Extracting structural components of concrete buildings from laser scanning point clouds from construction sites

Truong-Hong, L.; Lindenbergh, Roderik

DOI
10.1016/j.aei.2021.101490

Publication date
2022

Document Version
Final published version

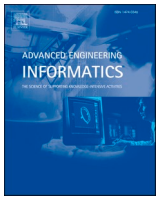
Published in
Advanced Engineering Informatics

Advanced Engineering Informatics 51 (2022) 101490

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei



Extracting structural components of concrete buildings from laser scanning point clouds from construction sites

L. Truong-Hong *, Roderik Lindenbergh

Department of Geoscience and Remote Sensing, Delft University of Technology, Delft, the Netherlands

ARTICLE INFO

Keywords:
Point cloud
Object extraction
Structure extraction
Segmentation
Inspection
Scan to BIM

ABSTRACT

In construction projects, inspection of structural components mostly relies on classical measurements obtained by measuring tapes, levelling, or total stations. With those methods, only a few points on the structure can be measured, and the resulting inspection may not fully reflect the actual, detailed condition of the complete object. Laser scanning is an emerging remote sensing technology to accurately and quickly capture surfaces of structures in high details. However, because of the complex, massive point cloud data acquired at a construction project, in practice, data processing is still manual work with computer aided programs. To improve upon current workflows, this paper proposes a method to automatically extract point clouds of individual surfaces of structural components of a concrete building, which subsequently can be used to inspect construction quality based on geometric information of the surfaces. The proposed method explores both spatial point cloud information and contextual knowledge of structures (e.g., orientation or shape) derived from building design specifications and practice. For extracting point clouds of surfaces of each structural component, the proposed method consists of 4 consecutive steps for extracting: (1) floors, ceiling slabs, and walls, (2) columns, and (3) primary and (4) secondary beams. Each step consists of two ingredients: (i) rough extracting the candidate points of the component and (ii) fine filtering of the surface points of the components via cell-based and voxel-based region growing segmentation (CRG and VRG) incorporating contextual knowledge of the structural members. Experimental tests on two different types of concrete buildings showed that the proposed method successfully extracts the structural elements, in which the completeness, correctness, and quality from the point-based evaluation are larger than 96.0%, 96.9%, and 92.0%, respectively. Moreover, the evaluation based on a shape similarity showed that the extracted floor, ceiling slab and wall overlap to the ground truth more than 92.5%.

1. Introduction

In construction projects, defects of structural components are inevitable. The rework costs would be minimized if any defect of the component can be identified at an early phase of the project. For example, Love [1] reported that rework to fix defects can amount to 10% of the complete costs in civil infrastructure projects, while Burati and Farrington [2] indicated that the construction costs can reach 16% of the total project costs when the defects were fixed at the last stage. In current practice, defect inspection is mostly manual interpretation of geometric data acquired from measuring tapes, levelling, or total stations. These methods are time-consuming in acquiring and interpreting geometric data [3,4], and inspection results do not reflect the complete actual condition of a structure because only discrete locations on surfaces of the components are measured. Therefore, project managers have difficulties in identifying defects timely and accurately, and making decisions based on objective results. In addition, inspection results in hard copies that are cumbersome to integrate to digital tools, which hampers the efficiency of project management.

Terrestrial laser scanning has the ability to capture visible surfaces

Abbreviations: BIM, Building Information Modelling; 3D, Three dimensional; 2D, Two dimensional; HT, Hough Transform; RANSAC, RANdom Sample Consensus; CRG, Cell-based region growing segmentation; VRG, Voxel-based region growing segmentation; PPRG, Patch-point region growing; CSC, Connected surface component; SbF, Surface-based filtering; PCA, Principal component analysis; rPCA, Robust principal component analysis; KDE, Kernel density estimation; PDS, Probability density shape; mbb, Minimum bounding box; PM, Proposed method; GT, Ground truth; TP, True possitive; TN, True negative; FN, False negative; Comp., Completeness; Corr., Correctness; Qual., Quality.

