TUDelft

Delft University of Technology

A framework to extract structural elements of construction site from laser scanning

Truong-Hong, L.; Lindenbergh, R.C.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

A FRAMEWORK TO EXTRACT STRUCTURAL ELEMENTS OF CONSTRUCTION SITE FROM LASER SCANNING

L. Truong-Hong [1, *], R.C. Lindenbergh [1]

[1] Dept. of Geoscience & Remote Sensing, Delft University of Technology, Delft, The Netherland, - (l.truong; r.c.lindenbergh)@tudelft.nl

Commission II, WG II/10

KEY WORDS: Point cloud, Feature Extraction, Cell-Patch-Based Segmentation, Scan to BIM, Structural Elements, Dimensional Quality Control, Surface Defect,

ABSTRACT:

This paper proposes a framework to automatic extract structural elements of reinforced concrete buildings from laser scanning data, which can be used in dimensional quality control and surface defect identification. The framework deploys both spatial information of a point cloud and contextual knowledge of building structures to extract the structural elements in a sequential order: floors and ceilings, walls, columns and beams. The method starts to extract a subset data containing candidate points of the structural elements and segmentation methods and filtered based contextual knowledge subsequently apply to obtain the final points of the elements. In this framework, a combination between kernel density estimation and a cell-patch-based region growing are to extract the floors, ceilings and walls, while the points of the columns and beams are achieved through a voxel-based region growing. 23.5 million data points of one story of the building is used to test a performance of the proposed framework. Results showed all structural components are successfully extracted. Moreover, completeness, correctness, and quality indicated through point-based performance report larger than 96.0%, 96.9% and 93.0%, respectively while overlap rates of the floors, ceilings and walls are no less than 95.3%. Interestingly, an executing time of the proposed method is about 7.7seconds per a million point.

1. INTRODUCTON

In construction projects, defects and failures of structural components result to delay, increase a total cost and may cause personal injuries and fatalities in the worst case. For example, rework to fix defects can cost about 10% of the complete costs in civil infrastructure projects (Love Peter, 2002). To minimize this cost, project managers are increasingly searching methods to identify any defects timely and efficiently to establish an appreciate planning. However, in current practice pipelines, manual inspection with visualisation check, measuring tapes, levelling or total stations is often used at a construction site. This method exposes time consuming in acquiring geometric data. Moreover, as only discrete locations on the components' surfaces are measured, quantify defects cannot identified completely and results may not fully reflect an actual condition of the structure. Therefore, with the current inspection pipelines, project managers cannot control quality of the construction project efficiently.

Recently, a terrestrial laser scanner can capture three-dimensional (3D) visible surfaces of objects in high details accurately and efficiently. The unit has been widely used in many civil engineering applications (Anagnostopoulos et al., 2016; Cabaleiro et al., 2017), and recently in construction projects for example, monitoring construction progress (Bosché, 2010) and creating as-built building information modelling (BIM) (Jung et al., 2018). As visible surfaces of components in a field of view of the scanner are captured, an entire raw data of a construction project is massive and complex, which are restricted to identifying defects or reconstructing 3D geometric models of the structural components in automated manner. In practice, computer aided design programs (e.g. Revit, Autodesk Navisworks, Archicad, and Leica Cyclone 3DR) are often used to extract data points of edges, surfaces and components manually and subsequently as-built geometric primitive (e.g.

planes, cylinders, spheres or irregular shapes) are used to fit to the point cloud to obtain 3D models. This pipeline requires experienced users to handle and process the massive and complex data set with a powerful computer.

On another hand, a research community has efforts to develop automatic methods for geometric modelling of 3D building models from a point cloud (Jung et al., 2018; Laefer and Truong-Hong, 2017; Truong-Hong et al., 2012). Those methods are mostly extract and create 3D components of a building, and then classifies them into semantic elements like floors, ceilings, walls, and doors and windows (Thomson and Boehm, 2015). In those methods, floor and ceiling are often recognized through points located in large bins of a histogram generated from elevations of the point cloud (Jung et al., 2018). Moreover, common segmentation methods like region growing (Rabbani et al., 2006), Hough transform (Hough, 1959), and RANSAC (Schnabel et al., 207) are employed to extract the point clouds of vertical walls (Hulik et al., 2014; Jung et al., 2014; Thomson and Boehm, 2015; Xiong et al., 2013). Subsequently, the point cloud of the wall is projected on a 2D plane to extract edges of the doors and windows using image-based techniques (Jung et al., 2014).

