17 neighboring voxels (4 sharing a vertex, 8 sharing an edge, and 5 sharing a face). Moreover, most existing voxelization techniques operate on surface representations of

objects, where a significant part of the problem is to identify through which voxels the surfaces pass. Such methods are referred to as surface-based voxelization (e.g. Cohen-Or and Kaufman 1995) (Figure 4a to 4b to 4c). In contrast, the point-based voxelization in this paper operates directly on the point data and does not require a derived surface (Figure 4a to 4c). Point-based voxelization is conceptually much simpler than surface-based voxelization algorithms, and while the mechanisms are well-known, they have not been applied to generating solid modeling of buildings from LiDAR data.

As abovementioned, each voxel is classified as active or inactive corresponding to binary values based on the sample points within that voxel (Equation 4):

$$f_v(n) = \begin{cases} \text{active} & \text{if } n \geq T_n \\ \text{inactive} & \text{if } n < T_n \end{cases} \qquad (4)$$

where the argument n is the number of points mapping to a voxel, and T_n is a user-specified threshold value. Typically, $T_n = 1$, which means that voxels containing at least one mapping point are classified as active and all others as inactive. More sophisticated density-based classification functions can be designed. An example is shown in Figure 5.

PROPOSED CONVERSION OF VOXELIZED MODELS INTO SOLID MODELS

To reconstruct vertical surfaces of building models, a voxel grid is used to divide data points in a bounded 3D region into smaller voxels. Important façade features such as windows and doors are subsequently detected based on a voxel's characteristic, where an inactive voxel represents the inside of an opening. Consequently, building models are converted into an appropriate format for computational processing.

An object is defined by its surface boundary, which must then be converted into an appropriate solid representation compatible with commercial computational packages. Although many schemes are available, B-Reps are herein adopted because of their compatibility to commercial structural analysis software (e.g. ANSYS software) (Laefer et al. 2011). The proposed method defines both the geometry and topology of an object by a set of non-overlapping faces, which approximate the boundary of the solid model. This section presents a brief description of the B-Rep scheme implemented in the proposed approach; for more details see Goldman (2009). Geometry is defined by key (singular) points, with each point representing a specific location in space. Topology is defined by connections between key points. When used together they can define a solid model (Figure 6). Data structures for describing B-Reps often capture the incidence relations between a face and its bounding edges and an edge and its bounding vertices, which ensure topological consistency. The vertices are stored in a table with their associated coordinates. Edges are represented by reference to vertices and adjacent edges.

Key points are represented by a 3D coordinate of a singular point. An edge is defined as the connection between exactly two key points, for example, the edge $e_{ij} = \{P_i, P_j\}$ is the edge with starting point P_i and end point P_j. Notably, edges have an orientation; as such $e_{ij} = -e_{ji}$. Thus, the edges e_{ij} and e_{ji} would be flipped. Edge flipping is important when defining an orientable face for distinguishing the inside from the outside.

Similarly, faces represent surfaces of a solid model, which are connections between edges. The faces are further connected to form volumes. A face is defined as a list of edges $f = \{e_{01}, e_{12}, \ldots, e_{(n-2)(n-1)}\}$ that involves closed paths. A face comprised from three key points is a triangle, while quads have four key points (herein assumed to lie in the same plane). The number of key points defines the "order" of the polygon. Two faces are connected, if there is

an edge in the first face that exists in a flipped form in the second face. Again, this implies a consistent face orientation. Depending on the orientation of the face's edges, the normal direction (or the face orientation) is given by the right-hand rule (Figure 7).

Larger models are comprised of smaller volume elements. Each volume element is bounded by a set of faces. These faces must have consistent orientations, such that there is a clearly defined volume interior. The solid models created herein use cubic volume elements arranged in a regular grid. Using non-overlapping cubes, connected at the faces provides a simple, yet effective, means of creating solid models. Each cube has six faces, and adjacent cubes share faces. The faces are established so that normal directions are facing outward, away from the center of the cube.

EXPERIMENTAL TESTS AND DISCUSSION

The long-term goal of this research is to develop a pipeline of solutions to automatically transform city-scale data sets from aerial and terrestrial point clouds into solid models appropriate for engineering usage (Figure 1). The applicability of this process is shown with relevant data sets acquired in the city center of Dublin, Ireland. The grid voxel approach is herein implemented to reconstruct building models from the LiDAR data, in which a value of one sample point is predefined as the threshold for classifying active and inactive voxels in the subset voxels. An active voxel represents a portion of a solid wall, and an inactive one is either inside an opening or outside the façade.

The proposed method offers a new, data-driven approach to convert point cloud data into solid models (Figures 1c-d). However, as the building detection step (Figure 1b) is not currently in the automated portion of the pipeline, any input data clearly outside the plane of the façade (Figure 8a) should be removed either manually or semi-automatically within the

proprietary software accompanying the terrestrial scanner or via a segmentation technique for the case of the aerial data. These relatively distant points can contain disturbing elements that are irrelevant for the structural analysis of the façade. An example of the dataset without any cleaning is shown in Figure 8a and its processing in Figure 8b. Extraneous points represent floors, interior walls, advertisement panels, and stray floating voxels. See Appendix A for more information on the pre-processing.

Application to ALS Data

The aerial data was acquired according to the detailed procedures outlined in Hinks et al. (2009), which enabled a significantly improved capture of façade data compared to existing ALS methods. For that, ALS data of the building at 32 Westmoreland St., Dublin, Ireland was selected as input (Figure 9). The voxel size of 0.5 m was defined for reconstructing a building model (Figure 9c). The ALS data fully converged without any manual intervention, and the building model was able to be imported directly into the FEM program ANSYS (ANSYS Academic Research Release 13.0) for mesh generation (Figure 9d). Subsequently, a reasonable stress distribution of the building subjected to self-weight was achieved as shown in Figure 9e. Despite that relative success, the solid model and the subsequent FEM mesh are far from being geometrically accurate representations of the actual building (Figures 9d and e) because of the relatively low density of ALS sampling points.

