FLYING VOXEL METHOD WITH DELAUNAY TRIANGULATION CRITERION FOR FAÇADE/FEATURE DETECTION FOR COMPUTATION

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

ABSTRACT: In an effort to reconstruct geometric models of building façades from terrestrial laser scanning data directly without either manual intervention or any third party computer-aid design package, a new algorithm is introduced. The algorithm detects building boundaries and features and converts the point cloud data into a solid model appropriate for computational modeling. The algorithm combines a voxel-based technique with a Delaunay triangulation based criterion. In the first phase, the algorithm detects façade boundary points from raw data. The algorithm's second phase creates a solid model using voxels in a quadtree representation. Finally, the algorithm determines whether holes are actual openings or data deficits caused by occlusions and then fills unrealistic openings. The algorithm was applied to the façades of three masonry buildings. For these buildings, the algorithm successfully detected all openings and reconstructs the façade details correctly. Geometric validation of the models against measured drawings showed overall dimensions correct to 1.2%, mostly opening areas to 3%, and simulation results within 5% of those predicted by a CAD-based model.

[(1)] Doctoral candidate, Urban Modelling Group (UMG), School of Civil, Structural, and Environmental Engineering (SCSEE), University College Dublin (UCD), Newstead G67, Belfield, Dublin 4, Ireland. Email: linh.truong-hong@ucdconnect.ie

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

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

[(4)] Senior Lecturer, School of Computing, Faculty of Engineering, University of Leeds, E C Stoner Building 9.25, UK. Email: h.carr@leeds.ac.uk

KEYWORDS: Terrestrial laser scanning, LiDAR, voxelization, octree, Delaunay triangulation, façade boundaries, feature detection, flying voxel, point cloud data

INTRODUCTION

Three-dimensional (3D) point clouds from laser scanning are increasingly used in science and engineering (Brilakis et al. 2011; Collins and Sitar 2004; Tang et al. 2009a). In Civil Engineering, modeling of buildings is of particular interest. Laser scanned data of building facades have been transformed successfully into surface models for visualization by detecting features such as windows and dormers and generating polygonal outlines. However, these outlines are incompatible with computational modeling for several reasons: (1) the resulting building models contain distorted surfaces that cause unrealistic finite element meshes (FEM) or degenerate shapes that cause difficulties in generating convergent meshes; (2) the algorithms depend heavily on user experience or supplemental data sets (e.g. photographs); or (3) the algorithms are unable to overcome sparse and missing data, resulting in inaccuracies.

In this paper, the FaçadeDelaunay algorithm is introduced to overcome these issues in the automatic reconstruction of computational models of building façades from laser scanning data. The technique focuses on two-dimensional (2D) modeling of brick buildings in an attempt to generate usable geometries of façades and their openings and to correct holes due to occlusions or other missing data.

The discussion will start by surveying existing work on boundary detection and the use of volumetric approaches for geometric modeling. Thereafter, the details of the FaçadeDelaunay algorithm are introduced, starting with Delauanay triangulation of meshes, as adopted by Pu and

Vosselman (2007). The Delauanay triangulation is then supplemented by automatic detection and correction of unrealistic holes due to missing or occluded data. The final parts of the algorithm are based on an inclusion criterion called the "flying voxel method" and an octree representation to generate the final solid model. The algorithm is then evaluated for efficiency and reliability by processing TLS data at four sampling densities for each of three building façades. The viability of the method is then confirmed by computational modeling from the reconstructed solid models and comparisons with empirical studies of excavation-induced, ground movements. Finally, conclusions and future work are presented.

RELATED WORKS

For building façades, terrestrial laser scanning (TLS) data have generally been preferred over airborne data because of greater data density, which is critical for feature extraction. Ripperda and Brenner (2009) categorized methods for extracting geometric features and reconstructing building facades from such data sets as top-down or bottom-up: top-down approaches first design models without the raw data (Wonka et al. 2003), while bottom-up (e.g. fitting geometric primitives and meshing-based) methods are data-driven (Becker and Haala 2007, 2009; Pu and Vosselman 2007, 2009). Many of these approaches rely on surface reconstruction: much of the extensive literature in this area has recently been summarized by Laefer et al. (2011). The following therefore focuses primarily on methods for boundary detection through Delaunay triangulation or derived techniques and methods for geometric building modeling.

Using Varnuška and Kolingerová's (2004) concept of an adaptive criterion based on the edge length and the angle between point and incident triangles to detect points on a boundary, Pu and Vosselman (2007) identified holes by searching for the long edges from various triangulated

irregular networks (TINs) to extract windows. The end points of TIN edges were classified as boundary points, if a length edge exceeded a specified threshold (as done by Tang et al. 2007). Sample points on boundaries of features were categorized into upper, lower, left and right groups. Subsequently, Pu and Vosselman (2009) generated a façade's upper boundary line from contour points by least-square fitting, then generated left and right boundaries by projecting the extreme vertices of the upper line to the ground plan. Additionally, a minimum bounding rectangle was fitted to each window in the façade. Building on this, Boulaassal et al. (2009) applied RANdom Sample Consensus (RANSAC) to automate planar part segmentation and extraction to which a 2D Delaunay triangulation was applied to extract contour boundary points of openings. Those boundary points were also classified directly into four aforementioned groups and then transformed into parametric objects. This approach can extract sufficient boundary points to generate outline polygons of major features, but is highly sensitive to a user predefined length threshold and generates varying levels of geometric accuracy. In related work to detect contours of a bounding rectangular window, Ali et al. (2008) introduced adaptive thresholds based on descriptive statistics and an image-based method established from local absolute differences of adjacent laser-measured distances and morphological operations to detect contours of a bounding rectangular window (Wang et al. 2011). As some window appears as holes in the facade, because the laser scanner does not return a signal (pulse) from highly reflective material, potential boundary points can be detected by examining neighboring spaces, defined as voxels along vertical and horizontal directions. The voxel is designated as having a boundary point, if at least one empty voxel appears in an interval width of the considered voxel. The methods can efficiently detect openings but with relatively low geometric accuracy and without the ability to distinguish windows from holes due to data occlusions.

Alternatively, detailed façade geometry can be generated based on geometric similarity or repetitive patterns of façade features (Mayer and Reznik 2005; Wonka et al. 2003). Several researchers have applied grammar based methods to split façade images into many regular regions according to differences in façade structure and similar or repeated rectangular shapes for windows and doors (Becker and Haala 2009; Ripperda and Brenner 2009).

Using inherent geometry, the generation of the convex hull from a finite point set leads to a family of straight-line graphs, in which the given points are end points of those lines. Intuitively, cavities can be created within the convex hull if any line in the graphs shorter than a fixed length value was removed. This is the primary idea of the α-shape, which is widely used in computer graphics to detect boundary points (Edelsbrunner and Mücke 1994). Edelsbrunner and Mücke (1994) decomposed sampling points into a Delaunay tetrahedrization and a Voronoi diagram, where all vertices of the tetrahedra were sampling points. If the minimum surrounding sphere (called the α-ball) failed to fit this tetrahedrization, then the tetrahedra, triangles, and edges the tetrahedrization were removed from the mesh. However, due to variable point density, a global radius may cause loss of object details. For example, if a global parameter is too small, the reconstructed surface can have gaps or be fragmented. This drawback can be overcome by using local scale parameter for reconstructing the local geometry. Similarly, Bernardini et al. (1999) proposed a ball-pivoting algorithm dictating that the ρ-ball (ball of radius ρ) cannot pass through the surface without touching sample points, for which the ρ-ball contacts only three sample points. Surface normals computed from range maps are added to data points to overcome missing or noisy data. Radii are selected based on sampling density and feature size. The selection of an appropriate radius is close to the definition of the length threshold for Delaunay triangulation mesh for detecting boundary points.

Since the second part of the approach proposed later presented in this paper as the research contribution empnloys a octree representation to reconstruct the façade model based on boundary points of features, related work on octree or other volumetric approaches are surveyed herein. In volumetric approaches, objects' surfaces are mostly reconstructed based on a signal function distance or oriented-charged approach. Curless and Levoy (1996) described a new algorithm for volumetric integration of range images to reconstruct an object's surface based on a cumulative weighted signed distance function. This method, however, cannot generate models for an arbitrary object, may not detect features smaller than the grid spacing, and requires significant memory and execution time. To overcome memory and speed problems, Pulli et al. (Pulli et al. 1997) used an octree representation to create meshes from multiple range maps, where the initial cubical volume was recursively subdivided into eight smaller cubes until reaching a predefined sub-division depth, and each voxel was classified as "inside", "boundary", or "outside", with respect to its location to the sensor and range data. Subsequently, a triangular mesh was created at surfaces shared between outside cubes and other ones.

In related work Guarnieri et al. (2005) used a consensus surface proposed by Wheeler et al. (1998), an octree representation and the marching cubes algorithm (Lorensen and Cline 1987) to build a triangulated mesh. Although the results were good, even with noisy data sets, the algorithm is not automatic and requires adjusting parameters for each data set. Similarly, Wang et al. (2005) used oriented charges to compute distance fields only employing input point positions, and then a zeros-set surface was determined from the distance fields. The algorithm is applicable to clean and noisy datasets and hole filling, but surface features smaller than the smallest octree size may go undetected. Additionally, Dalmasso and Nerino (2004) described objects at different scales of spatial resolution based on an octree structure. Efficient and fast

determination of the zeros level were obtained by a choice of local compact support radial basic functions. The approach is suitable for the surface reconstruction from multiple view images.

Four unresolved problems emerge from this assessment of existing work: inaccurate meshes, convergence issues; reliance on user experience or supplemental data; and an inability to overcome sparse and missing data. An algorithm that addresses these is introduced below.

PROPOSED WORKFLOW FOR RECONSTRUCTION OF SOLID MODELS

In addition to a new algorithm, a corresponding workflow must also be introduced. This workflow, shown in Fig.1 has two major stages: (i) feature detection and (ii) solid model reconstruction. This workflow is based on the assumption that that most buildings are quadrilateral in shape, with the structural elements residing within a planar facade, with primarily rectangular windows and glass-plated doors. Architectural details and non-structural elements (e.g. balconies and window ledges) are ignored. The algorithm, herein referred to as FacadeDelaunay (FD), currently reconstructs 2D façades, but could be extended to 3D representations.
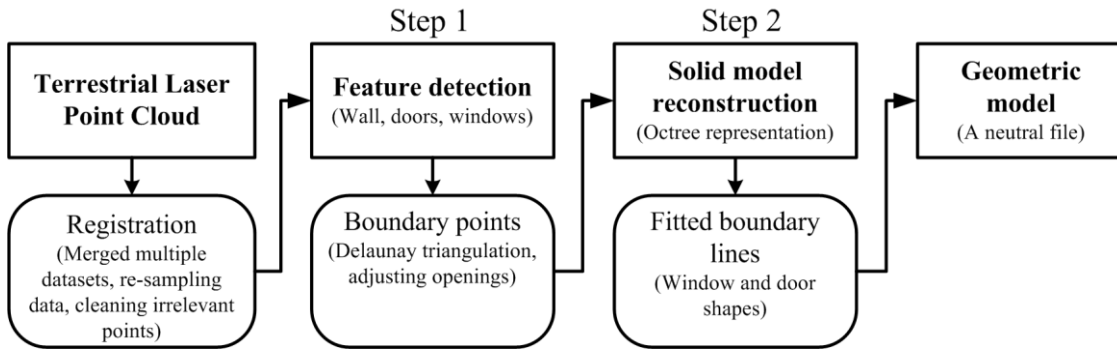
Figure 1. Building reconstruction process

The approach differs from Pu and Vosselman (2007, 2009) and Boulaassal et al. (2010), in that façade feature boundaries are determined from boundary points within each voxel on the façade feature's boundary, where some inaccurate boundary points can be also eliminated by verifying voxel grids.

Feature detection (Step 1)

The algorithm's feature detection involves initial boundary point determination followed by clustering whole boundary points on the same hole. Then unrealistic holes are eliminated, and the classification of those previously characterized boundary points of the un-realistic holes are changed to "interior" (Fig.2).