Addition to reconstruct 3D building models, a work on processing a point cloud of a construction project has been also attracted researchers. For example, to assess quality of full-scale precast concrete element, Kim et al. (2014) mapped as-design model to the point cloud via key features like surfaces, edges and corners derived from a point cloud. Deviations of these features between two models are given a quality of construction element. Similarly, a one-class support vector machine approach was developed to extract rebars in reinforced concrete members for quality assurance and control, in which a set of features including linearity, planarity and red-green-blue colors of the point cloud were used (Wang et al., 2017). Moreover, in a goal to extract concrete structural elements (e.g. slabs, beams and columns)

---

* Corresponding author

from a point cloud, Son and Kim (2017) used color information of a point cloud to identify a region of interest appearing as the same construction material class. Next, an edge-based segmentation was employed based on a voxelization model of the region of interest created from a supervoxel algorithm. The segmentation refinement was implemented to discard small segments based on the segment size, while segment's features, for example linearity, planarity and direction, are used to classify the segment as column, beam/girder, floor/slab/wall, and other by using a support vector machine. Although the proposed method succeeded to extract the structural components, it required to huge manual labor work to prepare the training set and computational time can be up to 282 seconds per a million point. In a similar goal, Maalek et al. (2019) proposed a hierarchical method to extract structural elements (e.g. floors, ceiling/slab, columns and rebars) from a construction site. The method started to use a histogram generated from elevation of data points. Next, planar and linear features of each points computed from a robust principal analysis were used to cluster the points of surfaces. A set of adjacent surfaces in a form of a symmetric section and parallel to main directions of a building is known as the surfaces of a column. Finally, linear segments within the column's boundary were considered as rebars.

In summary, as complexity of the building construction project with high number of structural elements with different shapes, size and orientation and a large amount of the point cloud, there is still lack of efficient methods to straightforwardly process the data. Existing methods are requirement of an as-design model and/or certain assumption to extract the structural components and time consuming. Thus, this paper proposes a new method to automatically extract point clouds describing the building components from a building construction project.

2. PROPOSED FRAMEWORK

In a construction site, to avoid any construction delay, an inspection report on dimensional quality control and surface defects of structural elements in each construction stage must be approved timely. In a building construction project, structural elements of each story must be inspected before starting a new story, as such a point cloud of these components in one story is used as input data. The proposed framework consists of 3 consecutive moduli to extract: (i) floors, ceilings and walls, (ii) columns and (iii) beams (primary and secondary beams), as shown in Figure 1. This method is a synergy between a spatial information of a point cloud and contextual knowledge of structures in terms of minimum size, shape, orientation and relationship.
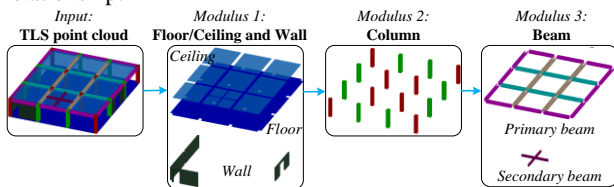

Figure 1. Workflow of the proposed framework

2.1 Modulus 1: Floor, ceiling and wall extraction

As floors, ceilings and walls mostly appear as planar surfaces, a cell-patch based segmentation is developed to segment a point cloud of these structural elements. The method consists of two main steps: (i) extract a patch (a local plane) of the surface and (2) segment those planar patches. In *Step 1*, the algorithm starts to employ a quadtree to decompose an input point cloud ($P = p_i \in R^3$) into 2D cells ($C = \{c_1, \ldots c_i, c_N\}, i = [1, N]$). For floors

and ceilings, an initial bounding box is recursively subdivided along the *x*- and *y*- directions in a Cartesian coordinate system. In this work, the cell size is used as a criterion to terminate the subdivision process, in which dimensions of the cells on leaf nodes are no larger than the predefined cell size (*cell_size*). Subsequently, the cell c_i is classified as an "*empty*" cell if it occupies the number of the points less than a predefined minimum number of the points (*min_ptc*); otherwise, the cell is known as the "*full*" cell. In this implementation, the *min_ptc* is set equal to 12 points, which implies at least 4 points are on each edge of the plane within the cell.