Application to TLS Data

Ground-based TLS data, acquired with a Trimble GS200 3D Scanner (Trimble, 1999), was also used in experimental tests to investigate the influence of sampling point density on the quality of reconstructed building models. In the experimental tests, the datasets were manually cleaned of all data involving internal walls/objects and of all vehicles or trees in

front of the buildings, leaving only points acquired on the actual facades by using proprietary software, RealWorks Survey (RealWorksSurvey 2005) associated with the Trimble GS200 scanner (Truong-Hong 2011). The voxel size was experimentally set to either one-half or one-quarter of the minimum opening size, 0.4m, which was obtained from general building knowledge (Pu and Vosselman 2007). Figures 10-12 show the generation of building models from three sets of TLS point clouds using the two aforementioned voxel sizes; a detailed discussion of these buildings and their datasets is available in Laefer et al. (2011).

In order to evaluate the quality of reconstructed building models from the TLS data, the boundary lines of the actual building geometry were superimposed on the resulting solid models (Figures 10c-d, 11c-d and 12c-d). When using the TLS data (Figures 10-12) vastly superior results were obtained compared to the ALS data, although the geometries are still not fully reflective of the actual conditions. Additionally, the bounding box is often larger than the actual façade, while the reconstructed openings are usually smaller than expected (Figures 10c-d, 11c-d and 12c-d). Boundaries of the façade and its openings can match real ones when a small voxel size is used, but a few holes appear in the solid wall. As such, with 0.1 m as the voxel size, the detected boundary is significantly improved, but the number of holes increases, because in some areas the voxel size may be less than the distance between sample points (Figures 10d, 11d, and 12d).

Despite improvements, models reconstructed from TLS data still exhibit notable problems: (1) unexpected openings in the solid model due to occlusions (Figures 10c-d) or several holes in the solid wall due to the defined size of the voxel (Figures 10d, 11d, and 12d); (2) poorly matched geometries compared to actual openings (Figures 10c-d, 11c-d and 12c-d); (3) stray elements (as seen in several doorways in Figure 12c-d); and (4) fragment voxels (Figure

12d). This indicates that the remaining issues are not simply ones that can be overcome with additional data density.

DISCUSSION

To circumvent shadowing based problems for ALS data acquisition, flight path optimization has been proposed (Hinks et al. 2009). For TLS data, this problem is less acute due to the relatively small scale of most manmade objects (e.g. bus, telephone box, advertisement panel) and can be surmounted by scanning a façade at multiple times or from more than one location. This latter solution was adopted herein, where the scanner position was assigned a local coordinate system for each scan. The point clouds were registered and merged by using RealWorks Survey program associated with the Trimble scanner. This step transformed the point clouds from multiple local coordinate systems into a unique global coordinate system. Thus, a building model reconstructed from point clouds with multiple, original, local coordinate systems was still successful for computational modeling. This is notable as to date, although there are commercial solutions to convert point cloud data into voxels, none are compatible for solid model generation.

The proposed approach was implemented in MatLab script (MathWorks 2007) and run on a HP Compaq nx6320 Laptop with Intel (R) Core (TM) 2 CPU speed 2GHz with 2 Gb RAM. Data points with x-, y-, and z-coordinates were used for all experimental tests. The registration and other data preparation took approximately 5 minutes for each TLS dataset (Figure 10-12); the maximum running time for the algorithm was only 19.6 sec for a dataset with 32,980 voxels (Building 2 Westmoreland St. with 0.1 m of voxel size). Predictably, the processing time was proportional to the number of voxels in the model.

The time complexity of this algorithm depended primarily on the cost of setting up and

maintaining the voxelization, and on the cost of processing each sample point. Given a voxelization with V = 256 x 256 x 256 = 16,777,216 voxels, the storage and setup cost would be O(V) for an array implementation with access cost of O(1) per voxel. Moreover, since the voxelization is a regular array, the adjacencies to be stored are linear in the number of voxels, again at a cost of O(V). When each of the N samples is processed, it is optionally stored in the corresponding voxel at an O(1) cost. The adjacencies of the voxel can be updated directly at this stage, or extracted separately in a post-processing step. Given O(1) access cost and a bounded number of adjacencies, this step is O(N) if performed immediately, or O(V) if performed during post-processing. The overall complexity of the algorithm is, therefore, O(V+N). Hence, the time cost is acceptably small, and no further optimization is required at present. The computation is more likely to be dominated by N than by V for large data sets, since a voxel with only a few sample points is only questionably solid for the purpose of FEM modeling. Thus, for practical purposes this algorithm is linear with the number of sample points.

There are, however, two critical parameters that affect the quality of building models: sample point threshold and voxel size (for TLS data) or subdivision along each axis (for ALS data). The building models reconstructed from ALS data were sensitive to the sample point threshold, because the data were irregular and sparse. A higher threshold number increased the quantity of holes and floating voxels when the voxel size was held constant (Figure 13 for a 0.4m voxel size). An improperly sized voxel is also problematic, as an unduly small voxel may increase the number of unexpected holes, because the size may be less than the distance between two adjacent points (Figure 14). Conversely, an overly large voxel can fail to detect openings. For example, in a dataset with an opening size of 0.4 m, if the voxel size is also set at 0.4 m, it is possible that no opening would be detected. The problem is overcome by setting the voxel size to 0.2 m. The initial values in this study were selected based on the

evaluation of 400 urban facades showing window widths 0.3-1.9 m and heights 0.5-2.4 m (Ripperda, 2008).

Ultimately, selection of these factors should be automated. So while the fundamental processes are shown to be sound with both ALS and TLS data, pre-processing steps to segregate the relevant data and feature detection algorithm are still both clearly needed, as are overcoming the complexities of "full 3D" modeling of a building with multiple surfaces. These items must be considered under future work.

Also of note is that in this implementation, there is no further feature detection approaches currently integrated, which leads to holes due to occlusions to be treated as real openings (doors and windows). This drawback can be solved by implementing a subsequent feature-detection step to distinguish whether or not a hole reflects a realistic feature. Furthermore, since a cubic voxel is used, the proposed algorithm is limited to rectilinear facades and openings. Thus the existence of wedged or arched windows is not explicitly depicted. However, this approximation generates only limited divergences in computational modeling results (Truong-Hong 2011). Conversely, the voxel format has further benefits. As noted by Mosa et al. (2012), a voxel-based representation of LiDAR data has the additional advantage of immediate compatibility with an octree storage format, which would support the simultaneous storage of semantic data, such as the red-green-blue color information typically collected during laser scans.