Figure 2. Feature detection processes

Boundary detection (Step 1.1)

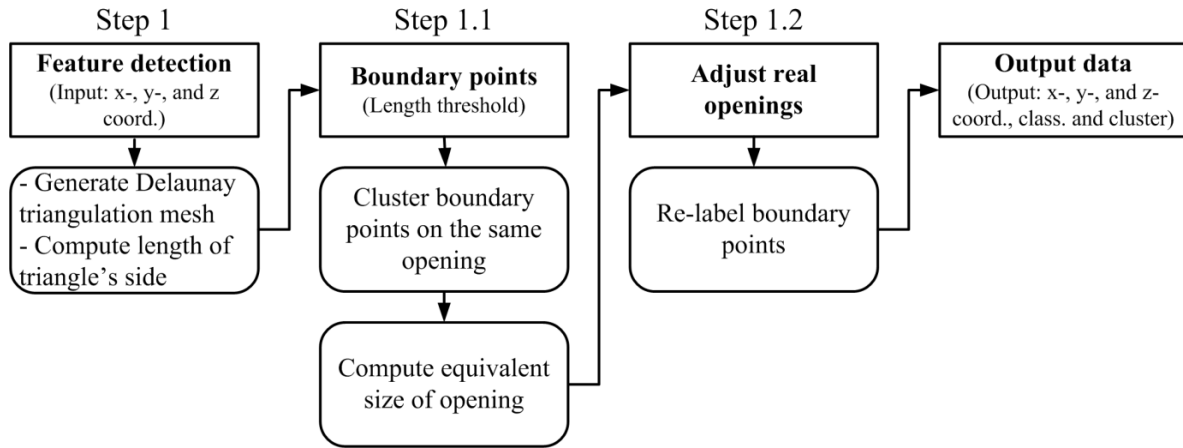As previously mentioned, in Delaunay triangulation of point cloud data, triangles with long sides form where holes exist (Fig.3a). Thus, triangles along a boundary have longer sides than those that are not. This observation was implemented to classify boundary points belonging to these large-sided triangles. In FacadeDelaunay, a 2D Delaunay triangulation mesh is automatically generated from façade sample points (Figure 3) by using existing library functions in MatLab based on Qhull (Barber et al. 1996). Accumulated lengths of all of triangle sides are then computed and stored. Using length distribution, a threshold can be determined to classify sides as short or long (Fig.4). End points of the sides on a border of a Delaunay mesh are added to the boundary points' set, if they are owned by only one triangle.

a) 3D TLS point cloud   b) Delaunay mesh   c) Close up triangles
inside the window
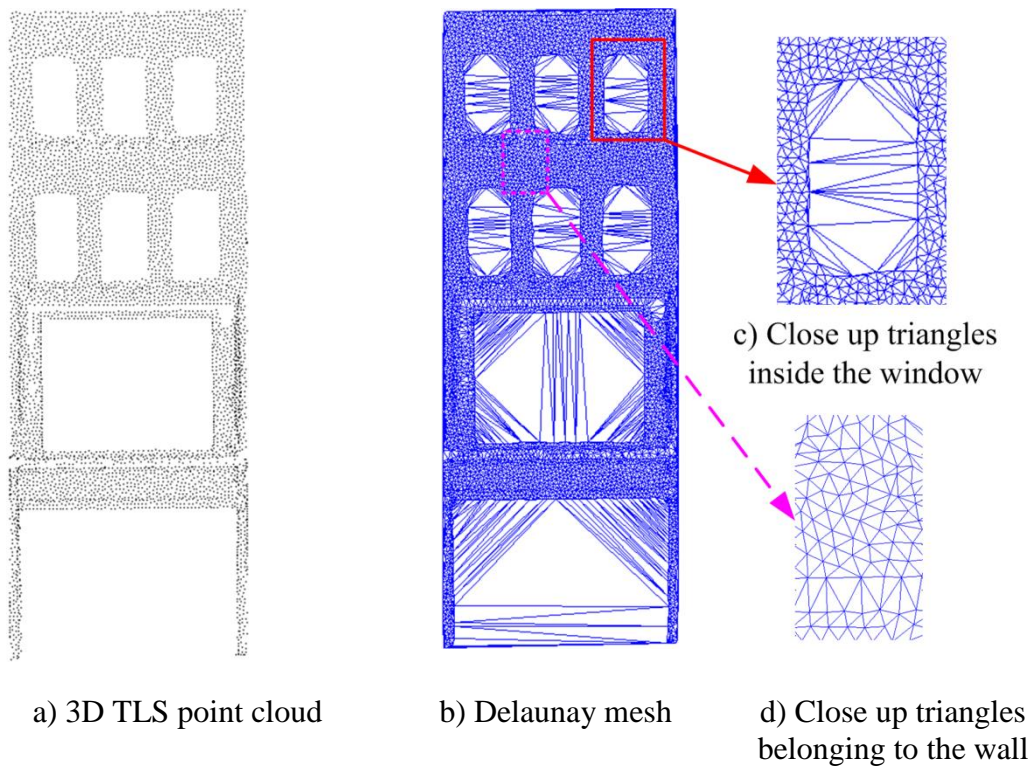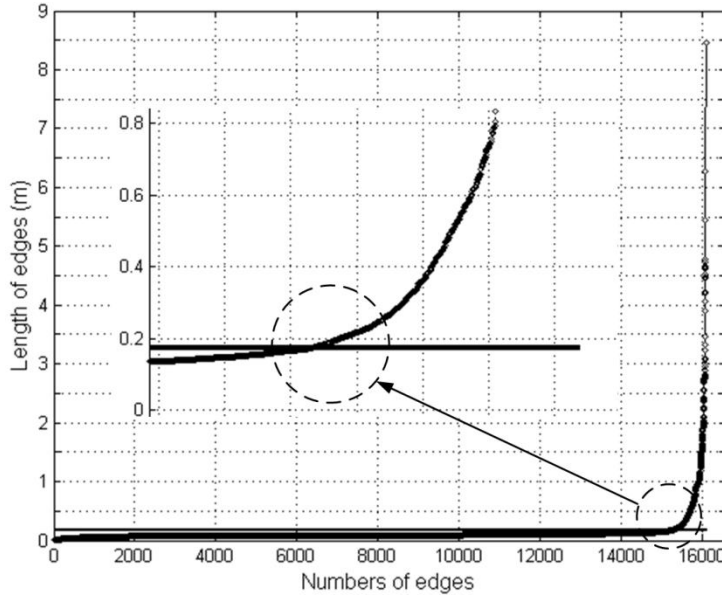
d) Close up triangles
belonging to the wall

Figure 3. Generation Delaunay triangulation mesh

a) Curve distribution of lengths of triangle sides resulting from Delaunay triangulation
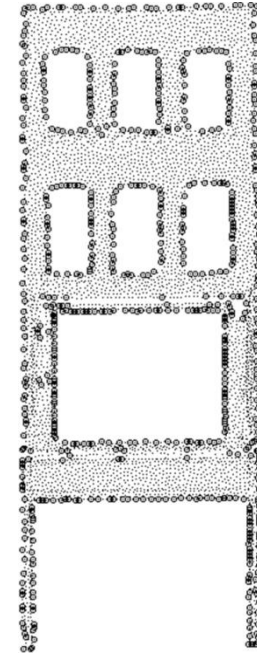
b) Initial classification of sample points: boundary and interior*

* Black dots describe interior points and black circle dots are boundary points.

Figure 4. Detection boundary points based on characteristics of Delaunay mesh

Thereafter, holes are classified as realistic (e.g. windows) or only occlusions. The boundary points on the same hole are defined as the same cluster. Intuitively, in Delaunay triangulation, boundary points on the same hole always belong to adjacent triangles. As such, starting with an arbitrary boundary point within a set of boundary points, attached triangles are searched, and then this boundary point is removed from the boundary point set. If any vertex of those triangles is a boundary point, then the point is extracted and added to the cluster. The clustering process of a hole is finished, if there are no longer boundary points to be extracted from those triangles. Searching continues for other holes, until the boundary point set is empty.

Adjusting openings (Step 1.2)

Next, detected holes are compared to characteristic openings (height, length, and rectangularity). Possible boundary lines of an opening are based on histograms (Fig.5). Histogram peaks along the x- and y-directions correspond to vertical and horizontal boundary lines, respectively (Fig.s 5b,5c), similar to work by Lee and Nevatia (2004) who used image pixel intensities across a façade or work by Okorn et al. (2010) who used a height-based histogram to detect floor and ceiling data.

a) Boundary points of window circled

b) X-coordinate histogram of boundary points along the x-direction

c) Y-coordinate histogram of boundary points along the y-direction

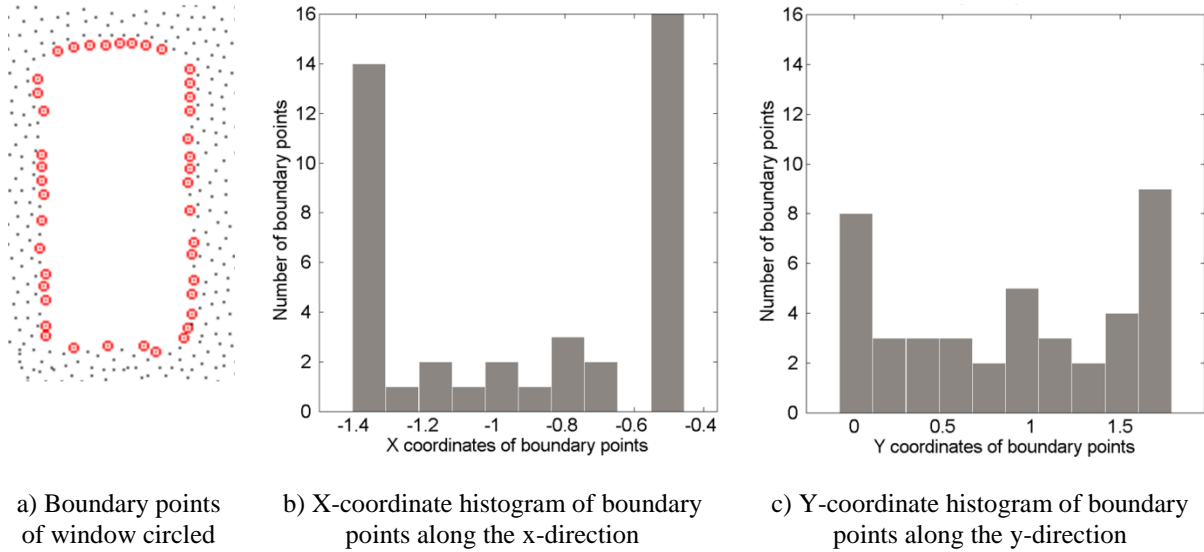Figure 5. Using histograms to determine height and length of a window

Holes are categorized as occlusions, if their characteristics differ from a predefined set of dimensions, in this case a minimum opening dimensions greater or equal to 0.4m (Pu and Vosselman 2007), and height (H_o) to length (L_o) ratio greater than 0.25 and less than 0.5 (Mayer and Reznik 2005; Ripperda 2008) (Equation 1).

$$f\left(H_o, L_o, H_o / L_o\right) = \begin{cases} \text{if } H_o \geq 0.4\text{m}; L_o \geq 0.4\text{m}; 0.25 \leq H_o / L_o \leq 5.0 : \text{Opening} \\ \text{Otherwise} : \text{Non} - \text{opening} \end{cases} \quad \text{Equation 1}$$

Solid model reconstruction (Step 2)