As the 2D cell often occupies a point cloud representing patches/planes of multiple structural elements along a depth direction of the cell, those patches can be extracted by using a kernel density estimation (KDE) generated from the coordinates of the points along the depth (Laefer and Truong-Hong, 2017) (Figure 2). Points within two consecutive minimum concave points are used to determine a patch $\psi_{ij}(p_{ij,0}, n_{ij})$, in which a robust principal component analysis (rPCA) is used to estimate a centroid $p_{ij,0}$ of the patch's points and the normal vector, n_{ij} (Laefer and Truong-Hong, 2017). Moreover, the residual value (r_{ij}) defined as a root mean square distances from the points to the patch $\psi_{ij}(p_{ij,0}, n_{ij})$ is also computed. Thus, each patch $\psi_{ij} \in c_i$ is described by a tuple ($p_{ij,0}, n_{ij}, r_{ij}$).

a) Points and KDE   b) Extracted planes   c) Extracted planes
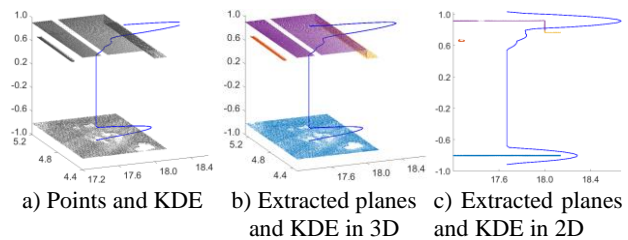and KDE in 3D   and KDE in 2D

Figure 2. Extract patches within a 2D cell

In *Step 2*, a cell-patch based region growing (CRG) is proposed to group patches ψ_{ij} of the floors, ceilings and walls, which consist of 3 sub-steps (i) cell-patch segmentation, (ii) patch filtering and (iii) point-patch region growing. In *Step 2.1*, a cell-patch segmentation cluster patches into a subset representing to the element's surface. The patch $\psi_{ij} \in c_i$ having the smallest residual value ($r_{ij} \rightarrow min$) is added to a region (R_k) and assigned as an initial seeding patch. Neighbouring patches ($\psi_{kl} \in c_k, k = 1$-8), where the cell c_k is adjoined to the cell c_i, are added to the region R_k if Eq. 1 is satisfied. Moreover, the adding patch ψ_{kl} is only considered as a seeding patch for a next iteration if its residual value r_{kl} is no larger than the residual threshold (r_0). The growing process is completed when all predefined patches are checked.

$$\begin{cases} \angle n_{ij}, n_{kl} \leq \alpha_0 \\ d(p_{kl,0}, \psi_{ij}(p_{ij,0}, n_{ij})) \leq d_0 \end{cases} \quad (1)$$

where $d(p_{kl,0}, \psi_{ij}(p_{ij,0}, n_{ij}))$ = Euclidean distance between the centroid point of the patch ψ_{kj} to the plane ψ_{ij}
     α_0 and d_0 = the angle and distance thresholds

As the plane is extracted from a 2D cell, the patch derived from KDE can contain data points of adjacent surface (Figure 2c). In this case, the patch can be segmented, or unsegmented and over- or under-segmentation is available, which can be solved through *Step 2.2* and *2.3*. In *Step 2.2*, a filtering algorithm aims to filter any points due to over-segmentation, which belong to the patches

on a boundary of the region. The algorithm first extracts patches ψ_{ij} on a boundary (denoted as $\psi_{ext,ij}$) of a region R_i. Second, neighbour patches ψ_{kl} of the patch $\psi_{ext,ij}$ are extracted and these neighbour patches can be interior patches $\psi_{int,kl} \in R_i$ and out-region patches $\psi'_{kl} \in R'_i$. Third, by assuming the patch $\psi_{int,ij}$ as a local surface of the region R_i, the points $p_{ij} \in \psi_{ext,ij}$ are considered as outlier points $p_{out,ij}$ if the distance $d(p_{ij}, \psi_{ext,ij}(p_{ij,0}, n_{ij}))$ are larger than the distance threshold d_0. The similar process is also to examine if the outlier points, $p_{out,ij}$, can be added to the region R'_i. From the patch $\psi'_{kl} \in R'_i$, the adjacent interior patches $\psi'_{int,kl} \in R'_i$ are extracted, and any points $p'_{ij} \in p_{out,ij}$ is added to the region R'_i, if the distance $d(p'_{ij}, \psi'_{int,kl}(p_{kl,0}, n_{kl}))$ is no larger than the distance threshold d_0.

Step 2.3, a point-patch region growing, starts with an exterior plane ($\psi_{ext,ij}$) of the region R_i *to* search adjacent, unsegmented patch $\psi_{mn} \in c_m$. Points $p'_{mn} \subseteq p_{mn}$ of the patch ψ_{mn} are added to the region R_i if the distance $(p'_{mn}, \psi_{ext,ij}(p_{ij,0}, n_{ij}))$ is no larger than the distance threshold d_0. Subsequently, adding points p'_{mn} can be used as a new exterior patch for the next iteration if a ratio $|p'_{mn}|/|p_{mn}|$ is large than 0.5.