CONCLUSIONS

A voxelization-based method has been proposed to transform point cloud data from aerial and terrestrial laser scanning data of building façades into solid models for computational modeling. The proposed method is fast and offers a high degree of automation in

reconstructing building models from LiDAR data and the resulting solid models are fully compatible for importation into commercial FEM programs for computational mesh generation, although only for two-dimensional representation. The approach is nearly automatic, as it requires only the pre-selection of the number of grids along each direction (for ALS data) or the selection of a voxel size (for TLS data). Experimental tests on one aerial and three terrestrial datasets for different buildings in the city center of Dublin, Ireland illustrate both the potential advantages and the remaining shortcomings of the approach. Specifically, further work was identified as being needed for (1) better shape detection of openings, (2) gaining the ability to distinguish real openings from data occlusions, and (3) increasing the level of automation, particularly related to initial data selection and segregation, as well as its expansion into full three-dimensional modeling.

ACKNOWLEDGMENTS

This work was generously supported by Science Foundation Ireland Grant 05/PICA/I830 and Ireland's Environmental Protection Agency Grant 2005-CD-U1-M1. Thanks to Donal Lennon of Urban Institute Ireland for assistance with terrestrial data acquisition.

APPENDIX A

The original scanned data sets from multiple scanner stations were registered and then primarily cleaned of irrelevant points using the RealWorks Survey program associated with the Trimble scanner. The processing was as follows:

First, point clouds were registered by selecting at least three reference points on the source and target datasets for merging the point clouds from multiple scanner stations (Figure A1), in which shape geometries of the facade (e.g. window corners or ledges) were selected as the

reference points instead of using special targets or nails. Then, the in-built segmentation technique within the RealWorks Survey program clusters points that are on the same plane (Figure A2). That step supports the efficient, automatic removal of irrelevant portions of the point cloud (e.g. internal walls/objects or advertisement panel). The processing time for merging and cleaning was typically less than 5 minutes for each dataset presented herein. However, the dataset may contain a few irrelevant points close to the façade's vertical surface. These points were removed by using a Matlab script based on a histogram, where the data points exporting from the RealWorks Survey program are used to generate the histogram along to z-axis (depth direction of the building). Notably, x- and y-coordinates are respectively parallel to width and height directions of the façade, while the z-coordinate is parallel to a depth direction. As such, the data points of the facade belong to the peak of the histogram and the process was removed all points out of the interval ±20 cm from the facade surface. Furthermore, there are several approaches that can be used to segment point clouds instead using the commercial software. For example, with TLS data Pu and Vosselman (2011) used the planar, surface-growing algorithm, while Martinez et al. (2012) choose the RANSAC algorithm to extract geometric features.

REFERENCES

ANSYS Academic Research Release 13.0. "Help System, Theory Reference for ANSYS and ANSYS Workbench."

Böhm, W., Farin, G., and Kahmann, J. (1984). "A Survey of Curve and Surface Methods in CAGD." Journal of Computer Aided Geometric Design, 1(1), 1-60.

Chen, J. and Chen, B. (2008). "Architectural Modeling from Sparsely Scanned Range Data." Int. J. of Comput. Vision, 78(2-3), 223-236.

Cohen, D., and Kaufman, A. (1990). "Scan-Conversion Algorithms for Linear and Quadratic Objects." Volume Visualization, A. Kaufman, Ed., IEEE Computer Society Press, Los Alamitos, CA, 280-301.

Cohen-Or, D., and Kaufman, A. (1995). "Fundamentals of Surface Voxelization." Graphical Models and Image Processing, 57(6), 453–461.

Curless, B., and Levoy, M. (1996). "A Volumetric Method for Building Complex Models from Range Images." Proceeding of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, August 4-9, 1996, 303-312.

Dong, H., Easa, S.M., and Li, J. (2007). "Approximate Extraction of Spiralled Horizontal Curves from Satellite Imagery." J. Surv. Eng. 133(1), 36-40.

Dorninger, P., and Pfeifer, N. (2008). "A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds." Sensors, 8(11), 7323-7343.

Goldman, R. (2009). An Integrated Introduction to Computer Graphics and Geometric Modeling, CRC Press, Taylor and Francis, New York.

Guarneri, A., and Pontin, M. (2005). "A volumetric approach for 3D surface reconstruction." CIPA 2005 XX International Symposium, Torino, Italy, September 26-October 1, 2005, 831-836.

Haala, N., Brenner, C., and Anders, K.h. (1998). "3D Urban GIS From Laser Altimeter and 2D Map Data." International Archives of Photogrammetry & Remote Sensing, 32(3), 339-346.

Haala, N., and Kada, M. (2010). "An Update on Automatic 3D Building Reconstruction." ISPRS J. Photogramm. Remote Sens, 65(6), 570-580.

Henricsson, O., Bignone, F., Willuhn, W., Ade, F., Kubler, O., Baltsavias, E., Mason, S. and Grun, A.(1996). "Project Amobe: Strategies, Current Status, and Future Work." In International Archives of Photogrammetry and Remote Sensing, XXXI(Part B3), 321-330.

Hinks, T., 2011. Geometric processing techniques for urban aerial laser scan data. University College Dublin, PhD thesis.

Hinks, T., Carr, H., and Laefer, D.F. (2009). "Flight Optimization Algorithms for Aerial LiDAR Capture for Urban Infrastructure Model Generation." J. Comput. Civil Eng, ASCE, 23(4), 330-339.

Hoffmann, C.M. and Rossignac, J.R. (1996). "A Road Map to Solid Modeling." IEEE Trans on Visualization and Computer Graphics, 2(1), 3-10.

Hofmann, R. D., Maas, H.-G., and Streilein, A. (2003). "Derivation of Roof Types by Cluster Analysis in Parameter Spaces of Airborne Laserscanner Point Clouds." ISPRS Commission III WG3, Workshop, 3D Reconstruction from Airborne Laser Scanner and InSAR Data, Dresden, Germany, 112-117.

Hoppe, H. (1994). "Surface Reconstruction from Unorganized Points," PhD thesis, University of Washington.

Hu, J., You, S., Neumann, U., Park, K.K. (2004). "Building Modeling from LiDAR and Aerial Imagery." ASPRS 2004, Denver, Colorado, USA, May 23-28, 2004, pp. 6.

Karabassi, E.-A., Papaioannou, G., and Theoharis, T. (1999). "A Fast Depth-Buffer-Based Voxelization Algorithm." Journal of Graphics Tools, 4(4), 5-10.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). "Poisson Surface Reconstruction." Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006, 61-70.