This step constructs a solid model from raw data by adding classification labeling to the original data points (Fig.6). A quadtree representation as the 2D derivation of an octree was employed to represent the solid model (Meagher 1982; Pulli et al. 1997). As the long-term goal herein was to fully reconstruct 3D building models, the algorithm was designed with an octree representation. The current implementation is predominantly 2D with a predefined wall thickness to generate a solid model with appropriate elements in a finite element program. Thus, the current sub-division mechanism does not fully exploit the 3D capabilities of an octree. Presently, a parent voxel is subdivided into four child voxels along the height and length directions, but no division occurs in the depth direction of the building. As the ultimate goal of the work is in 3D, the structure will be described, herein, as an octree, although the correct appearance is that of a quadtree.
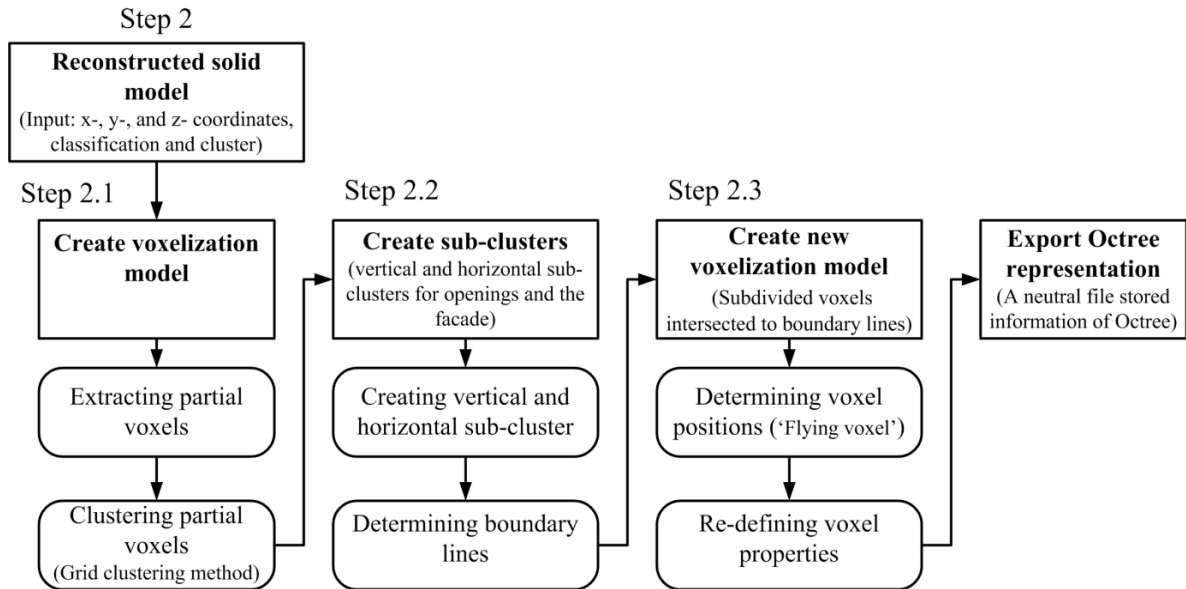
Figure 6. Reconstructed geometric model process

Octree implementation involves recursively subdividing the bounding box into smaller voxels, until a pre-designated terminal condition is reached. Various termination criteria could be used: minimal voxel size (Ayala et al. 1985), predefined maximum depth tree (Pulli et al. 1997), or a maximum number of sample points within a voxel (Wang et al. 2005). In the proposed FacadeDelaunay algorithm, a maximum voxel size is used that is less than half of the minimum feature size (e.g. if minimum opening dimension is 0.4m, voxel size must be less than 0.2m.)

Initially, a bounding box enclosing the building's entire façade is established corresponding to the initial voxel. Traditionally, a bounding box has equal edge lengths (Meagher 1982), but here it is defined by equations (2) and (3) implying non-cubic voxels

$$\text{Length} = x_{max} - x_{min}$$ Equation 2

$$\text{Height} = y_{max} - y_{min}$$ Equation 3

where $x_{max}, x_{min}, y_{max}, y_{min}$ are minimum and maximum coordinates of input sample points.

The initial voxel's origin is the point of intersection of a pair of orthogonal axes across x- and y-minimum coordinates (Fig.7 at point O) – the voxel's lower left back corner. Each voxel's geometry is stored as x-, y-, z-coordinates based on the lower-left back corner and the upper-right front corner (Fig.7 points O and O'). Additionally, each voxel is classified based on the sample points within. Voxels are empty, full, or partial. The voxel is "empty", if it contains no data, "full" if it contains exclusively interior points, and "partial" otherwise. The stored information consists of addresses, properties, and x-, y-, and z-coordinates of two corners of the voxel.
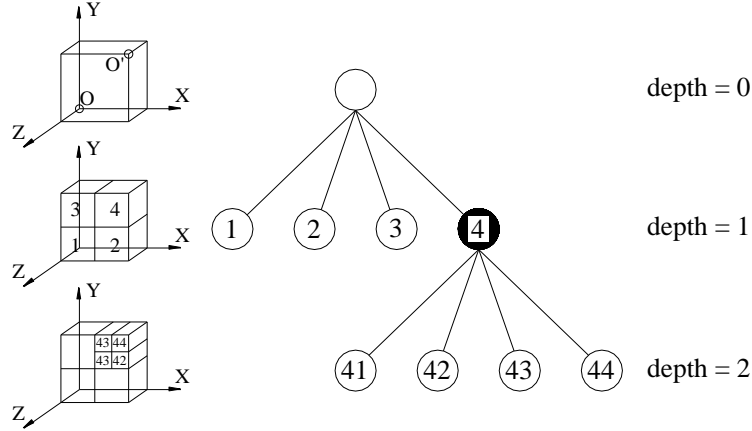
Figure 7. Octree representation for depth 2; depth 3 would have 3 digit addresses

Building façade and opening boundary lines are determined based on boundary points underlying partial voxels by using a least-squares method (Pighin and Lewis 2007). Based on voxel classification mentioned above, partial voxels containing boundary points around openings are clustered by using a flood-filling algorithm (Agoston 2005). In actual buildings, the façade and its openings generally have straight boundary lines involving left and right lines and bottom horizontal lines. This implies that these side boundary points should be in vertical and horizontal grid voxels. Based on this observation, incorrect boundary points can be removed by use of a grid clustering technique. The boundary points in the grid voxels are considered boundary lines, if the boundary points in the grid satisfy the following conditions: (1) a maximum distance between two boundary points in the grid is not less than the minimum opening size, and (2) a minimum distance between two boundary points belonging to two adjacent partial voxels is not greater than a half of the opening size (herein 0.4m is adopted minimum opening size). Subsequently, these grid voxels were divided into voxel groups representing boundary lines by comparing voxel group center coordinates to façade feature center coordinates. A voxel group center is calculated as the average coordinates of all voxel centers in the group, where the

voxel's center is based on coordinates of all its vertices, while the façade feature centers are computed as average centers of all partial voxels around the façade features.(Fig.8a). For example, a window has four voxel groups, while a door on the ground floor has three. Boundary lines with two end points are determined from coordinates of all boundary points contained within the group voxels (vertical or horizontal) by a least-squares method (Fig.8b).
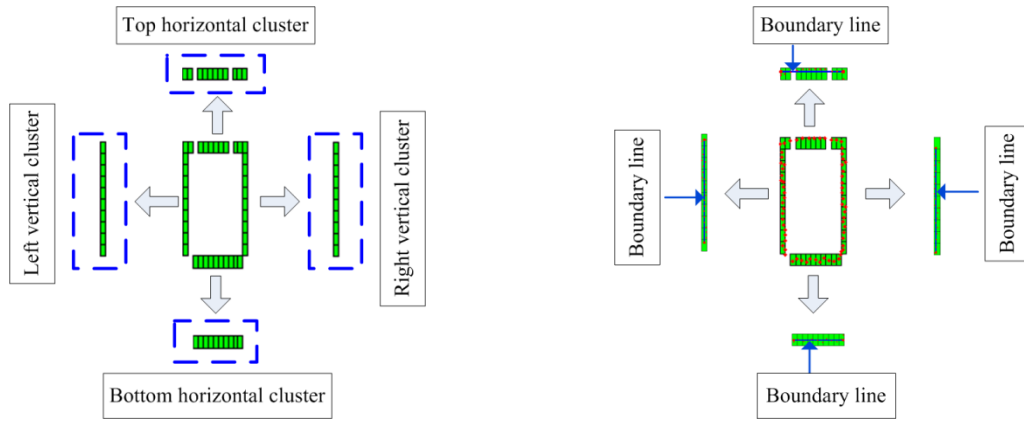
a) Clustered vertical and horizontal clusters   b) Determined boundary lines of the opening

Figure 8. A modified grid clustering technique is employed to cluster vertical and horizontal voxels and to determine boundary lines of the opening

The full voxels are then stored in a neutral file describing the geometric model of the solid wall for importing directly into commercial finite element packages. As such, voxel properties must be re-defined based on their positions. For example, they are either "inside": openings (belonging to a solid wall) or "outside" the façade (exterior to a set of façade boundary lines). To do this task, a new depth of the octree was created by dividing voxels that intersect boundary lines (Fig.9). Here, the number of child voxels depends on the number of boundary lines intersecting the parent voxel. For example, four child voxels are created, if two boundary lines

intersect the parent voxel. Whereas, no subdivision occurs, if no boundary line intersects the voxel, or if a boundary lines(s) coincides with a surface plane(s) of the parent voxel (Fig.9c).

| a) Voxelization model – octree depth 7 | b) Boundary lines | c) New voxelization model - octree depth 8 with boundary lines shown | d) Re-determination of voxel properties through "flying voxel" method* |
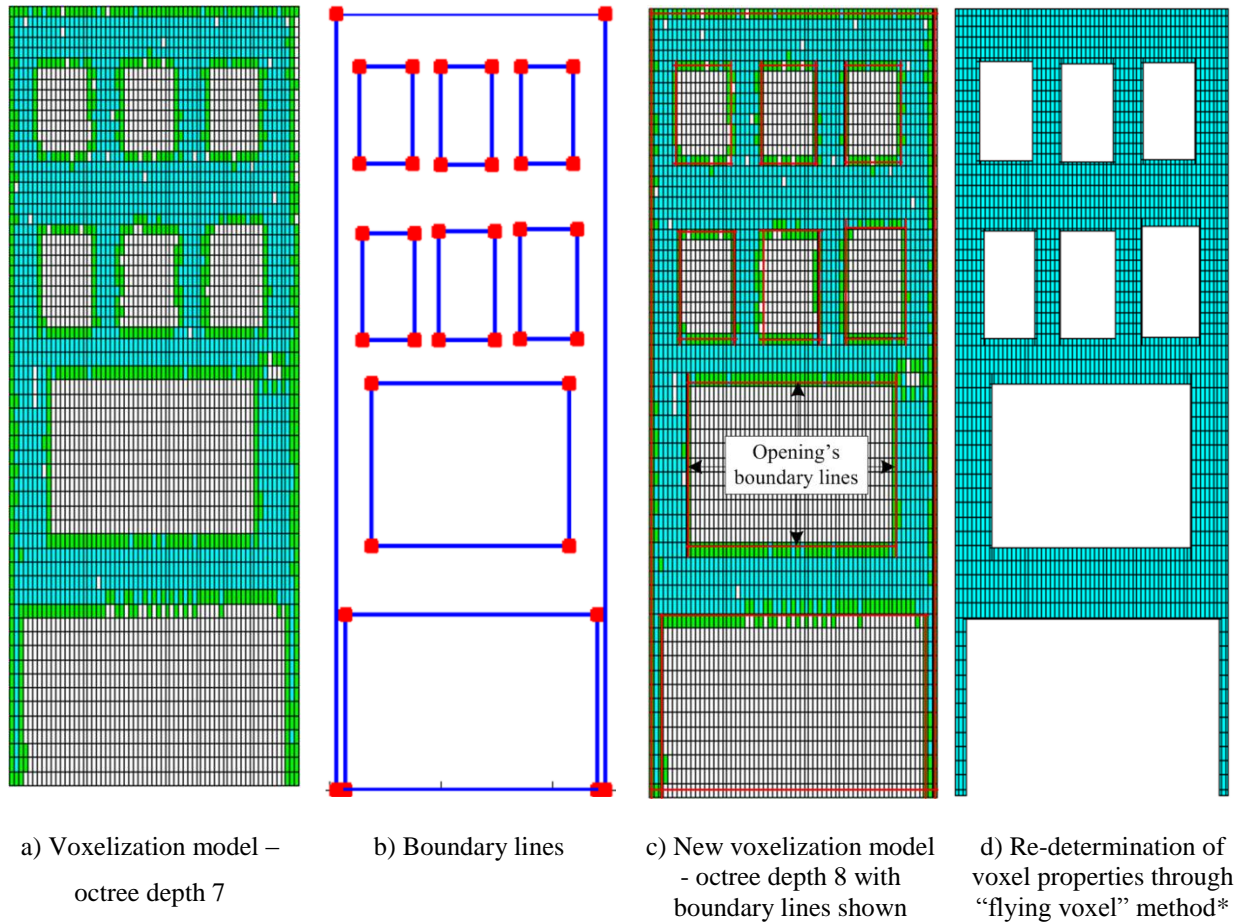
*Described in subsequent section

Figure 9. Determined façade boundaries and its openings' boundaries

In the re-voxelization model, voxels inside of openings or outside of the façade are now labeled as "empty" and all others as "full". Intuitively, voxels inside the façade and its openings were bounded by a set of closed boundary lines. The boundary lines separate voxels inside from those outside, which is similar to the Jordan curve theorem (Jordan 1887). Thus, to change categories would require a voxel to "fly" over the boundary line of an object. So, a new concept entitled the

"flying voxel" approach is introduced. This helps to determine one of three possible voxel positions (Fig.10): Case 1 − voxel outside of the façade; Case 2 − voxel inside the façade and inside an opening; and Case 3 − voxel inside the façade but not inside any opening. To do this, first the center-of-gravity of the voxel (CGV) is established based on coordinates of all vertices of the voxel. From the CGV, the first set of line segments (LS1) is established based on two end points including the CGV and then projected onto each of the bounding boxes' four sides (see four magenta lines in Figure 10a&c). A second set of line segments (LS2) is then connected from the CGV to the centre-of-gravity of each opening (the average of CGVs of all empty voxels representing the inside of an opening). Only one line segment LS2 exists per opening (see red lines in Figure 10b&c). If none of the LS1 lines intersect any of the façade boundary lines, the voxel is outside the façade – Case 1 (Fig.10a). A voxel is inside an opening (Case 2), if and only if one of the LS2 lines fails to intersect any of the boundary lines of a single opening (Figure 10b). Otherwise, the voxel is outside of any opening but still within the façade (Case 3) [Figure 10c]. Based on these cases, new voxel properties are assigned. Figure 9d shows re-characterized voxels of Figure 9c. Subsequently, all full voxels are exported into an appropriate format file for use within a commercial FEM package.