After the floors and ceilings are completed extracted, the points of these structural elements are immediately discarded. The remaining data points are used to extract vertical walls through a similar procedure presented in *Step 1* and *2*. However, in this case, the quadtree subdivides the data points along the *x-* and *z-*, and *y-* and *z-*directions to extract the walls in *xz* and *yz* planes, respectively. Details of the floor, ceiling and wall extraction can refer to Truong-Hong and Lindenbergh (2020). Notably, if the points are assigned to the building components, these points are immediately discarded.

2.2 Modulus 2: Column Extraction

In concrete buildings, columns are vertical elements connecting the floor and ceiling. A cross-section of the concrete column is often symmetric, rectangle or square shape. In this Modulus, unassigned points within the 2D cells in *xy* plane are used to extract the column extraction through two steps (Figure 3).

Step 1 - rough extraction, starts to compute features of each cell c_i, consisting of the cell height (H_{ci}) and the maximum gap (ΔH_{ci}) (Eq. 2 and 3).

$$H_{ci} = max(p_{ci}.z_j) - min(p_{ci}.z_j) \qquad (2)$$
$$\Delta H_{ci} = max(p_{ci}.z_j - p_{ci}.z_{j+1}) \qquad (3)$$

where $\quad p_{ci}.z_j$ = z-coordinate of a point p_{ci}

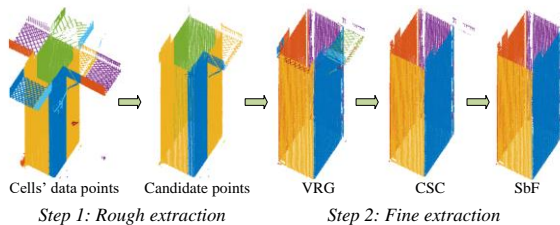*Step 1: Rough extraction*   *Step 2: Fine extraction*
Figure 3. Illustration of Column extraction

Next, as the point cloud of the column distributes along a vertical direction from the floor to the ceiling, if the cell contains the data points of the column, the cell must satisfy Eq. 4.

$$c_i \rightarrow c_{col,i} \; if \; H_{ci} \geq 0.75H_0 \; and \; \Delta H_{ci} \leq \Delta H_0 \qquad (4)$$

where $\quad H_0$ = the building storey height is an average distance of the storey height computed from patches within the cells, $d(p_{ik,0}, \psi_{ij}(p_{ij,0}, n_{ij}))$, in which $\psi_{ij}(p_{ij,0}, n_{ij}) \in c_i$ and $\psi_{ik}(p_{ik,0}, n_{ik}) \in c_i$ respectively represent to the floor and ceiling.

$\Delta H_0 = 0.25H_0$, which is introduced to overcome a missing data issue due to occlusion or obstruction.

Subsequently, the cell connectivity is employed to group the cells, $c_{col,i}$, which share vertices and edges. Each cluster represents to the column (Figure 3), but the cluster may contain points of other elements, like beams.

Step 2 - fine extraction, aims to group point clouds of the columns' surfaces. The method adopts a voxel-based growing segmentation (VRG) (Vo et al., 2015) to extract the data point of each column's surface. The VRG method operates similar the CRG method in *Modulus 1*, but 3D voxels are used instead of the 2D cells in the CRG. The VRG employs an octree to subdivide the point cloud of the column cluster into the small voxel with the voxel size is no larger than the predefined voxel size (*voxel_size*). Subsequently, the points within the voxel is assumed to describe a plane, and the rPCA is employed to estimate features of each voxel v_i through a fitting plane, which consist of a centroid ($p_{v0,i}$) and a normal vector (n_{vi}) and residual (r_{vi}). Figure 3 shows a progress of the column extraction.

Next, irrelevant surfaces which belong to other components can be eliminated using a connected surface component (CSC) algorithm (Figure 3). As the column' surfaces connect in a form of a close loop and the column's cross-section is symmetric, the CSC algorithm starts with the initial segment to iteratively search connected surfaces until no more surface is found (Figure 4). In this work, the largest surface in a term of an area is set as the initial surface. The surfaces S_i connects to the surface S_j if an angle $\angle n_i n_j$ is larger than the angle threshold α_l, and the overlap length ($|P_{i2}P_{j1}|$) is no less than $0.5H_0$ (Figure 4a). Notably, only data points of the surfaces S_i and S_j within a buffer equal to a half of a minimum width of the column from the intersection line L_{ij} are projected onto the line L_{ij} to determine line segments $P_{i1}P_{i2}$ and $P_{j1}P_{j2}$. When the link of connected surfaces is built, the surface is to be removed out of the link if any surface cannot make a close form (Figure 4b and c).