Laefer, D.F, and Pradhan, A. (2006). “Evacuation Route Selection Based on Tree-Based Hazards Using LiDAR and GIS” J. Transportation Eng., 132(4), 312-20.

Laefer, D.F., Truong-Hong, L., and Fitzgerald, M. (2011). "Processing of Terrestrial Laser Scanning Point Cloud Data for Computational Modelling of Building Facades." Recent Patents on Computer Science, 4(1), 16-29.

Lang, F., and Forstner, W. (1996). "3D-City Modeling with a Digital One-eye Stereo System." International Archives of Photogrammetry and Remote Sensing, Vol. XXXI (B4), 261-266.

Lorensen, W. E., and Cline, H. E. (1987). "Marching Cubes: A High Resolution 3D Surface Construction Algorithm." Compututer Graphics, 21(4), 163-169.

Maas, H. and G. Vosselman (1999). "Two Algorithms for Extracting Building Models from Raw Laser Altimetry Data." ISPRS J. Photogramm Remote Sens. 54(2-3), 153-163.

Martinez, J., Soria-Medina, A., Arias, P., and Buffara-Antunes, A.F. (2012). Automatic Processing of Terrestrial Laser Scanning Data of Building Façade. Automation in Construction, 22, 298-305.

MathWorks (2007). "MATLAB Function Reference." Release 2007a.

Mosa, A.S.M., Schön, B., Bertolotto, M., Laefer, D. (2012). Evaluating the Benefits of Octree-based Indexing for LiDAR Data. PE&RS, 78(9), 1-8.

Mukherji, B. (2011). “Estimating 3-D Volume Using Finite Elements with an Application to Pit Excavation.” J. Surv. Eng., 138(2), 85-91.

Olsen, M.J., Johnstone, E. Kuester, F. Driscoll, N. and Ashford, S.A. (2011). “New Automated Point-Cloud Alignment for Ground-Based Light Detection and Ranging Data of Long Coastal Sections.” J. Surv. Eng., 137(1), 14-25.

Olsen, M.J., Johnstone, E. Driscoll, N., Ashford, S.A., and Kuester, F. (2009) “Terrestrial Laser Scanning of Extended Cliff Sections in Dynamic Environments: Parameter Analysis.” J. Surv. Eng., 135(4), 161-169.

Parrish, C.E. and Nowak, R.D. (2009) “Improved Approach to LIDAR Airport Obstruction Surveying Using Full-Waveform Data.” J. Surv. Eng., 135(2), 72-82.

Pu, S., and Vosselman, G. (2007). "Extracting Windows from Terrestrial Laser Scanning." ISPRS Workshop on Laser Scanning and SilviLaser 2007, Espoo, Finland, September 12-14, 2007, 320-325.

Pu, S., and Vosselman, G. (2011). Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 64(6), 575–584.

RealWorks Survey, 2005. Technical notes-RealWorks Survey, http://www.trimble.com. (Accessed 26 November, 2011).

Ripperda, N. (2008). "Determination of Facade Attributes for Facade Reconstruction." The International Archives of the Photogrammetry, Remote and Spatial Information Sciences. Beijing, China, July 3-11, 2008, 285-290.

Rossignac, J.R., and Requicha, A.A.G. (1984). "Constant-Radius Blending in Solid Modeling." ASME Comput. Mechanic. Eng., 3(1), 65-73.

Rossignac, J.R., and Requicha, A.A.G. (1999). "Encyclopedia of Electrical and Electronics Engineering." Solid Modeling, J. G. Webster, ed., John Wiley and Sons.

Samet, H. (1989). Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS, Addison-Wesley Pub.

Shan, J. and Lee, S.D. (2005). “Quality of Building Extraction from IKONOS Imagery.” J. Surv. Eng. 131(1), 27-32

Stewart, J.P., Hu, J., Kayen, R.E., Lembo, A.J., Collins, B.D., Davis, C.A. and O'Rourke, T.D., (2009) “Use of Airborne and Terrestrial Lidar to Detect Ground Displacement Hazards to Water Systems.” J. Surv. Eng. 135(3), 113-125.

Trimble (1999). Trimble GS200 3D Scanner. <http://www.trimble.com/gs200.shtml> (Sep 1st, 2011)

Truong-Hong, L. (2011). Automatic Generation of Solid Models of Building Facades from LiDAR Data for Computational Modelling. University College Dublin, PhD thesis.

Vosselman, G. and Dijkman, S. (2001). " 3D Building Model Reconstruction from Point Clouds and Ground Plans." International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4, Annapolis, MD, October 22-24, 2001, 37-43.

Wheeler, M. D., Sato, Y., and Ikeuchi, K. (1998). "Consensus Surfaces for Modeling 3D Objects from Multiple Range Images." Proceedings of the Sixth International Conference on Computer Vision, IEEE Computer Society, Bombay, India, January 4-7, 1998, 917-924.

You, S., Hu, J., Neumann, U., Fox, P., (2003). Urban Site Modeling from LiDAR. Proceedings of the 2003 International Conference on Computational Science and Its Applications, Montreal, Canada, May 18-21, 2003, 579-588.

Zhang, D., Huang, T., Li, G., and Jiang, M. (2011). "A Robust Algorithm for Registration of Building Point Clouds Using Planar Patches." J. Surv. Eng., 138(1), 31-37.

Zhou, Q.-Y., and Neumann, U. (2010). "2.5D Dual Contouring: A Robust Approach to Creating Building Models from Aerial LiDAR Point Clouds." 11th European Conference on Computer Vision (ECCV 2010), Crete, Greece, September 5-11, 2010, 1-14.

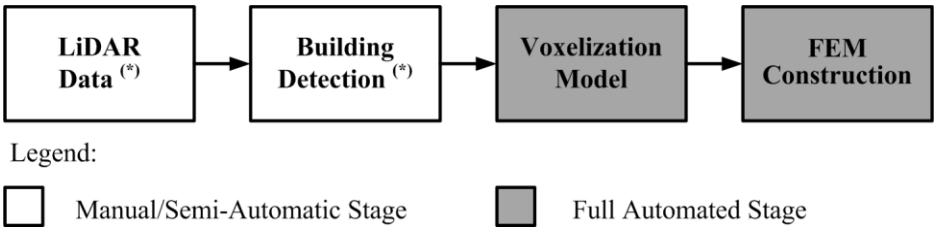


Figure 1. Workflow of the proposed approach

(*)Note: Collection and preparation of LiDAR data involves multiple steps outside the scope of this paper's scientific contribution. These generally include planning, collection, registration and filtering; see Truong-Hong (2011) and Hinks (2011) for further details.