Namely, topology and geometry of the full voxels are converted to a Boundary Representation (B-Rep) scheme that defines a solid model. Edges and faces containing edges are converted to a Non-Uniform, Rational and B-spline (NURBs) format by a number of knots (ANSYS Academic Research Release 13.0); see Appendix A for details of converting the full voxels into a neutral file. The approach was done according to Hinks et al. (Hinks 2011) and as explained in Appendix A.

|a) Case 1: a voxel outside the façade|b) Case 2: a voxel inside the opening|c) Case 3: a voxel outside the opening but inside the façade|

* While rectangles: empty voxels
* Dark gray rectangles: full voxels belonging to the solid wall
* Light gray rectangles: full voxels either inside the opening or outside the façade should be determined as empty voxels by use of the flying voxel method
* Black rectangle: the voxel's centre-of-gravity
* Unfilled black circle: projection of the voxel's centre-of-gravity onto the bounding box
* Filled black circle: the centre-of-gravity of each opening
* Double dash dot lines: bounding box
* Dash dot lines: façade boundary lines
* Dash lines: boundary lines of the opening
* Centre lines: lines connecting a voxel's centre to its projection on the bounding box (LS1)
* Continuous line: lines connecting a voxel's centre to an opening centre (LS2)

Figure 10. Determining position of voxels in the model

EXPERIMENTAL RESULTS AND DISCUSSION

To validate the algorithms, four datasets were created for each of three brick buildings in Dublin, Ireland selected for their proximity to an upcoming metro project (Fig.s 11-13). Point cloud data were collected with a Trimble GS200 terrestrial laser scanner. The first set was of the original scans (NS00) after being co-registered and cleaned of points behind the expected building

facade. The other 3 were subsets using random re-samplings with expected surface sampling step size of 20mm (S20-2500pts/m^2), 50mm (S50-400pts/m^2), and 75mm (S75-175pts/m^2) (Table 1). These enabled testing of the algorithm's sensitivity to sampling density.
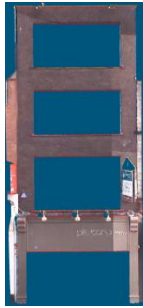
Table 1. Dataset Sizes

Building	Sampling dataset			
	NS00	S20	S50	S75
Building 1: 2 Anne St. South	264,931	51,171	9,909	4,643
Building 2: 5 Anne St. South	190,865	51,884	11,119	5,366
Building 3: 2 Westmoreland St.	650,306	353,848	71,155	35,468

Each building had four geometric models: B1FDNS00 describes the geometric models of Building 1 (B1) reconstructed by using the FacadeDelaunay algorithm on the original dataset (NS00), while B2FDS20 describes the same applied to the sampling point S20 (distance of 20mm between points) dataset for Building 2 (B2). Due to space limitations, only one set of solid models is graphically presented herein for each building.

| a) Photo of Building 1 (4.95m w x12.16m h) | b) Scanning data in the RealWorks Survey program | c) Point cloud after cleaning and resampling | d) Boundary points shown in red |

| e) Initial octree representation | f) Octree representation subdivided by boundary lines | g) Solid model representation | h) CAD drawing from physical survey |

Figure 11. Facade reconstruction for Building 1 based on dataset of 2500pts/m^2 (S20-distance

between two adjacent sample points no less than 20mm)

| a) Photo of Building 2 (4.90m l x 13.28m h) | a) Scanning data in the RealWorks Survey program | b) Point cloud after cleaning and resampling | d) Boundary points shown in red |

| e) Initial octree representation | f) Octree representation subdivided by boundary lines | g) Solid model representation | h) CAD drawing from physical survey |

Figure 12. Facade reconstruction for Building 2 based on a dataset of 400 pts/m^2 (S50-distance between two adjacent sample points no less than 50mm)

20

a) Photo of Building 3: (19.36m l x 17.0m h)

b) Scanning data in RealWorks Survey program

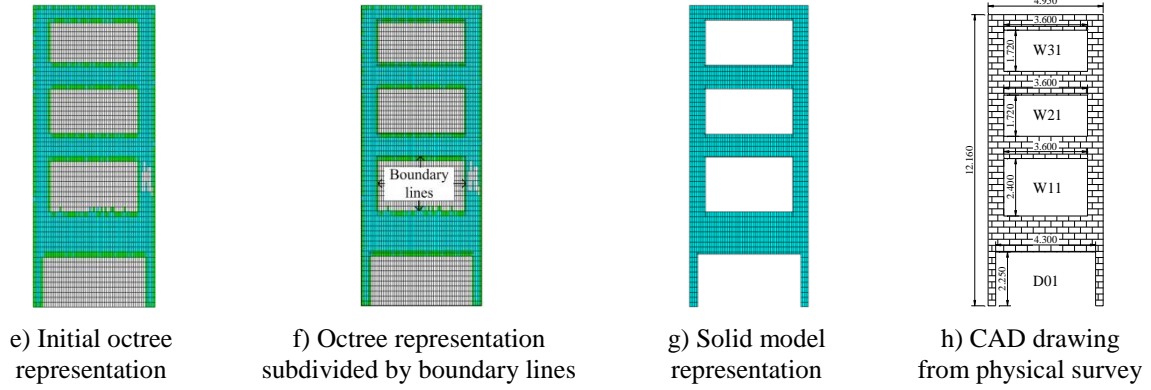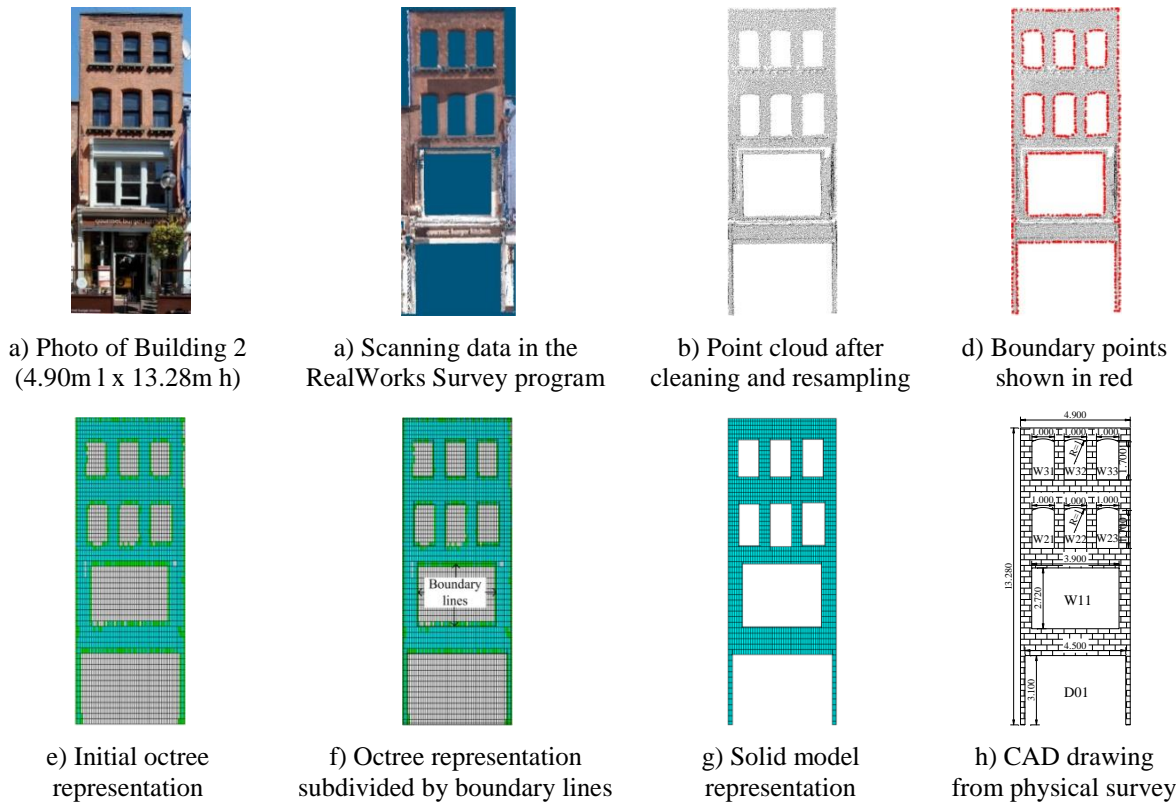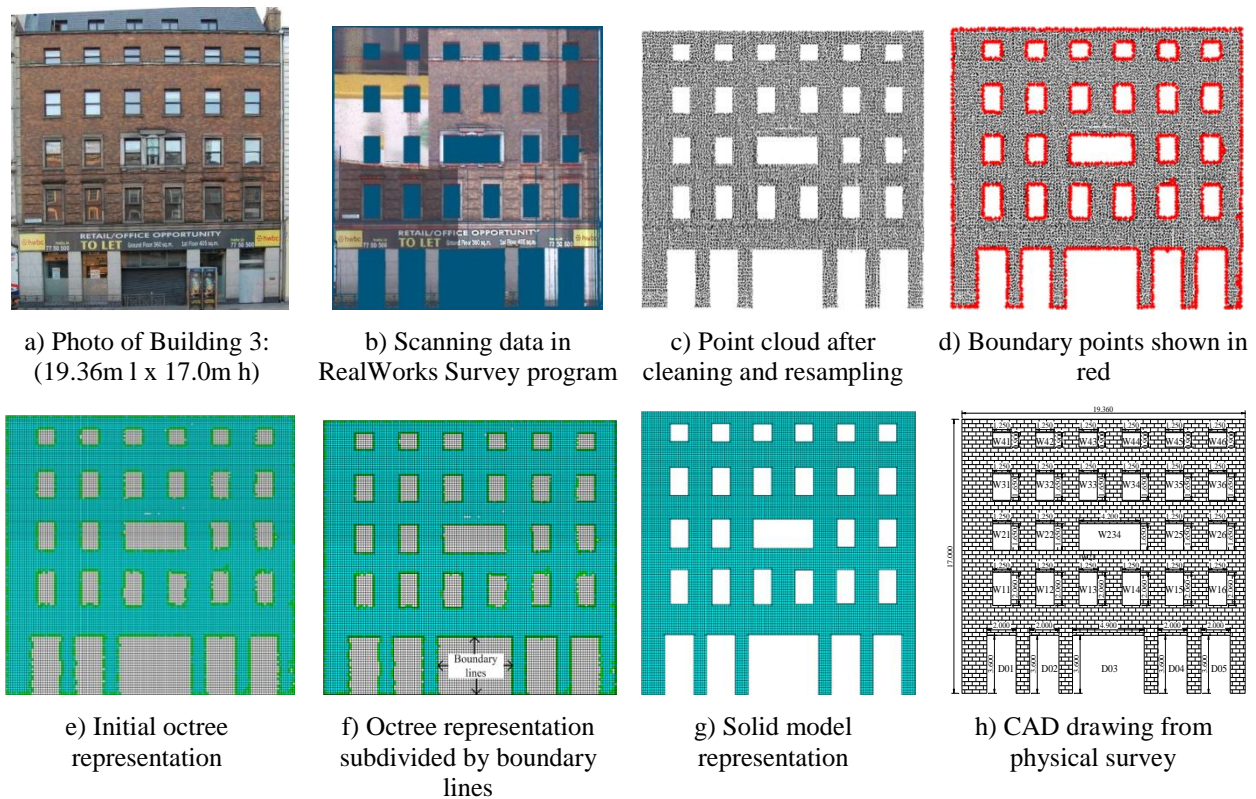c) Point cloud after cleaning and resampling

d) Boundary points shown in red

e) Initial octree representation

f) Octree representation subdivided by boundary lines

g) Solid model representation

h) CAD drawing from physical survey

Figure 13. Facade reconstruction for Building 3 based on a dataset of 175 pts/m^2

(S75-distance between two adjacent sample points no less than 75mm)

Quality of boundary point detection

The FaçadeDelaunay algorithm consistently detected boundary points of all openings for each of the three building facades. However, some boundary points on the corner of openings were not detectable, because the triangle's longest side was smaller than or equal to the threshold. When this occurs, the end points of that side are mistakenly characterized as interior points.