a) Identify an intersection between two adjacent surfaces

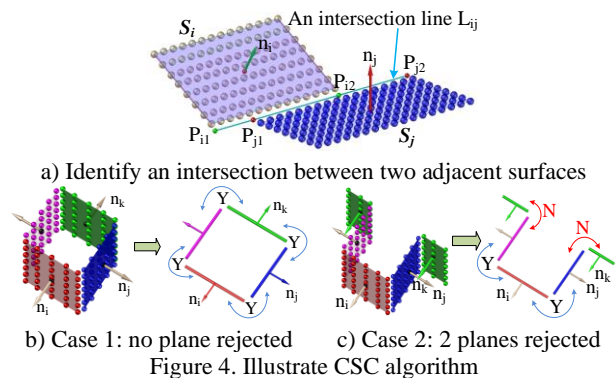b) Case 1: no plane rejected   c) Case 2: 2 planes rejected
Figure 4. Illustrate CSC algorithm

Finally, the surface's points are bounded by intersection lines with the adjoined surfaces, and any points outside the boundaries can be eliminated by using a surface-based filtering (*SbF*) algorithm. from the points $p_i \in S_i$, sign distances $d(p_i, L_{ij})$ to the intersection line L_{ij} are used to classified the points into two groups based on the sign (+ or -), and the small group in a term of the number of the points is outlier.

2.3 Modulus 3: Beam extraction

This work is limited to extract a primary beam connected between two columns or a column and a wall. The algorithm consists of two steps (Figure 5).
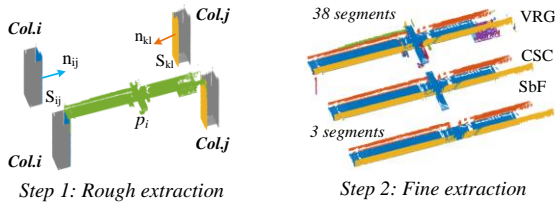

Step 1: Rough extraction   Step 2: Fine extraction

Figure 5. Extract the points of the primary beams

Step 1 starts with a surface $S_{ij}(p_{ij,0}, n_{ij})$ of a column Col_i to search the closest, parallel surface $S_{kl}(p_{kl,0}, n_{kl})$ of a column Col_k, which based on the deviation of the normal vectors and the distance between two surfaces (Eq.5).

$$\begin{cases} \angle n_{ij}, n_{kl} \leq \alpha_1 \\ d(p_{kl,0}, S_{ij}(p_{ij,0}, n_{ij})) \to min \end{cases} \quad (5)$$

where α_1 = the angle threshold

Subsequently, candidate points (p_i) of a beam B_i connected between the columns Col_i and Col_k can be extracted (Eq. 6), as shown in Figure 5a. Notably, if the surface S_{kl} is not available, the distance $d(p_{kl,0}, S_{ij}(p_{ij,0}, n_{ij}))$ is set as an infinity, and only $S_{ij}.w$ is used in the second condition.

$$\begin{cases} 0 \leq d\left(p_i, S_{ij}(p_{ij,0}, n_{ij})\right) \leq d(p_{kl,0}, S_{ij}(p_{ij,0}, n_{ij})) \\ |d(p_i, L_{jl})| \leq max(S_{ij}.w, S_{kl}.w) + tol \end{cases} \quad (6)$$

where $d(p_i, L_{jl})$ = a sign distance from the points p_i to the line L_{ij} defined by a projection of $p_{ij,0}$ $p_{kl,0}$ on the *xy* plane

$S_{ij}.w$ = the width of the surface S_{ij}

$S_{kl}.w$ = the width of the surface S_{kl}

tol = a tolerance to overcome data errors, $tol = 0.1m$

Once the candidate points of the beams are extracted, a *Step 2* in *Modulus - Column Extraction* are used to extract and filter the final surface of the column. Notably, in the CSC algorithm, the bottom surface of the beam is set an initial surface while the threshold for determining an overlap length is set equal to a half of the distance between two surfaces S_{ij} and S_{kl} (Figure 5b).

3. EXPERIMENT AND RESULTS

A carpark (18.5m wide x 29.5m long x 3.45m high) on a basement of a building on Pham Ngu Lao st., Vietnam consists of the concrete floor, ceiling, walls, columns and beams and parts of a MEP system is selected as a case study was scan by a Trimble TX8 (Trimble, 2020b), with a point spacing of 11.3mm at a range of 30m and a total of 11 scanning stations (Figure 6). The point clouds were registered by the Trimble RealWork software v11.2 (Trimble, 2020a) with the registration error about 1.57mm, and 23.5 million points with x-, y- and z- coordinates was used to test this proposed method.