* Corresponding author.
E-mail addresses: l.truong@tudelft.nl (L. Truong-Hong), r.c.lindenbergh@tudelft.nl (R. Lindenbergh).

https://doi.org/10.1016/j.aei.2021.101490
Received 26 November 2020; Received in revised form 17 November 2021; Accepted 1 December 2021
Available online 11 December 2021

accurately, quickly, and efficiently. Laser scanning has been widely used in many civil engineering applications [5–8], and recently in construction projects, for example, for dimensional quality control [9], monitoring construction progress [10,11] and creating as-built building information modelling (BIM) [12]. As a scanned point cloud typically represents multiple objects in a scene, point cloud processing is often required to extract the point cloud of object surfaces of each object, which is a nontrivial task. In practice, users often use commercial software (e.g. Revit and FARO 3D Sense software) to manually extract rough data points of edges and surfaces of the components or of the component, and subsequently fit geometric primitives (e.g., planes, cylinders, spheres, or irregular shapes) applied to these points to obtain 3D models [13]. For example, FARO 3D Sense software [14] requires users to define two endpoints of the wall projected on the xy plane for creating wall models. This procedure is also required when PointFuse software [15] is used to extract data points of floors or walls. Similarly, users roughly crop a region surrounding a column and use plug-in fitting tools to subsequently create 3D column models [16]. Although software can create 3D models, users are still required to correct the final model. As such, the current pipeline based on commercial software is still time-consuming [13], requires experienced users to handle a massive, complex data set with a powerful computer, and produces 3D models with relatively low accuracy [17].

Additionally, a large amount of research groups has attempted to develop methods to automatically process point clouds for various applications, for example, to extract building structural elements [18,19], to create 3D building interiors [20–22], as-built building models [23–25], and to inspect and assess structural components [3,26]. However, due to the highly complex, massive data collected from a construction site, only few methods have been introduced to automatically extract the building elements [19,27,28]. As consequence, inspection of structural quality mostly relies on information of the structures obtained from a traditional survey. Moreover, these methods extract limited structures, for example, Maalek et al. [19,27] only extracted flat floor and ceilings, and columns. To leverage utilization of laser scanning point clouds and improve the workflow for point cloud processing for inspecting building components at a construction phase, this paper proposes a new method to extract the building components including floors, ceiling slabs, walls, columns, and beams. The proposed method uses spatial information of the point cloud and contextual knowledge of the building structure. In extracting each component, the method consists of two main steps called rough and fine extraction, respectively. In the first step, candidate points of a component of interest are roughly extracted based on the location, orientation, and relationship with adjoined components, while in the second step, segmentation methods are implemented to obtain the point clouds representing individual surfaces of the structural elements.

2. Related work

As a complex 3D scene of a building construction project includes objects of interest (OoI) and irrelevant objects, the OoI must be captured from different fields of views, which leads to dozens to hundreds of million points to be collected. Handling and processing those raw data sets to extract meaningful information are always huge challenges. Many segmentation methods have been developed to extract point clouds representing surfaces and/or geometric primitives including Hough Transform (HT) [29], RANdom Sample Consensus (RANSAC) [30] and region-growing [31]. These methods can only extract linear and flat planes from clean, low complexity data sets because these methods require appropriate and specific input parameters or thresholds. Parameter tuning is a nontrivial task because the data sets often represent various surfaces and objects of different shape and orientation, and the surfaces have different characteristics in terms of curvature, roughness, and size, for example. That implies these segmentation methods have difficulty in extracting data points of a variety of surfaces with a single set of input parameters or thresholds. Moreover, these methods also show low performance for a large data set or a complex geometry of the scene as the methods often compute salient features of all data points, although Vo et al. [32] introduced an octree-based region growing to reduce exhausted computation. This section is restricted to investigate recent methods extracting building components from a point cloud, and an extensive review on segmentation and geometric modelling can be found in [33,34].

The research community makes efforts to develop automatic methods for extracting 3D building components from a point cloud [7,12]. Those methods consist of two steps: extract and create 3D components of a building, and classify the detected component into semantic elements like floors, ceilings, walls, and doors and windows [35]. For building indoor reconstruction, floors and ceilings are often recognized through points located in large bins of a histogram generated from the elevations of a point cloud [12,23,36]. Moreover, common segmentation methods like HT [29], RANSAC [30] and region growing [31], are employed to extract point clouds of vertical walls [21,35,37,38]. Finally, the point cloud of the wall is projected on a 2D plane to extract edges of the doors and windows using image-based techniques [38]. For example, Zhang et al. [39] clustered candidate points of planes by using linear dependence of each point, while final points of the planes were extracted using maximum likelihood sample consensus (MLESAC) [40]. Murali et al. [41] used RANSAC to extract planes from point clouds of interior buildings, and then classified these planes as ceilings, floors, walls or clutters based on features of the plane including point density, height in a vertical direction, and orientation. Bassier et al. [42] used octree-based region growing to decompose point clouds of a building into a set of planar patches and a random forest classifier with geometric and contextual features to assign labels for the planes, which were known as beams, floors, ceilings, walls, roofs, and clutters. The study reported that building components were extracted with average recall and precision of 87% and 85% for all components but of 62% and 81% for beam elements. Interestingly, in reconstructing structural elements of Manhattan world buildings from point cloud, Brilakis et al. [43] used region growing-based void voxel starting from an empty space of a room to search point clouds of furthest planes known as floors, ceilings and walls, which were located between two adjacent void voxels. Similarly, by using region growing developed by Rabbani et al. [31], Fang et al. [20] classified extracted planes of indoor buildings as floors and walls based on a deviation between the plane normal and the unit vector of the oz axis. Moreover, a set of thresholds including a surface size, and distances from vertices of triangulation meshes and points to the planes to eliminate unrealistic walls. Interestingly, Romero-Jaren and Arranz [24] used a pair of holes in both extracted floor and ceiling to identify locations of the columns, while the proposed method required user-predefined base and dimensions of columns to create 3D column models. In this work, 3D models of floors and walls were created by using Delaunay triangulation established from data points of floors and ceilings, which were located within a buffer of 10cmm from lower and upper evaluation boundaries. Although the study reported that 90% data points were classified correct labels, the proposed method still required user interaction during creating the model. However, those works are applicable for building configuration matching a Manhattan-world assumption [43], and slanted floors and ceiling slabs of the buildings can be unsuccessfully extracted. Additionally, most methods are limited to extract and reconstructed planar components, for example, floors, ceilings, and walls. An additional surveying on indoor object detection and reconstruction can be found in Bassier and Vergauwen [22].