Processing time and data density

All models were run on a Dell Precision Workstation T5400 with Intel (R) Pentium (R) Xeon
(8CPU) CPU speed 2GHz with 24 Gb RAM. Octree representation consumed only a minor
portion of the total processing time. For example, 1.466 minutes for a dataset of 190,865 points
(Building 2) [shown as 5.28 (log190865) on x-axis on Figure 14] versus the 117.143 minutes for
the whole process (Figure14), because the feature detection algorithm must pass through the
entire dataset.

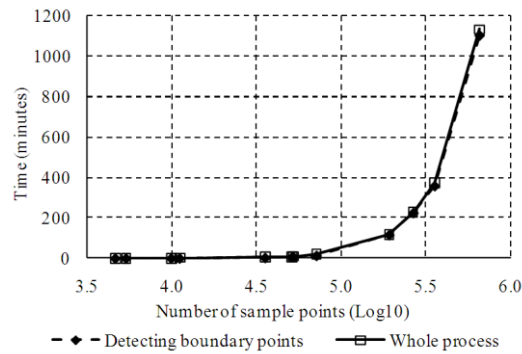Figure 14. Running time of the algorithms

A length threshold is needed for distinguishing triangle sides (long versus short), which depends
on sampling density. In this study, lengths 0.15 m-0.175 m reliably detected sufficient boundary
points to reconstruct geometric models. Overly sparse data will generate boundary triangle
characteristics too similar to compare to the solid wall ones (Fig. 15).

22

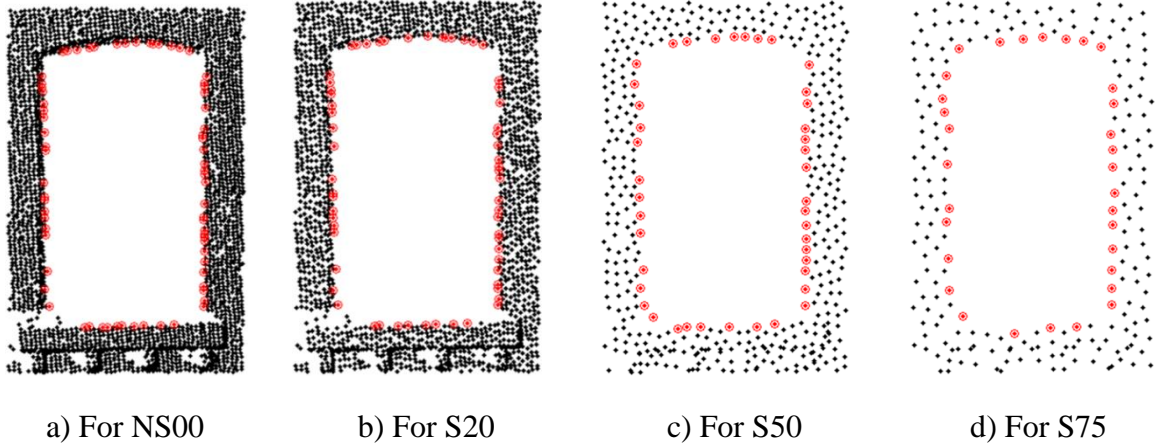|			
a) For NS00	b) For S20	c) For S50	d) For S75

Figure 15. Boundary points of a top left window of Building 2 with various sampling density of datasets to show parameter sensitivity

GEOMETRIC VALIDATION

The geometric correctness of the automatically generated solid models was compared to CAD-based models derived from published, measured drawings (held Dublin City Council) [Fig. 11-13]. Validation considered both global and local responses, as well as an estimated error of the geometric models (see Appendix B). Overall façade dimensions and total opening area are indicative of global response. Locally, opening dimensions and positions are important. Global dimensions were assessed by use of relative errors. Building components (windows and doors) and their positions were validated by a validation metric, to provide an estimated standard of uncertainty and an error bounds by means of a statistical procedure. From these approaches, solid model reliability was assessed.

Global quantities of interest

Auto-generated façade dimensions and opening areas (Tables 2) were compared to CAD models by using relative error to quantify discrepancies (see Equation 4 in Appendix B).

23

Table 2. Derived overall dimensions of three facades

Aspects		Building 1			Building 2			Building 3		
Method	Dataset	Overall façade parameters			Overall façade parameters			Overall façade parameters		
		Length (m)	Height (m)	Opening area (m^2)	Length (m)	Height (m)	Opening area (m^2)	Length (m)	Height (m)	Opening area (m^2)
CAD		4.950	12.160	30.699	4.900	13.280	30.699	19.360	17.000	96.200
Façade Delaunay	NS00	4.940	12.062	29.945	4.884	13.306	29.945	19.299	16.922	95.976
	S20	4.931	12.052	30.007	4.881	13.284	30.007	19.308	16.918	96.373
	S50	4.916	12.051	30.138	4.863	13.274	30.138	19.288	16.908	96.682
	S75	4.896	12.025	30.568	4.845	13.248	30.568	19.268	16.900	97.598

Observing these, the algorithm slightly underestimated lengths and heights – generally less than 1.12% (Figure 16a). That is believed to have been due to the removal of sample points on actual boundaries of the façade during segregation of the façade's data set from whole scan (Tang et al. 2009b). Maximum and minimum relative errors of the façade lengths were respectively -0.2% (<10 mm) [B1FDNS00] and -1.1% (<54 mm) [B1FDS75] for Building 1, -0.33% (<16 mm) [B2FDNS00], -1.12% (<55 mm) [B2FDS75] for Building 2, and -0.27% (<52 mm) [B3FDS20] and -0.48% (<100 mm) (B3FDS75) for Building 3. Similarly, the relative errors of the façade heights ranged from -0.81% (B1FDS20) to -1.11% (B1FDS75) for Building 1, corresponding to 98mm-135mm, from -0.24% (<32 mm) [B2FDS75] to 0.2% (~26 mm) [B2FDNS00] for Building 2, and from -1.59% (B3FDS75) to -0.46% (B3FDNS00) for Building 3, corresponding to 100 mm and 78 mm, respectively (Fig.16a).

The absolute relative errors of opening areas were <3% for Buildings 1 and 3, and 4.47% (1.53m^2) for B2FDNS00. Additionally, most Building 3 solid models overestimated the opening area with a maximum absolute relative error of 1.45% (~1.4 m^2) [B3FS75], except for B3FDNS00 which underestimated -0.23% (<0.22 m^2). Minimum absolute relative errors were

0.43% (B1FDS75), -2.73% (B2FDS75) and 0.18% for Building 3 (B3FDS20), corresponding to

~0.1 m^2, <0.9 m^2, and ~0.5 m^2 for Building 1, 2 and 3, respectively (Fig.16b and Table 2).

a) For height and length of façades   b) For façade opening area
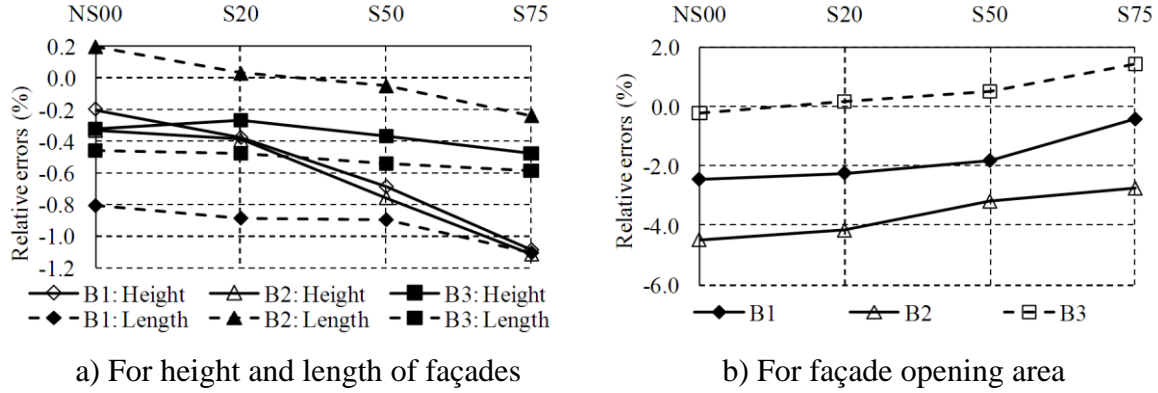
Figure 16. Relative error of quantities of interest for three building facades

Local quantities of interest

At a local level, opening dimensions and positions were benchmarked by using a validation metric. Statistical results are shown in Tables 3 and 4 for dimensions and positions, respectively. Generally, errors were small, <22.9 mm of an average absolute error (B2FDNS00) and the standard uncertainty was <41.8 mm (B1FDS75). The larger Building 3 generated smaller errors than those for Buildings 1 and 2. Standard uncertainties of Building 1 varied from 36.4 mm (B1FDNS00) to 41.8 mm (B1FDS75) and for Building 2 from 29.9 mm (B2FDNS00) to 34.3mm (B2FDS75). Whereas those of Building 3 varied between 6.2 mm (B2FDNS00) and 6.7 mm (B2FDS75) [Table 3].

Table 3. Statistical error of dimensions of the openings of the façade (Unit: mm)

Aspects	Building 1				Building 2				Building 3			
	NS00	S20	S50	S75	NS00	S20	S50	S75	NS00	S20	S50	S75
Average	-19.7	-17.3	-10.6	6.4	-22.9	-12.6	-0.4	7.7	0.7	5.7	9.5	18.4
Min	-197.0	-203.0	-208.0	-175.0	-371.0	-369.0	-362.0	-360.0	-57.0	-71.0	-45.0	-39.0
Max	90.0	95.0	98.0	137.0	124.0	154.0	163.0	168.0	137.0	138.0	163.0	162.0
Std.	102.8	110.0	112.4	118.1	119.6	130.0	132.9	137.1	46.1	47.2	48.9	50.1
Std. uncert.	36.4	38.9	39.7	41.8	29.9	32.5	33.2	34.3	6.2	6.3	6.5	6.7
Lower est.	-88.6	-90.9	-85.9	-72.7	-75.3	-69.6	-58.7	-52.3	11.0	16.2	20.4	29.7
Upper est.	49.1	56.4	64.7	85.5	29.6	44.3	57.8	67.8	45.7	47.1	49.4	53.0

Geometric accuracy was largely proportional to sampling density for Building 1 – standard uncertainty varied from 36.4 mm to 41.8 mm when input datasets changed from NS00 to S75. However, in the larger building (Building 3), standard uncertainty was not greatly changed (from 6.2 mm to 6.7 mm) with decreasing density (Table 3).

With a traditional level of confidence of 90%, estimated absolute errors of the smaller buildings (1 and 2) were higher than those of the larger Building 3. Maximum estimate absolute errors were 90.9 mm (B1FDS20) for Building 1 and 75.3 mm (B2FDNS00) for Building 2, while only 53.0 mm (B3FDS75) for Building 3 (Table 4, Fig.18). Those values reflected differences in opening dimensions of 5.3%, 7.5%, and 5.9%, respectively, where the actual minimum opening dimensions were respectively 1.72 m, 1.0 m and 0.9 m for Buildings 1, 2 and 3. The lowest and highest absolute errors for Building 1 were respectively -90.9 mm (B1FDS20) and 85.5 mm (B1FDS75); for Building 2, -75.3 mm (B2FDNS00) and 67.8 mm (B2FDS75), and for Building 3 only 11.0 mm (B3FDNS00) and 53.0 mm (B3FDS75) [Table 3 and Fig.17].
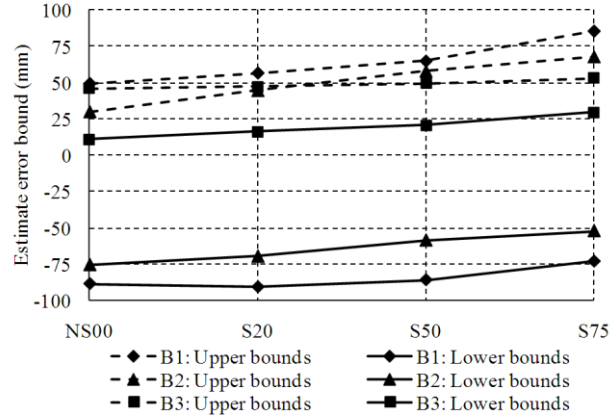
Figure 17. Estimate error bounds of opening dimensions from various sampling datasets

To consider discrepancies of opening positions, the local coordinate system of each model was roughly assumed as the lower left corner of the model. The lower left corner of each opening is referred to as the origin of the local coordinate system of that opening. Statistical errors of position (Table 4) show good detection of opening position, although Buildings 1 and 2 models had higher errors than those of Building 3 and were similar to the errors of previously described opening dimensions. Standard uncertainties were less than 29.7 mm (B1FDS75) for Building 1, 27.6 mm (B2FDS75) for Building 2, and 7.7 mm (B3FDS75) for Building 3 (Table 4).