In this experiment, *cell_size* = 1.0m and *voxel_size* = 0.1m selected are to ensure at least one cell or voxel can be generated from the smallest slab by 2mx2m and the smallest cross-section of the column and beam by 0.2mx0.2m. Moreover, the chosen bandwidth (*bw*) by 0.1m allows to separate two surfaces of the

thinnest component like the slab or the wall with the smallest thickness by 0.2m. As the cell or voxel is used to represent the data of the building or component, the cell or voxel can contain the points of two adjacent surfaces, the selected angle threshold $\alpha_0 = \alpha_1 = 5$ degrees is to prevent over-segmentation, particularly when the MEP components are available. Notably, the distance and residual thresholds, $d_0 = 10mm$, $r_0 = 10mm$ are empirically selected nearly equal the sampling step but can adjust based on a data error budget and the surface roughness.
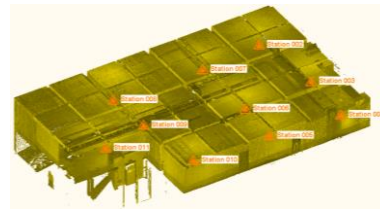

Figure 6. A point cloud of a basement of the building

a) Floors

b) Slabs of a ceiling
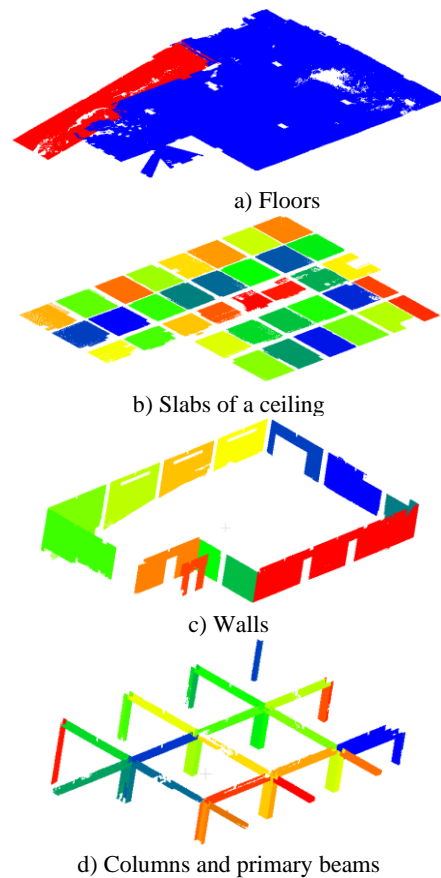

c) Walls

d) Columns and primary beams

Figure 7. Results of structural element extraction

As only a basement story of a building was scanned from inside, the first and last patches of a cell in the vertical direction are possible parts of floors and ceilings, respectively (Figure 2). As such, these patches are respectively used as seeding patches in floor and ceiling extraction by using the cell-patch-based region growing in Modulus 1. However, other patches of the cells use in *Step 2.2* and *2.3* to refine the segmentation results. Similarly, outmost patches of the cell in *xz* and *yz* planes are used as seeding patches for wall extraction. Notably, data points are immediately deactivated when they were assigned to the structural components and remaining points are used in subsequent process. Figure 7 shows results of structural elements extracted from the proposed method.

Visualisation evaluation reports all surfaces of the structural elements (floors and ceilings, columns and beams) successfully extracted. Moreover, to measure the location deviation, quality indicators including a true positive (*TP*), false positive (*FP)* and false negative (*FN*) use to identify a difference between extracted components and ground truth. A point-based performance is used to determine these indicators. Subsequently, evaluation quantities can be interpreted through completeness, correctness, and quality. Notably, definition of those indicators can refer to Truong-Hong and Laefer (2015) and the ground truth was manually extracted from the point cloud of the case study. Resulted evaluation of the location deviation are shown in Table 1, in which the proposed method can extract the points of the structural components with completeness, correctness and quality no smaller than 96.0%, 96.9% and 93.0%, respectively. Moreover, in the best case, the quality quantity can reach 98.6% for extracting the floor.