A work on processing a point cloud of a construction project has also attracted researchers. For example, to assess the quality of full-scale precast concrete element, Kim et al. [44] mapped an 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 the two models indicated quality of the construction elements. To assess flatness

of a concrete surface, Li et al. [45] used RANSAC [30] to extract point cloud of ceilings from a sub-sampling data set. The flatness is measured as distances between the surface points to the reference surface determined from 80% points of the surface for quality assurance and control based on point clouds, a one-class support vector machine approach was developed to extract rebars in reinforced concrete members, in which a set of features including linearity, planarity and red–greenblue colors of a point cloud were used [46]. Similarly, Kim et al. [26] used machine learning based on 7 geometrical features derived from algebraic fittings and principal component analysis (PCA) to determine diameters of rebars and their spacing. The study discovered that a support vector machine (SVM) is the optimal machine learning classifier from the set of methods (Naïve Bayes, Discriminant Analysis, Classification Tree, Nearest Neighbor, and Support Vector Machines). SVM can predict with an accuracy of 97.2% for rebar diameters over 25 mm but of 56.0% for the rebar diameters less than 20 mm. However, for dense data points, the study has little improvement of efficiency, in which the executing time of the proposed method was 44.5 min compared to 53 min from a manual work. Laefer and Truong-Hong [25] used kernel density estimation (KDE) to detect primary surfaces of a steel cross-section to determine its shape and dimensions, and subsequently the final section of the steel member was identified by comparing the standard sections to the point cloud-based section. Similarly, de Souza et al. [47] proposed a slicing method to determine the steel members and measure deformation of individual members.

In a recent effort on automatically extracting structural components from a point cloud, Bosche [10] mapped a 3D CAD model to the point cloud to extract members of an as-built model. The integration of the BIM model and the point cloud were used to determine the flatness of the floor [28]. Similarly, Chen and Cho [48] aligned the point cloud of a construction site to a BIM model to identify construction deviation. To extract columns from a point cloud, Díaz-Vilariño et al. [18] employed 2D HT transform to extract columns from a rasterized image from point clouds of a building projected into the xy plane. Subsequently, detected pixels of the columns were mapped to the point cloud to extract rough data points of the columns. The study reported that 12/19 and 7/10 circular and rectangular columns were respectively detected. However, the method required the column faces to be parallel to x- and y-axes, and the thresholds to distinguish pixels on edges of the columns and of other regions. Moreover, in order to extract concrete structural elements (e.g. slabs, beams and columns) from a point cloud, Son and Kim [49] used color information of a point cloud to identify a region of interest consisting of the same construction material class. Next, an edge-based segmentation was employed based on a voxelization model of the region of interest created by a supervoxel algorithm. The segmentation refinement was implemented to discard small segments based on segment size, while segment's features, for example linearity, planarity, and direction, are used to classify segments as column, beam/girder, floor/slab/wall, and other by using SVM. Although the proposed method succeeded in extracting the structural components, it required huge manual labor work to prepare the training set and computation time can reach 282 s per one million points. In a similar goal, Maalek et al. [19] proposed a hierarchical method to extract structural elements (e.g. floor, ceiling/slab, columns and rebars) from a construction site. The method started to use a histogram generated from point elevations. Next, pointwise planar and linear features of each point computed from a robust principal analysis were used to cluster the points of surfaces. The column was identified from a set of adjacent surfaces in a form of a symmetric section and parallel to main directions of a building. Finally, linear segments within a boundary of the column were considered as rebars. Moreover, in a work on detecting planes from a photogrammetric point cloud of a construction site, Xu et al. [50] decomposed the point cloud in to voxels, and the global clustering in the form of graph-based segmentation was adapted to cluster the planar facets within the voxels. Subsequently, 3D HT constrained by an orientated bounding box was used to estimate the planar model.