As expected, models reconstructed from denser data gave more accurate results than sparser ones. For example, the standard uncertainty was 25.8 mm for B1FDNS00 compared to 29.7 mm for B1FDS75. Estimated absolute errors of position of openings were generally less than 100mm (Table 4 and Fig.18) but varied widely by building. For example, for datasets of S75, they ranged from -124.2 mm to -11.8 mm for Building 1 and -159.5 mm to -62.8 mm for Building 2, and -116.3 mm to -90.5 mm for Building 3 (Table 4 and Fig.18).

Table 4. Statistical error of position of the facade components (Unit: mm)

Aspects	Building 1				Building 2				Building 3			
	NS00	S20	S50	S75	NS00	S20	S50	S75	NS00	S20	S50	S75

|											
Average	-47.3	-55.4	-75.6	-86.6	-81.4	-96.1	-108.4	-175.8	-83.7	-83.2	-102.6	-120.3
Min	-200.1	-208.6	-233.1	-250.5	-236.2	-280.2	-279.3	-420.1	-188.3	-184.3	-197.2	-225.6
Max	8.6	0.0	0.0	0.0	151.0	145.1	131.5	115.3	0.0	17.8	0.0	0.0
Std.	81.1	81.5	86.9	88.3	103.8	112.6	110.6	162.3	53.3	55.5	57.5	63.7
Std. uncert.	28.7	28.8	30.7	31.2	26.0	28.1	27.6	40.6	7.1	7.4	7.7	8.5
Lower est.	-101.7	-110.1	-133.8	-145.7	-126.9	-145.5	-156.9	-246.9	-95.6	-95.6	-115.5	-134.6
Upper est.	7.0	-0.8	-17.4	-27.4	-35.8	-46.8	-59.9	-104.6	-71.7	-70.8	-89.7	-106.1
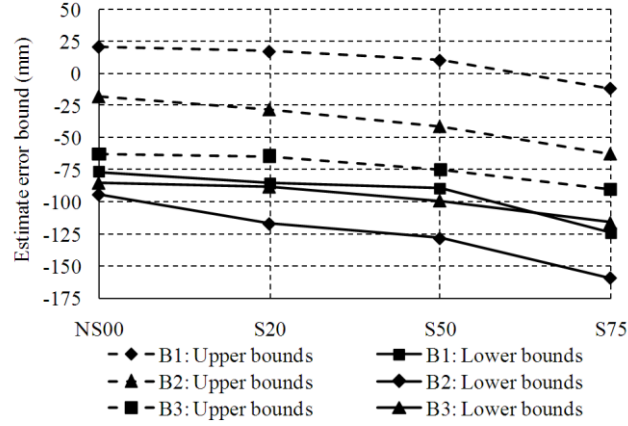
Figure 18. Estimate error bounds of opening position from various sampling datasets

NUMERICAL ASSESSMENT

As the main goal of the proposed algorithm is reconstructing solid models for existing building façades, for generating FEM meshes the impact of the aforementioned geometric discrepancies must be discussed. To test the usability of these models for a relevant case, the responses of the FEM models derived from the algorithm were compared to ones based on CAD drawings derived from measured drawings submitted for planning permission. A solid model (B1FDNS00) of Building 1 was selected for further investigation. The FEM parameters were previously validated based on large-scale experimental work (Truong-Hong and Laefer 2008).

Non-linear analysis is adopted for analyzing the solid model of Building 1 (Fig.19a and 20a), where a macro modeling strategy was employed to model the building wall by using a SOLID65 element in an ANSYS Mechanical APDL product (ANSYS Academic Research Release 13.0). Additionally, a William Warnke (WW) failure criterion and Drucker-Prager (DP) yield criterion

28

built into the ANSYS program are respectively to model masonry behavior in tension and compression. The WW failure criterion provides a tension cut-off for the DP yield criterion. The models were tested under a set of mechanical properties selected from existing experimental reports and the peer-reviewed literature (see Laefer et al. 2011b for a full description). A sample material, which represents the medium masonry properties used for this analysis were for elastic behavior: 3,480 MPa of Young's modulus and 0.16 of Poisson ratio and for plastic behavior: 26.15/1.15 MPa of compressive/tensile strength, 6.81 MPa of internal cohesion, 35^0 of internal friction angle and 10^0 of dilatancy angle. The analysis was conducted under self-weight and imposed displacements due to excavation-induced foundation settlements, in which the displacements were directly applied to nodes on the bottom of the model (Fig.s19b and 20b). The minimum/maximum vertical and horizontal displacements were respectively, -44.5/-17.8 mm and -28.3/-17.7 mm (note that negative values shows imposed displacements vectors opposite positive direction of axes in global coordinate system).
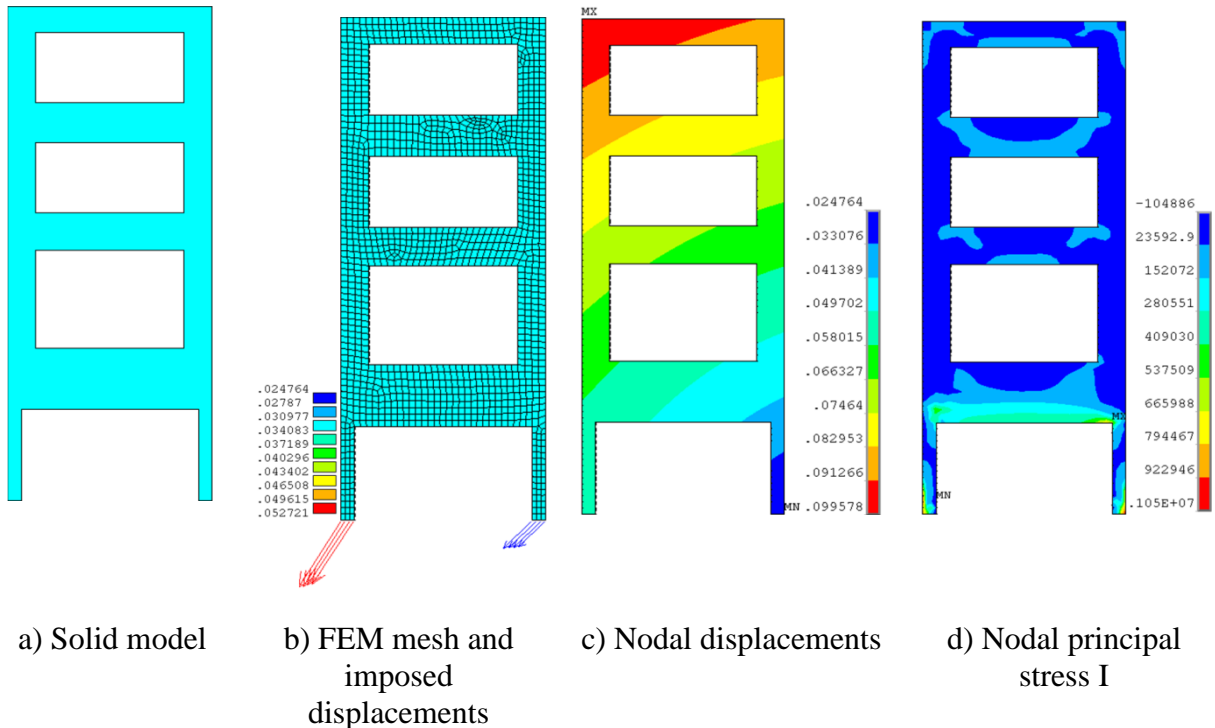

| a) Solid model | b) FEM mesh and imposed displacements | c) Nodal displacements | d) Nodal principal stress I |

Figure 19. Structural analysis of Building I with a solid model obtained from CAD drawing

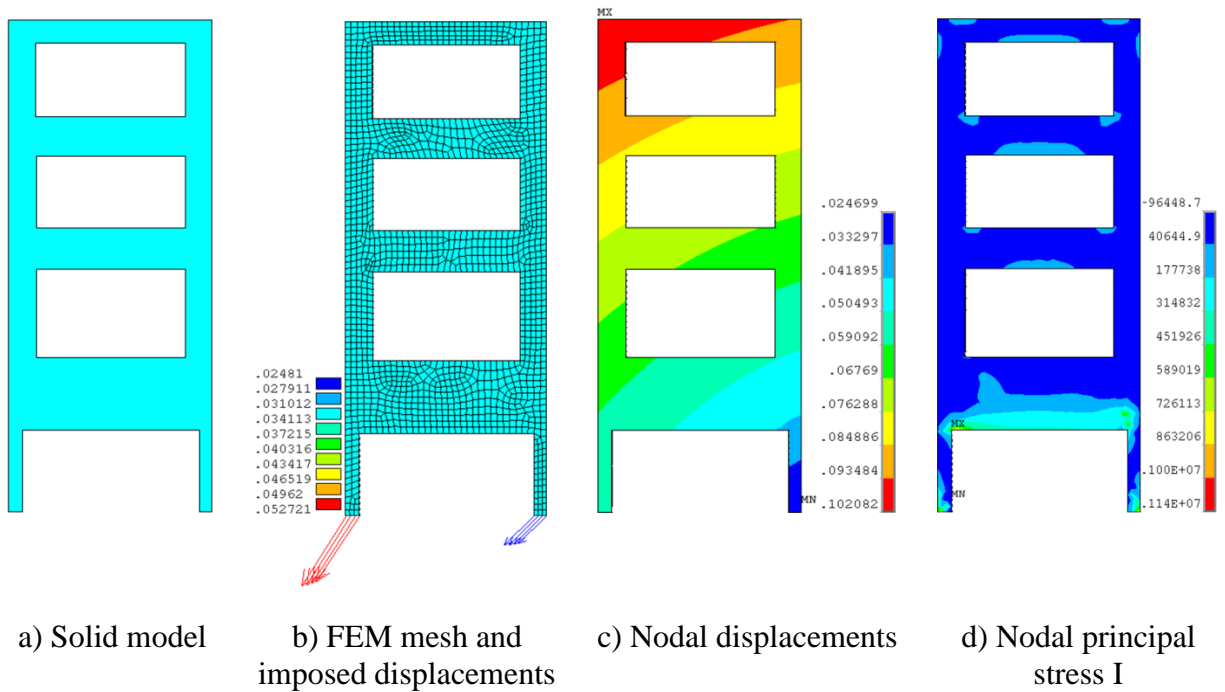| a) Solid model | b) FEM mesh and imposed displacements | c) Nodal displacements | d) Nodal principal stress I |

Figure 20. Structural analysis of Building I with a solid model obtained from the algorithm

The numerical analysis showed a consistency of nodal displacements and principal stress I between the FEM models based on the solid model generated from the proposed method and ones from the CAD drawing (Fig.s 20c-d and 21c-d). There is no more than a 2.5% difference in maximum nodal displacements, (maximum displacement is 99.6 mm for FEM model B1CAD
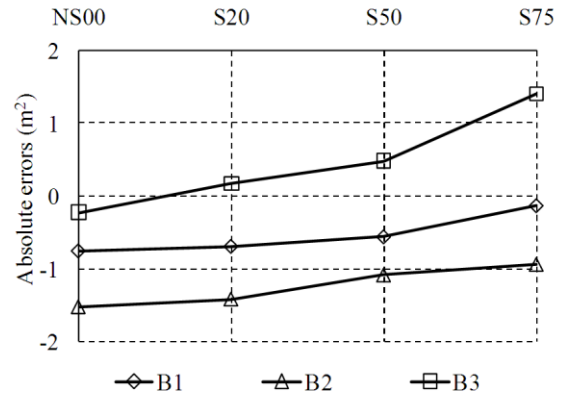
and 102.1 mm for algorithm-based B1FDNS00). Similarly, the maximum nodal principal stress 1 in the FE based B1FDNS00 was also 8.5% greater than the CAD based one, in which the maximum principal stress 1 in FE based B1FDNS00 and B1CAD are 1.14 MPa and 1.05 MPa, respectively. In terms of an engineering perspective, these differences of in FE results due to small geometric discrepancies with the solid models were mostly less than 5%, which is generally an allowable uncertainty level within structural design [e.g. the Load and Resistance Factor Design specification allows a nominal force effect increase of 5% to consider ductility, redundancy, and operational importance (Hoffman et al. 1996)]. Thus, this proposed method can be used for auto-generating computational models from TLS data.