Table 1. Resulted evaluation

Component	Completeness	Correctness	Quality
Floor	98.9%	99.7%	98.6%
Ceiling	96.0%	96.9%	93.0%
Walls	99.0%	99.3%	98.3%
Column	97.6%	99.0%	96.7%
Primary Beam	97.8%	98.6%	96.5%

In addition, to evaluate a shape similarity between the extracted component and the ground truth is measure through surfaces' areas in terms of area difference and overlap rate (Truong-Hong and Laefer, 2015). The floors, ceilings and walls are selected to measure the shape similarity because these components are simple, planar surfaces, which can simply to determine the surface parameters, while the other components (e.g. columns and beams) are required more complicated algorithms to obtain their shapes, which is out of the scope of this study. First, the point cloud of the structural element's surface ($p_i \in S_i$) is projected onto a fitting surface (Figure 8a). Next, a boundary points ($p_{ext,i}$) of the surface is extracted from the projected points by employing the alpha shape algorithm (Edelsbrunner et al., 1983), in which the radius threshold is set equal to *cell_size* = 1.0m (Figure 8b). Subsequently, the polygon representing the surface is created from the boundary points $p_{ext,i}$ (Figure 8c).
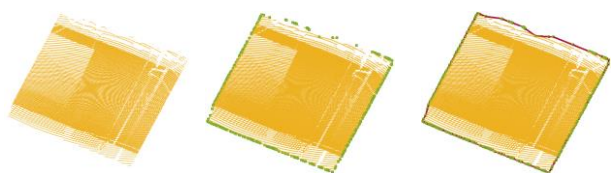

a) Data points of a surface b) Boundary points of a surface c) Polygons describing a surface
Figure 8. Create a polygon describing the element's surface

Finally, the surface area is computed from the polygon of the surface, and overlap area is determined as an intersection area between the polygons of the components' surfaces from the ground truth and the proposed method. Table 2 summarizes of a shape similarity from the floor, ceiling and walls appearing as the planar surfaces. Results show that the proposed method extracts the components (e.g. floor, ceiling and wall) differing ones from the ground truth no more than 5%, in which the lowest overlap rate is 95.3% for the ceiling.

Table 2. Summarized shape similarity

Component	Area of a component's surface (m²)			Overlap rate
	The proposed method	Ground truth	Overlap Area	
Floor	481.74	496.20	479.59	96.7%
Ceiling	368.57	373.96	356.26	95.3%
Walls	325.44	329.16	323.38	98.2%

In both evaluation quantities, it shows that the ceilings get a lower extracted performance comparing to other components. That is because a MEP system available causes an obstruction, which lead to a big gap in the point cloud of the slab's surface (Figure 9). The proposed method cannot extract the entire point cloud of the surface (Figure 9).
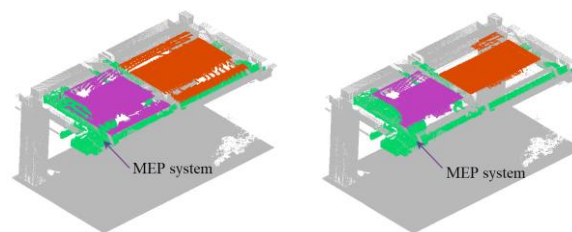
a) A ground truth   b) The proposed method
Figure 9. Overlaid data points of the slab on input data

A computational performance shows the proposed method takes a total of 180.2 seconds (equivalent to 7.7 seconds per 1 million points) to extract all floors, ceilings, columns and primary beams, which includes 146.5 seconds for floors and ceilings, 26.4 seconds for walls, 3.6 seconds for columns and 3.7 seconds for beams. Arguably, this indicates the proposed method is outperformance compared to the work of Son and Kim (2017), which required about 282 seconds per a million point, although both methods can achieve nearly the same accuracy. This efficiency is due to on the proposed method introduces a rough object extraction step to extract relevant point cloud from an entire data, and only a small subset with less complexity is subsequently used in extracting structural components. With this computational performance, it can prove that the proposed method can process a large data set of a construction project efficiently. Notably, this performance is based on an implementation of the proposed method in MATLAB 2019b (2019b) and processing on Dell Precision Workstation with a main system configuration as follows: Intel(R) Xeon(R) W-2123 CPU @ 3.6GHz with 32GB RAM.

4. CONCLUSIONS

The paper proposes a framework to automatically extract structural elements of reinforced buildings captured during construction. In this framework, both spatial information of a point cloud and fundamental contextual knowledge of structural elements are used to extract a point cloud of the elements in a sequential order from floors, ceilings, walls, columns, and primary beams. The proposed method is first to extract a subset data that is relevant desired structural elements, and then the segmentation and filtering algorithms are used to obtain the final point clouds of the elements' surface. A basement of a building, which consists of the floors, ceilings, walls, columns and beams, was scanned with a total of 23.5 million data points used to evaluate performance of the proposed framework. A resulted test showed that all the structural elements are successfully extracted with the completeness, correctness and quality larger than 96.0%, 96.9% and 93.0%, respectively when the point-based

performance is used to indicate evaluation quantities. Moreover, a shape similarity shows the floor, ceiling and wall are extracted with an overlap rate over 95.3%. Interestingly, the proposed method is required average 7.7seconds to process a million point. Although all structural elements are successfully extracted, additional tests with different types of the buildings should be carried to investigate a robustness of the proposed method. Moreover, the proposed method will also extend to adopt for creating as-built BIM.