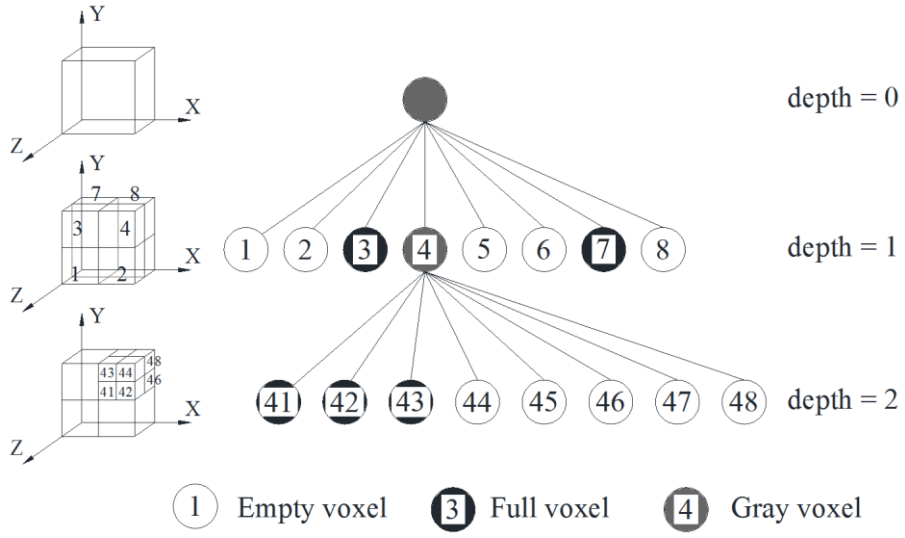


Figure 2. Octree representation

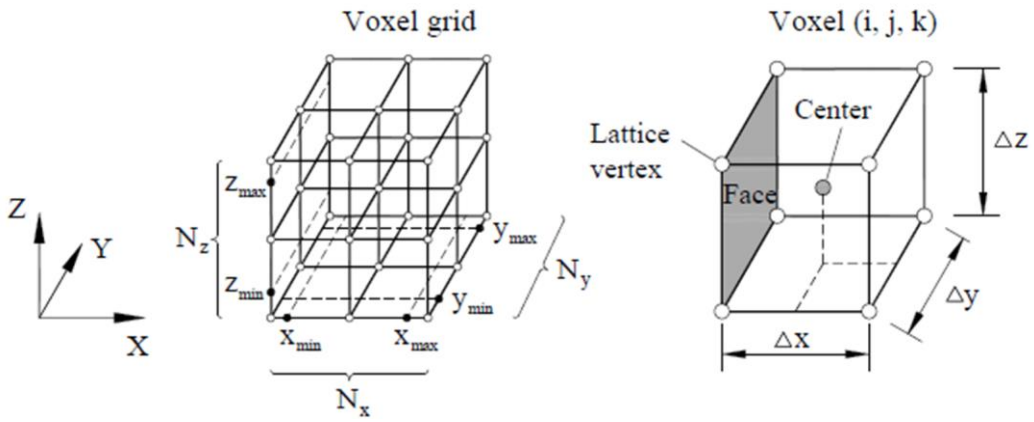


Figure 3. Voxel grid spanning a volume in a 3D space bounded by (x_{min}, x_{max}), (y_{min}, y_{max}), and (z_{min}, z_{max})

Note:

- (Δx, Δy, Δz) are voxel size

- (N_x, N_y, N_z) are the number of voxels in each direction

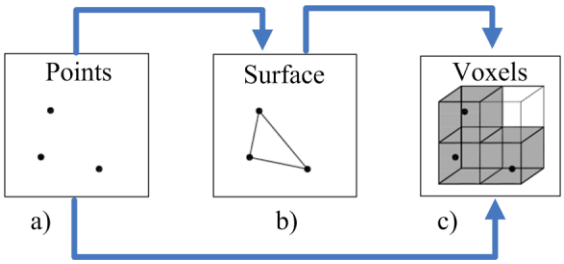


Figure 4. Point-based voxelization avoids surface reconstruction and operates directly on point data

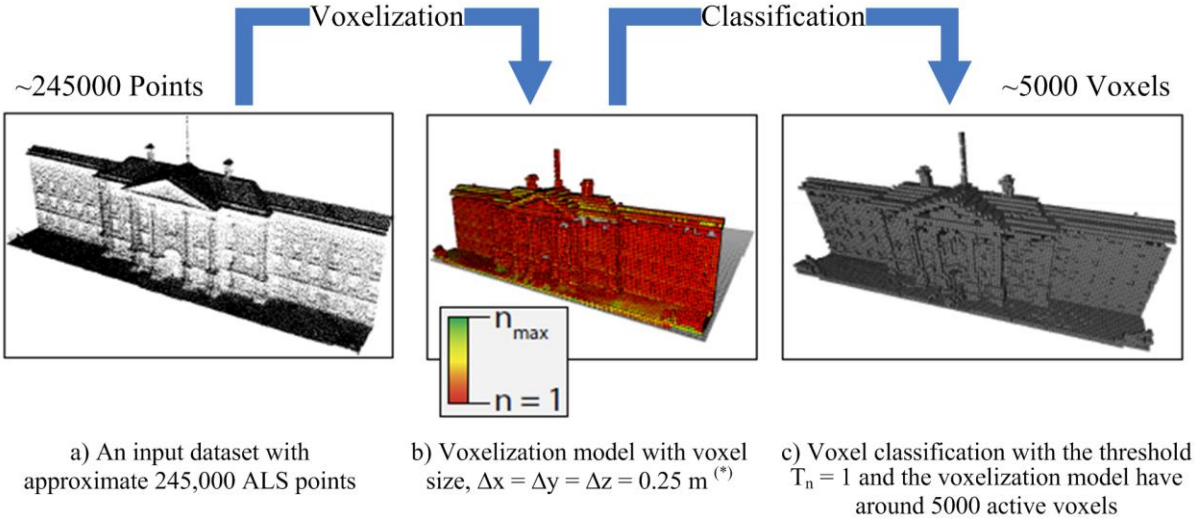


Figure 5. A voxelization model of the front building of Trinity College Dublin, Ireland, created by a voxel grid