Discussion

By comparing input densities to the quality of auto-generated geometric models (Figure 19), the re-sampling datasets S20 and S50 showed reconstruction with errors mostly less than 30 mm for the façade dimensions and 0.7 m^2 of opening area, when compared to solid models generated from the original dataset (NS00), whereas processing times for the S20 and S50 datasets were 3 to 350 times faster than for the NS00 datasets. For example, for Building 3 the processing time was 19.5 minutes for S50 but 1,132.5 minutes for NS00. Errors increased at a fairly reasonable rate through S50, but tended to quickly accelerate beyond this point when densities were further diminished. For example, absolute errors between B3FDS75 against B3FDNS00 increased 31mm for the façade length and 22 mm for the façade height and ~1.6 m^2 for opening area (Figure 21).
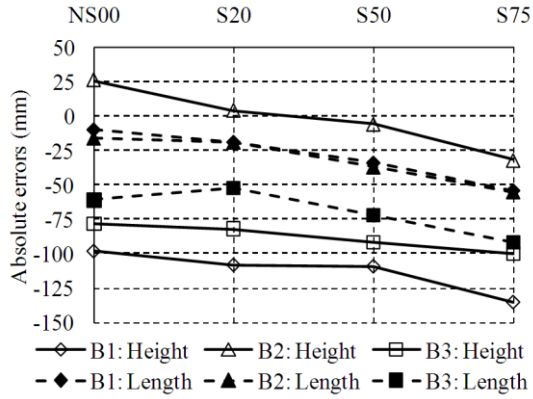
| a) For façade height and façade length | b) For opening area |

Figure 21. Absolute errors of the important parameters (façade dimensions and opening area) under various sampling density datasets

Data from the smaller buildings (Building 1 and 2) generated larger errors than those from the larger building (Building 3). This can be explained by Building 1's errors being dominated by a part of a canopy atop the door and highly articulated window ledges obstructing data collection for upper floors (see Figure 11-13). Additionally, for Building 2, on the first floor, the opening dimensions in the CAD drawing excluded the window frame, while the window reconstructed from TLS data included that. The reconstructed first floor window was significantly smaller than in the CAD drawing, for which the window height of B1FDS50 was less 208 mm and window length of B2FDNS00 was also less than 371 mm (Table 3).

While, errors could be reduced by incorporating the scanner's perspective when deriving an equivalent plane of a building façade, instead of directly projecting the data, as done herein, the question remains one of contribution. To better assess this, the solid model for Building 1 is visually compared to those generated from a non-CAD based, one from a commercial software package, and one from a previous voxel based approach, that does not incorporate refined feature detection or hole filling (Figure 22). These models were selected for comparison, as all of the

processes were specifically for transformation of TLS into solid models for FEM usage. The FacadeDelaunay approach is the only one of three that successfully filled unrealistic holes caused by data occlusions. Furthermore, a rapid visual inspection compared to the CAD drawing also shows a truer representation. Based on this comparison and the geometric and functional closeness of the FacadeDelaunay model compared to that generated from measured drawings presented in the previous section the potential usefulness of the approach is clearly established.

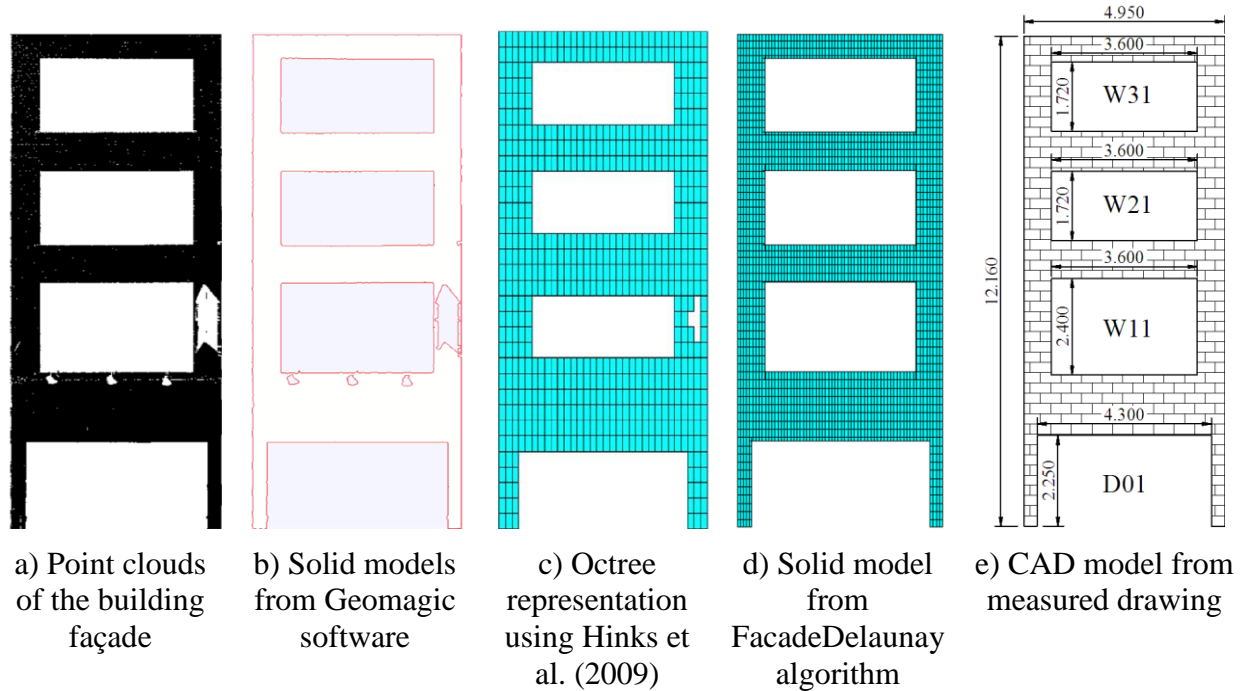| a) Point clouds of the building façade | b) Solid models from Geomagic software | c) Octree representation using Hinks et al. (2009) | d) Solid model from FacadeDelaunay algorithm | e) CAD model from measured drawing |

Figure 22. Original data and three derived solid models solid of Building 1 [(mage (b) and (c) adapted from Laefer et al. 2011)

While the proposed approach was successful in filling unrealistic holes caused by data occlusions this may fail in cases where (1) the occlusion pattern is similar to common openings

in building façade; (2) there is no distinction between two adjacent openings; or (3) the data occlusions are of the same relative size as the actual openings.

CONCLUSIONS

The FacadeDelaunay algorithm was proposed for boundary and feature detection for automatic reconstruction of building façades for computational modeling from point cloud data. Length thresholds in the range 0.15-0.175 m and knowledge about window dimensions are shown to generate sufficient boundary points for reconstructing brick building facades and their openings even when the original data set contains occlusions. By introducing the "flying voxel" method, voxels in an octree representation points were quickly classified as whether or not they belonged to the solid wall in a subsequent step of building model reconstruction.

Through experimental tests, the proposed algorithm successfully detected all openings and reconstructed all building façades, as well as automatically filling occlusion-based holes. Relative geometric errors were less than 1.2% for overall dimensions and 3% for opening area when compared to CAD based models. The validation process showed that TLS re-sampled datasets of as little as S20 and S50 could be adequate for facade reconstruction when compared to the functionality of those generated from CAD drawings. Results from application of an excavation-induced subsidence trough showed resulted within the general 5% allowable level of uncertainty.

To further enable the functionality of this approach the algorithm it is necessary to expand the range of detectable shapes, especially for building openings. Additional improvements in the geometric results might be achievable through the incorporation of previously unused, latent

sample point information (such as intensity or RGB values) at local regions such as window frames. Finally, increasing automation and applicability of this method will require its extention to fully 3D models and its integration with a procedure appropriate to eliminate irrelevant sample points.

APPENDIX A. CONVERSION OF VOXELS INTO A NEUTRAL FILE

The octree representation with all voxels characterized as "full" or "empty" iss converted into a neutral file for computational modeling in the commercial FEM code ANSYS (YEAR?). Only full voxels are represented in the neutral file. Topology and geometry of these were converted to a Boundary Representation (B-Rep) scheme that defined a solid model (Figure 20). Topologically volume area, lines, and keypoints were stored. Each voxel is converted into eight vertices, twelve edges, six faces, and one volume. Eight vertices of the voxel are converted into eight keypoints associated with their x-, y-, and z- Cartesian coordinates. Edges and faces are converted to a Non-Uniform, Rational and B-spline (NURBs) format by a number of knots (ANSYS YEAR?), in which the knots used for the definition must have multiplicity in the first and last knot equal to the NURBs order, which are respectively 4 and 16 control points for an edge and a face. An edge is defined by two keypoints (Figure 20a) and is oriented (as discussed below). As such, edge E_{12} differs from edge E_{21} ($-E_{12}$) even though they both connect vertices P_1 and P_2 (Figure 20a). Similarly, each face is defined by a list of edges, which is created a closed path (Figure 20b). These are arranged according to a right-hand rule, to generate an outward pointing normal vector. If the edge direction is opposite its face orientation, then the line number is negative and otherwise positive (ANSYS YEAR?). Lastly, a volume is defined by a list of faces. These are implemented in MATLAB (The MathWorks 2007a).

P₁ E₁₂ P₂

P₁ -E₁₂ P₂

P₁ E₁ P₂

E₄ F E₂

P₄ E₃ P₃

V

F₂

F₃ F₆ F₄

F₁ F₅

$P_1 = \{x_1, y_1, z_1\}$
$P_2 = \{x_2, y_2, z_2\}$
$E_{12} = \{P_1, P_2\}$
$-E_{12} = \{P_2, P_1\}$

$F = \{E_1, E_2, E_3, E_4\}$

$V = \{F_1, F_2, F_3, F_4, F_5, F_6\}$

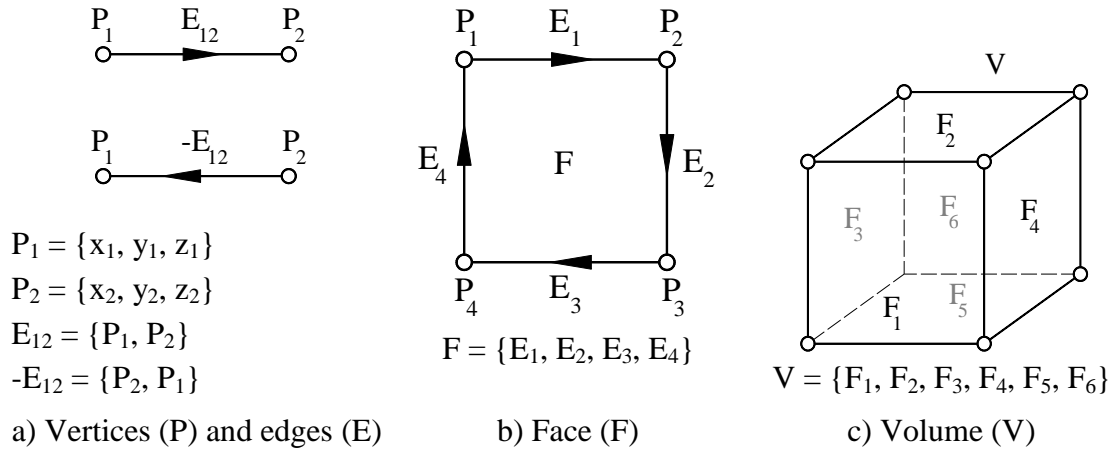a) Vertices (P) and edges (E)   b) Face (F)   c) Volume (V)

Figure 23. Label and organization of vertices, edge and faces of the voxel by B-Representation

APPENDIX B. FUNDAMENTAL VALIDATION METRICS

The relative error between auto-generated and CAD-based solid models can be expressed as Equation 4

$$\text{Relative error} = \frac{\Delta_i^{TLS} - \Delta_i^{CAD}}{\Delta_i^{CAD}} \times 100 \qquad\qquad \text{Equation 4}$$

where superscripts TLS and CAD respectively indicate solid models from the TLS data and CAD models.