ACKNOWLEDGEMENTS

REFERENCES

Anagnostopoulos, I., Pătrăucean, V., Brilakis, I., Vela, P., 2016. Detection of Walls, Floors, and Ceilings in Point Cloud Data. *Construction Research Congress 2016*, 2302-2311.

Bosché, F., 2010. Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics* 24, 107-118. doi.org/10.1016/j.aei.2009.08.006

Cabaleiro, M., Lindenbergh, R., Gard, W.F., Arias, P., van de Kuilen, J.W.G., 2017. Algorithm for automatic detection and analysis of cracks in timber beams from LiDAR data. *Construction and Building Materials* 130, 41-53. doi.org/10.1016/j.conbuildmat.2016.11.032

Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory* 29, 551-559. 10.1109/TIT.1983.1056714

Hough, P.V., 1959. Machine analysis of bubble chamber pictures, Conf. Proc., pp. 554-558.

Hulik, R., Spanel, M., Smrz, P., Materna, Z., 2014. Continuous plane detection in point-cloud data based on 3D Hough Transform. *Journal of Visual Communication and Image Representation* 25, 86-97. doi.org/10.1016/j.jvcir.2013.04.001

Jung, J., Hong, S., Jeong, S., Kim, S., Cho, H., Hong, S., Heo, J., 2014. Productive modeling for development of as-built BIM of existing indoor structures. *Automation in Construction* 42, 68-77. doi.org/10.1016/j.autcon.2014.02.021

Jung, J., Stachniss, C., Ju, S., Heo, J., 2018. Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM. *Advanced Engineering Informatics* 38, 811-825. doi.org/10.1016/j.aei.2018.10.007

Kim, M.-K., Sohn, H., Chang, C.-C., 2014. Automated dimensional quality assessment of precast concrete panels using terrestrial laser scanning. *Automation in Construction* 45, 163-177. doi.org/10.1016/j.autcon.2016.08.035

Laefer, D.F., Truong-Hong, L., 2017. Toward automatic generation of 3D steel structures for building information

modelling. *Automation in Construction* 74, 66-77. doi.org/10.1016/j.autcon.2016.11.011

Love Peter, E.D., 2002. Influence of Project Type and Procurement Method on Rework Costs in Building Construction Projects. *J. Constr. Eng. M.* 128, 18-29.

Maalek, R., Lichti, D.D., Ruwanpura, J.Y., 2019. Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction. *Remote Sensing* 11, 1102. doi.org/10.3390/rs11091102

MathWorks, 2019b. MATLAB Function Reference., 2019b ed.

Rabbani, T., Heuvel, F.v.d., Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint, Int. *Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, pp. 248-253.

Schnabel, R., Wahl, R., Klein, R., 207. Efficient RANSAC for point-cloud shape detection, Computer graphics forum, pp. 214-226.

Son, H., Kim, C., 2017. Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity. *Advanced Engineering Informatics* 34, 114-124. doi.org/10.1016/j.aei.2017.10.001

Thomson, C., Boehm, J., 2015. Automatic Geometry Generation from Point Clouds for BIM. *Remote Sensing* 7, 11753-11775. doi:10.3390/rs70911753

Trimble, 2020a. Trimble RealWorks v11.2. *Trimble*.

Trimble, 2020b. Trimble TX8 LASER SCANNER. *Trimble*.

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

Truong-Hong, L., Laefer, D.F., Hinks, T., Carr, H., 2012. Combining an angle criterion with voxelization and the flying voxel method in reconstructing building models from LiDAR data. *Computer-Aided Civil and Infrastructure Engineering* 28, 112-129. doi:10.1111/j.1467-8667.2012.00761.x

Truong-Hong, L., Lindenbergh, R., 2020. Quantitative assessment of structural components for construction management using laser scanning data, FIG Working Week 2020, Amsterdam, the Netherlands, p. 15.

Vo, A.-V., Truong-Hong, L., Laefer, D.F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS J Photogramm Remote Sens* 104, 88-100. doi.org/10.1016/j.isprsjprs.2015.01.011

Wang, Q., Cheng, J.C.P., Sohn, H., 2017. Automated Estimation of Reinforced Precast Concrete Rebar Positions Using Colored Laser Scan Data. *Computer-Aided Civil and Infrastructure Engineering* 32, 787-802. 10.1111/mice.12293

Xiong, X., Adan, A., Akinci, B., Huber, D., 2013. Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction* 31, 325-337. doi.org/10.1016/j.autcon.2012.10.006