Note: $^{(*)}n_{max}$ is the largest number of points mapping to a single voxel

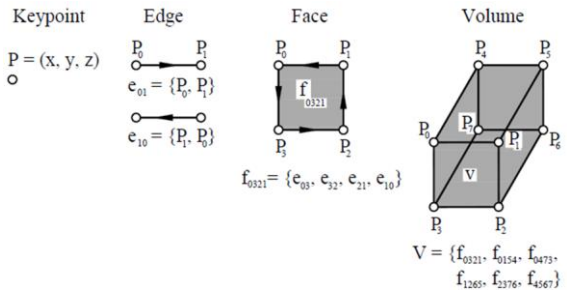


Figure 6. Solid model components

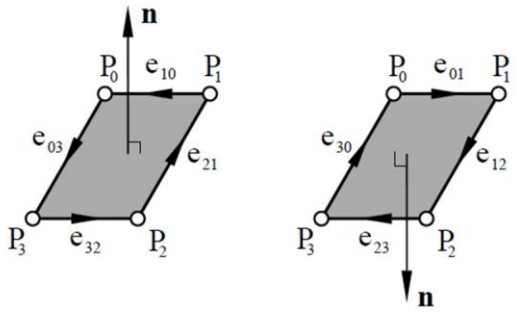
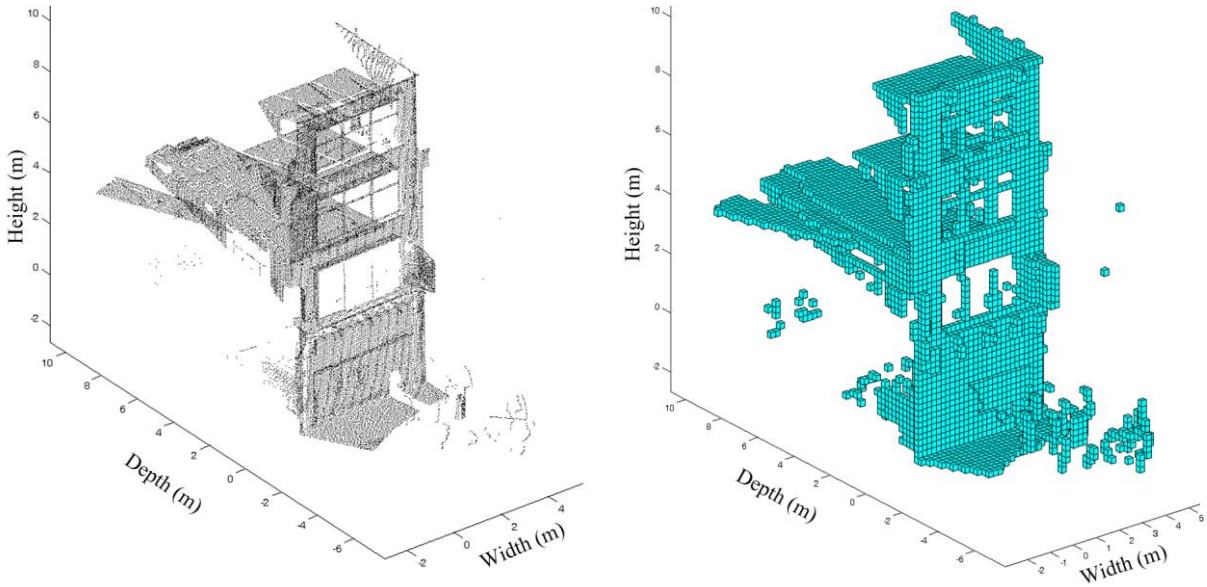


Figure 7. Face orientation as dictated by the right-hand rule

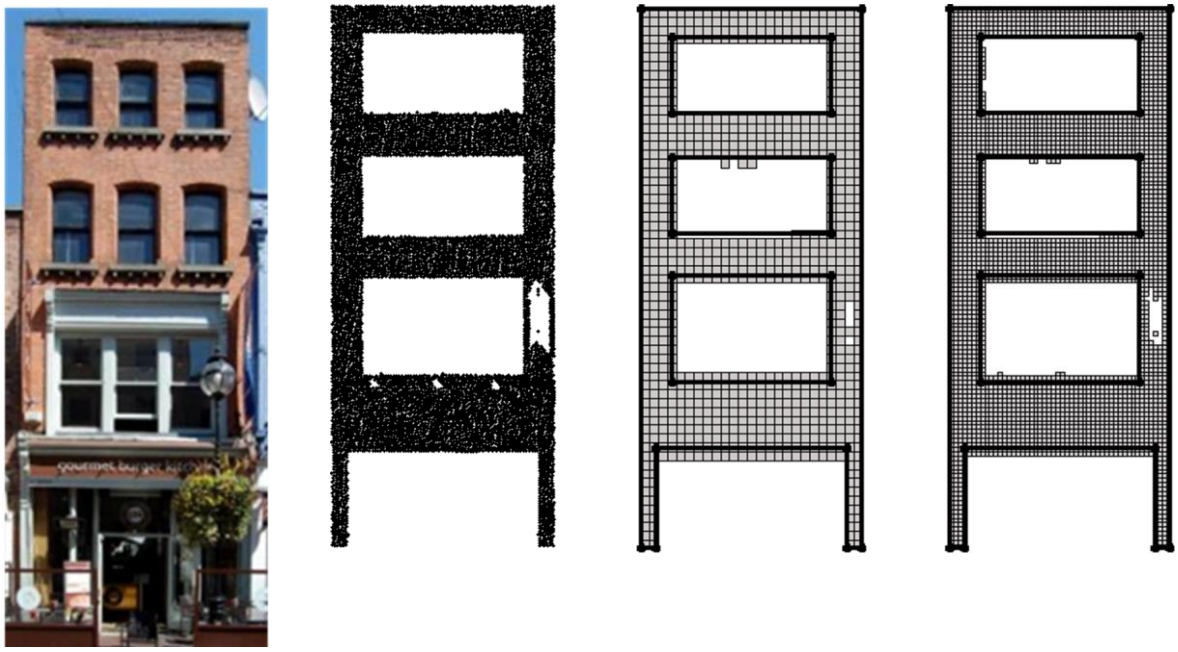


a. 3D point clouds b. Voxelisation model

Figure 8. A building model reconstructed from 3D point clouds without cleaning irrelevant points

a. Photo of Building b. LiDAR data c. Voxelization model d. Solid model e. FEM results