For determining accuracy levels of façade openings and opening positions, a validation metric was used for local features. A standard uncertainty and error bounds are determined by means of a statistical procedure. Deviations for the confidence interval of different geometric models are established. Oberkampf and Barone (Oberkampf and Barone 2006) proposed use of confidence intervals for validation between computational and experimental work as devised by Devore (2000), as summarized below.

 Assuming a random sample $X_1, X_2, \ldots, X_n$ that is from a normal distribution with a mean (μ) and a standard deviation (σ) has the actual sample observations $x_1, x_2, \ldots, x_n$. Then, for any a number of samples (n), the sample means ($\overline{X}$) is normally distributed, with a mean and standard deviation $\sigma/\sqrt{n}$. From that, the variable Z is expressed as in Equation 5.

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} \qquad\qquad \text{Equation 5}$$

This has an approximate standard normal distribution. For a large sample, the Central Limit Theorem implies that $\overline{X}$ has a normal distribution irrespective of the nature of the population distribution. Then, the variable Z can be written as Equation 6

$$P(z_{-\alpha/2} < Z < z_{+\alpha/2}) \approx 1-\alpha \qquad \text{Equation 6}$$

where $z_{\alpha/2}$ is the value of the random variable Z, at which the integral of Z from $z_{\alpha/2}$ to $+\infty$ is $\alpha/2$, and $z_{-\alpha/2}$ is analogue to $z_{\alpha/2}$. The total area of both tail intervals of the distribution is α.

In practical computation of the interval, the value of σ will almost never be known and must be replaced by the sample standard deviations. Substituting into Equation 5 and then re-arranging the Equation 6 allows that the expression to be written as Equation 7

$$P\left(\overline{X} - z_{\alpha/2}\frac{s}{\sqrt{n}} < \mu < \overline{X} + z_{\alpha/2}\frac{s}{\sqrt{n}}\right) \approx 1-\alpha \qquad \text{Equation 7}$$

which can be rewritten as a confidence interval (i.e., a probability interval) for the population mean using sampled quantities for the mean and standard deviation as in Equation 8

$$\mu \approx \left(\overline{x} - z_{\alpha/2}\frac{s}{\sqrt{n}}, \overline{x} + z_{\alpha/2}\frac{s}{\sqrt{n}}\right) \qquad \text{Equation 8}$$

where $\overline{x}$ and s are the sample mean and standard deviation, respectively, based on n observations. Note that $\overline{x}$ and s are computed from the realization $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$. Generally speaking, for the condition $n > 40$ the population is sufficient to justify the use of this interval (Devore 2000). Consider the case of estimating a confidence interval for an arbitrary

number of experimental observations (n), with n as small as two. The interval analogous to Equation 8 is shown in Equation 9

$$\mu \approx \left(\overline{x} - t_{\alpha/2} \frac{s}{\sqrt{n}}, \overline{x} + t_{\alpha/2} \frac{s}{\sqrt{n}} \right)$$ Equation 9

where the confidence level is given by $100(1-\alpha)\%$ and $t_{-\alpha/2,v}$ is $1-\alpha/2$ quartile of the t-distribution for $v = n-1$ degree of freedom. For n greater than 16, the cumulative t distribution and the cumulative standard normal distribution differ by less than 0.01 for all quantities. In the limit as $n \to \infty$, the t-distribution approaches the standard normal distribution (Devore 2000).

To quantify any inaccuracies of the resultant geometries, absolute errors between the algorithm based models and the CAD based ones, parameters of interest (e.g. window length and height) can be assumed as random variables as $x_1, x_2, \ldots, x_n$ (Equation 10). A mean and standard deviation of the quantity of interest can be given by Equation 11 and 12, respectively.

$$x_i = \Delta_i^{TLS} - \Delta_i^{CAD}$$ Equation 10

$$\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$ Equation 11

$$s = \sqrt{\frac{\sum_{i=1}^{n} \left(x_i - \overline{x} \right)^2}{n-1}}$$ Equation 12

where n is a number of singular nodes of interest.

Additionally, measurement results can be experimentally evaluated by a statistical method through application of type A standard uncertainty according to the 1998 ISO Guide (ISO Guide 1998). The standard uncertainty estimate uncertainty of the standard deviation of the absolute errors generated during the reconstruction process is given by Equation 13

$$u = \frac{s}{\sqrt{n}}$$ Equation 13

REFERENCES

Agoston, M.K. (2005). *Computer Graphics and Geometric Modeling: Implementation & Algorithms,* Springer Verlag London Limited.

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

Ayala, D., Brunet, P., Juan, R., and Navazo, I. (1985). "Object representation by mean of nominal division quadtrees and Octrees." *ACM Transaction on Graphics*, 4(1), 41-59.

Barber, C.B., Dobkin, D.P., and Huhdanpaa, H. (1996). "The Quickhull Algorithm for Convex Hulls." *ACM Transactions on Mathematical Software*, 22(4), 469-483.

Becker, S., and Haala, N. (2007). "Refinement of Building Facades by Integrated Processing of LIDAR and Image Data." *PIA07 - Photogrammetric Image Analysis*, Munich, Germany, September 19-21, 2007, 7-12.

Becker, S., and Haala, N. (2009). "Grammar Supported Facade Reconstruction From Mobile Lidar Mapping." *CMRT09*, Paris, France, September 3-4, 2009, 229-234.

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. (1999). "The ball-pivoting algorithm for surface reconstruction." *Visualization and Computer Graphics, IEEE Transactions on*, 5(4), 349-359.

Boulaassal, H., Chevrier, C., and Landes, T. (2010). "From Laser Data to Parametric Models: Towards an Automatic Method for Building Façade Modelling." *Digital Heritage: Third International Euro-Mediterranean Conference*, Lemessos, Cyprus, November 8-13, 2010, 42-55.

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), 1-20.

Brilakis, I., German, S., and Zhu, Z. (2011). "Visual Pattern Recognition Models for Remote Sensing of Civil Infrastructure." *Journal of Computing in Civil Engineering*, doi:10.1061/(ASCE)CP.1943-5487.0000104.

Collins, B.D., and Sitar, N. (2004). "Application of High Resolution 3D Laser Scanning to Slope Stability Studies." *Proceeding of 39th Symposium on Engineering Geology and Geotechnical Engineering*, Butte, MT, May 18-21, 2004, 79-92.

Curless, B., and Levoy, M. (1996). "A Volumetric Method for Building Complex Models from Range Images." Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, New Orleans, LA, USA, August 04 - 09, 1996, 303-312.

Dalmasso, P., and Nerino, R. (2004). "Hierarchical 3D Surface Reconstruction Based on Radial Basis Functions." *Proceedings of 2nd International Symposium on 3D Data Processing, Visualization and Transmission.*, Thessaloniki, Greece, September 6-9, 2004, 574-579.

Devore, J.L. (2000). *Probability and Statistics for Engineering and the Sciences,* Statistics Series, Duxbury.

Dublin City Council. Planning Application. Available at: http://www.dublincity.ie/Planning/PlanningApplication/Pages/Planning%20Application.aspx. Accessed on: Dec 20, 2010. (Access on: Dec 20, 2010)

Edelsbrunner, H., and Mücke, E.P. (1994). "Three-dimensional alpha shapes." *ACM Transactions on Graphics (TOG)*, 13(1), 43-72.

Guarnieri, A., Vettore, A., and Pontin, M. (2005). "A Volumetric Approach for 3D Surface Reconstruction." CIPA 2005 XX International Symposium, Torino, Italy, September 26-October 01, 2005.

Hinks, T. (2011). "Geometric Processing Techniques for Urban Aerial Laser Scan Data," PhD thesis, University College Dublin.

Hinks, T., Carr, H., Laefer, D.F., Morvan, Y., O'Sullivan, C., Truong-Hong, L, Ceribasi, S. (2009). Robust Building Outline Extraction. Patent PTO 56793223.

Hoffman, E.S., Gustafson, D.P., Gouwens, A.J., and Rice, P.F. (1996). *Structural Design Guide to the AISC (LRFD) Specification for Buildings,* Second edition, Chapman and Hall, London.

ISO Guide. (1998). "Uncertainty of Measurement - Part 3: Guide to The Expression of Uncertainty in Measurement (GUM:1995)." International Organization for Standardization (ISO).

Jordan, C. (1887). "Cours D'analyse L'École Polytechnique." *Paris*, 587-594 (French).

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

Laefer, D.F. Truong-Hong, L., Cording, E.J., Long, J., & Erkal, A. (2011b). "Manufacturing, assembly, and testing of scaled, historic masonry for one-gravity, pseudo-static, soil-structure experiments" Construction and Building Materials. Special Edition "Masonry Research and Practice" Elsevier, DOI 10.1016/j.conbuildmat.2011.03.066

Lee, S.C., and Nevatia, R. (2004). "Extraction and Integration of Window in a 3D Building Model from Ground View Images." *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Washington, D.C., USA, June 27-July 02, 2004, 113-120.

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

Mayer, H., and Reznik, S. (2005). "Building Façade Interpretation From Image Sequences." CMRT05. IAPRS, R. F. Stilla U, Hinz S, ed., Vienna, Austria, 29-30 August, 2005, 55-60.

Meagher, D. (1982). "Geometric Modeling Using Octree Encoding." *Computer Graphics and Image Processing*, 19(2), 129-147.

44

Oberkampf, W.L., and Barone, M.F. (2006). "Measures of Agreement between Computation and Experiment: Validation Metrics." *Journal of Computational Physics*, 217(1), 5-36.

Okorn, B.E., Xiong, X., Akinci, B., and Huber, D. (2010). "Toward Automated Modeling of Floor Plans." *Proceedings of the Symposium on 3D Data Processing, Visualization and Transmission*, Espace Saint Martin, Paris, France, May 17-20, 2010.

Pighin, F., and Lewis, J.P. (2007). "Practical least-squares for computer graphics." *ACM SIGGRAPH 2007 courses*, 1-57.

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. (2009). "Knowledge Based Reconstruction of Building Models from Terrestrial Laser Scanning Data." *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6), 575-584.

Pulli, K., Duchamp, T., Hoppe, H., McDonald, J., Shapiro, L., and Stuetzle, W. (1997). "Robust Meshes from Multiple Range Maps." *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, Ottawa, Canada, May 12-15, 1997, 205-211.

Ripperda, N. (2008). "Determination of Facade Attributes For Facade Reconstruction." *ISPRS Congress, Proceedings of Commission III*, Beijing, China, July 3-11, 2008, 285-290.

Ripperda, N., and Brenner, C. (2009). "Application of a Formal Grammar to Facade Reconstruction in Semiautomatic and Automatic Environments." 12th AGILE International Conference on Geographic Information Science, Leibniz Universität Hannover, Germany, June 2-5, 2009.

Tang, P., Huber, D., and Akinci, B. (2009a). "Characterization of Laser Scanners and Algorithms for Detecting Flatness Defects on Concrete Surfaces." *Journal of Computing in Civil Engineering*, 25(1), 31-42.

Tang, P., Akinci, B., and Huber, D. (2009b). "Quantification of Edge Loss of Laser Scanned Data at Spatial Discontinuities." *Automation in Construction* 18(8), 1070-1083.

Tang, P., Huber, D., and Akinci, B. (2007). "A Comparative Analysis of Depth-Discontinuity and Mixed-Pixel Detection Algorithms." *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling* Montreal, Quebec, Canada, August 21-23, 2007, 29-38.

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

Truong-Hong, L., and Laefer, D.F. (2008). "Micro vs. Macro Models for Predicting Building Damage Underground Movements." International Conference on Computational Solid Mechanics - CSM2008, Hochiminh City, Vietnam, 241-250.

Wang, J., Oliveira, M. M., Xie, H., and Kaufman, A. E. (2005). "Surface Reconstruction Using Oriented Charges." Computer Graphics International, Stony Brook, New York, USA, June 22-24, 2005, 122-128.

Wang, R., Bach, J., and Ferrie, F. P. (2011). "Window Detection from Mobile LiDAR Data." *IEEE Workshop on Applications of Computer Vision*, Kona, Hawaii, January 5-7, 2011, 58-65.

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 04-07, 1998, 917-924.

Wonka, P., Wimmer, M., Sillion, F., and Ribarsky, W. (2003). "Instant Architecture." *ACM SIGGRAPH 2003*, San Diego, California, July 27-31, 2003, 669-677.