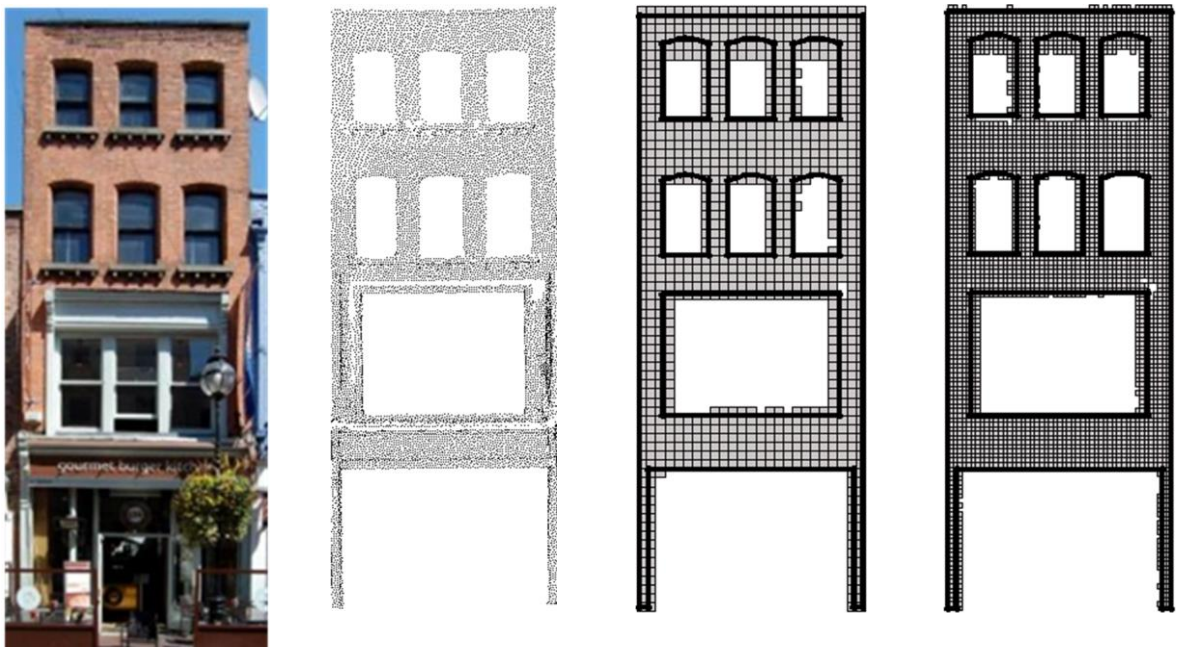
Figure 9. Solid model reconstruction of 32 Westmoreland from ALS data



a. Photo of Building 1 (4.95m length x 12.16m height) | b. Point cloud after cleaning (9000 pts/m^2) | c. Solid model representation with voxel size = 0.2m (*) | d. Solid model representation with voxel size = 0.1m (*)

Figure 10. Solid model reconstruction of 2 Anne St. South from TLS data

* Note: Superimposed boundary lines of the façade and its openings are from a manual survey of the selected building

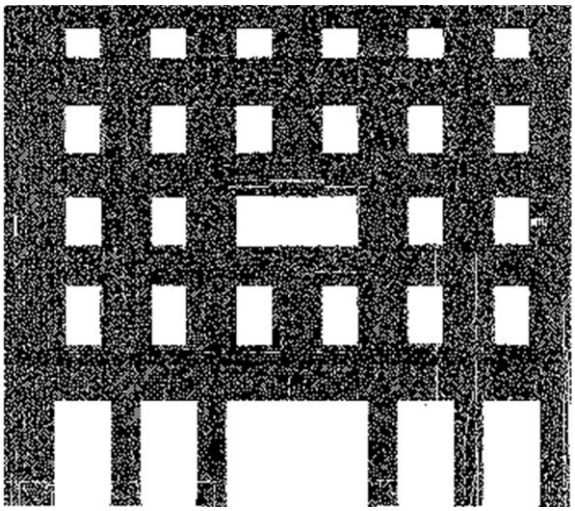


a. Photo of Building 2 (4.9m length x 13.28m height) | b. Point cloud after cleaning (400 pts/m^2) | c. Solid model representation with voxel size = 0.2m (*) | d. Solid model representation with voxel size = 0.1m (*)

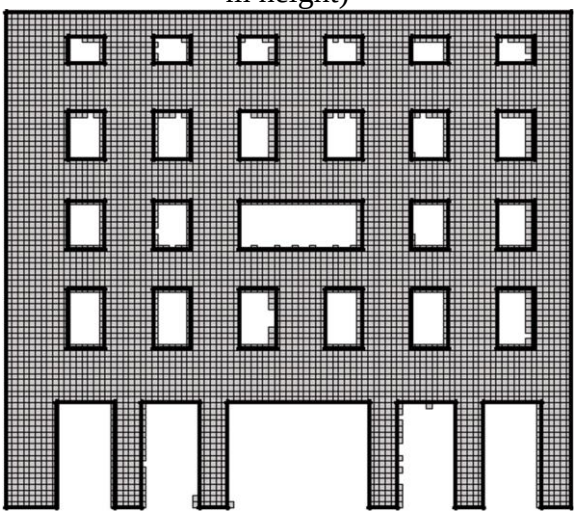
Figure 11. Solid model reconstruction of 5 Anne St. South from TLS data

* Note: Superimposed boundary lines of the façade and its openings are from a manual survey of the selected building

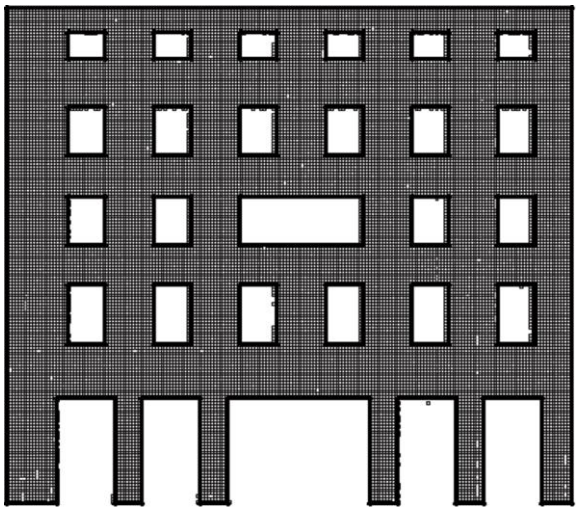
a. Photo of Building 2 (19.36 m length x 17.0 m height)



b. Point cloud after cleaning (175 pts/m^2)



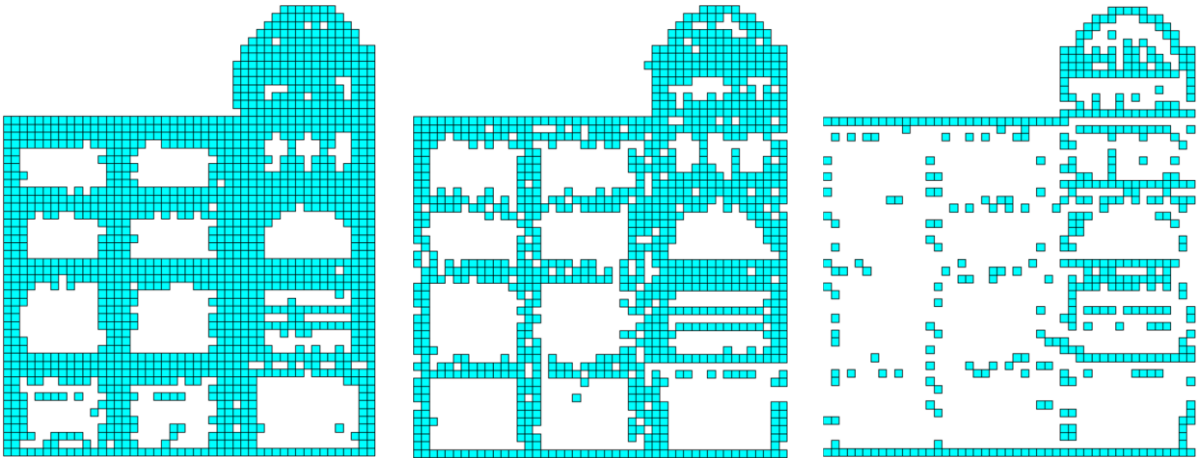
c. Solid model representation with voxel size = 0.2m (*)



d. Solid model representation with voxel size = 0.1m (*)

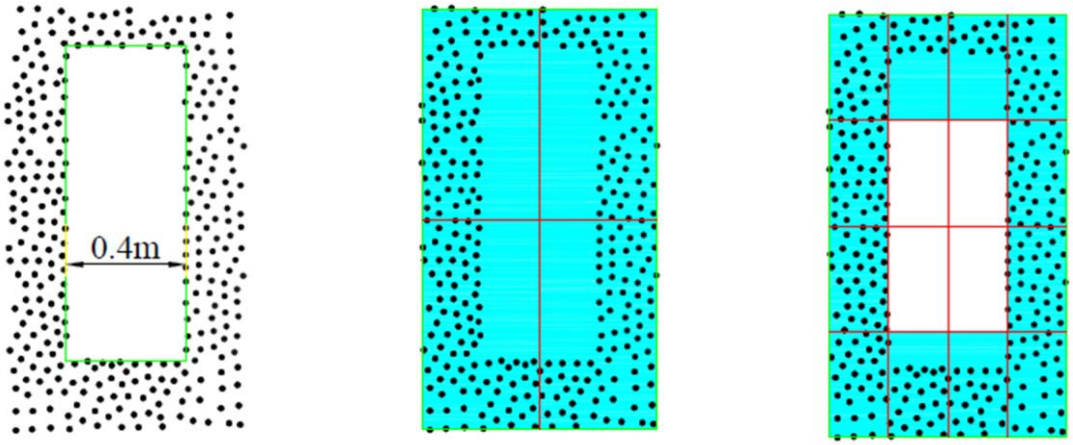
Figure 12. Solid model reconstruction of 2 Westmoreland St. from TLS data

* Note: Superimposed boundary lines of the façade and its openings are from a manual survey of the selected building



a. Point threshold = 1 b. Point threshold = 3 c. Point threshold = 6

Figure 13. Voxelization models with various thresholds

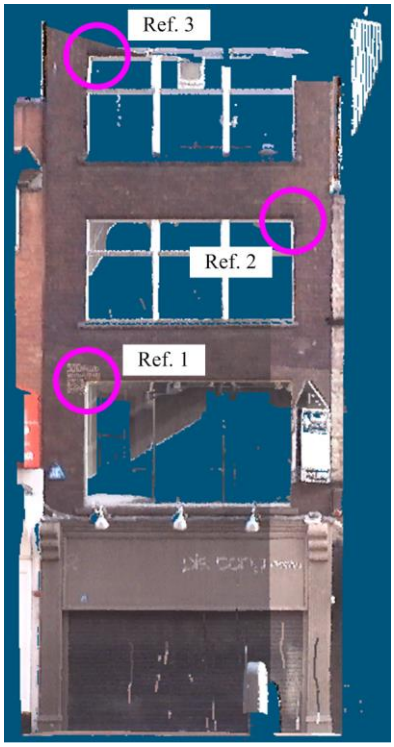


a. Expected opening in TLS data shown in green lines

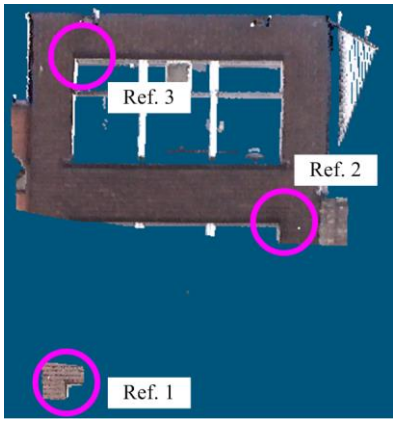
b. Voxelisation model with 0.4m of voxel size

c. Voxelisation model with 0.2m of voxel size

Figure 14. Voxelization models with various defined voxel sizes



a. Point clouds of station 1



b. Point clouds of station 2

Figure A1. Registering and merging point clouds of Building 1 from two scanner stations

a. Full point clouds of Building 1

b. Cleaned of irrelevant sample points (*)

Figure A2. Illustration of merging and cleaning irrelevant sample points of Building 1

* Pink regions represent irrelevant points