In summary, given the complexity of building construction projects with a high number of structural elements with different shape, size and orientation, and the large amount of the point cloud, there is still a lack of efficient methods to straightforwardly process the data [17,43]. Existing methods require an as-design model and/or certain assumption (e.g., a floor and ceiling are horizontal planes [27]) to extract structural components and are time-consuming [49]. Thus, this paper proposes a new method to automatically extract point clouds describing structural elements of a concrete building at its construction phase.

3. Proposed method

3.1. Scope of work

This paper proposes a method to automatically extract structural elements of concrete buildings at construction stage from a terrestrial laser scanning point cloud and quantify the construction quality. Main components of a building consist of floors, ceiling-slabs, walls, columns, and primary and secondary beams. When using a laser scanning to capture building geometry, only geometric information of visible surfaces of the building components can be obtained. As such, there are only one surface representing the floors, ceilings, and walls, and may be three surfaces describing embedded columns (except for the embedded column at a corner explicitly has two surfaces) and beams. Contextual knowledge of building structures is used to develop a hypothesis of the proposed method, select parameters for processing 3D point clouds of the building, and filter unrealistic objects to be recognized. As buildings have different funtionalities, for example office or residential buildings, features and dimensions of their structure elements are also varied. However, their components have common features as below.

Feature 1: floors, ceilings and walls are planar surfaces. The floors and ceilings are often nearly horizontally, while the walls are mostly perfect vertical.

Feature 2: columns are vertical, and a cross-section of the column is either square or rectangle. Each column must be made from at least two orthogonal planes. The columns are often distributed in rectangular grids with at least two columns per grid.

Feature 3: a primary beam connects to at least one column head, and another column head or a wall or none (for a cantilever beam). Notably, primary beams connecting between two walls are out of a scope of this study.

Feature 4: secondary beams are supported by primary beams and end points of the secondary beam connect to two primary beams.

Feature 5: minimum dimensions of structural elements (floors, ceiling slabs, columns, and beams) used in this study are obtained from design requirements and practice implementation (Table 1).

Table 1
Minimum dimensions of structural elements of buildings used in this study.

Name	Notation	Values	References
Minimum width of floor/ceiling slab/wall (m)	$S.W_{min}$	2.5/ 1.2(*)	[51]
Minimum length of floor/ceiling slab/wall (m)	$S.L_{min}$	2.5	[51]
Minimum thickness of floor/ceiling slab/wall (m)	$S.T_{min}$	0.1	[52]
Maximum inclined angle of floor/slab/wall (degree)	α_{obj}	10/ 45(**)	[53]
Minimum cross-section of a column and beam (m)	$CB.W_{min}$ & $CB.H_{min}$	0.2x0.2	[54]
Minimum length of a beam (m)	$B.L_{min} = S.W_{min}$	2.5/ 1.2(*)	

(*) *Values used for office and residential buildings respectively.*
(**) *Values respectively used for the floor/slab and wall.*

From design practice, the span length ($S.W_{min}$ or $S.L_{min}$) of ceiling slabs is often larger than 2.5 m derived from the minimum dimensions of rooms [55]. However, for residential buildings, the width of the celling slabs or walls can be about 1.2 m ($S.W_{min}$). In a concrete building, to satisfy allowable deflection, the thickness of the floors and slabs is designed no smaller than 0.1 m [52]. Moreover, a wall thickness depends on the type of the walls, and its thickness is often larger than the brick width. As such, the minimum thickness ($S.T_{min}$) of floors, ceiling slabs and walls is set at 0.1 m. Surfaces of the floors, ceiling slabs and walls may not be parallel to the building axes (e.g., the longitudinal direction). In building structure, a slope of the floor (or ceiling slab) is about 12% or 6.8 degrees [53]. As such, the maximum angle (α_{obj}) between the surface nomals of the floors and ceilings and the vertical direction is about 10 degrees, which can add compesation of data errors. In addition, for the building walls, the maximum angle (α_{obj}) between the surface normals of the walls to the building axes is no larger than 45 degrees. That is because when the angle α_{obj} of the wall along an long edge of the building is larger than 45 degrees, it can be treated as the wall along the short edge, and vice versa. In practice, cross-sections ($CB.W_{min} \times CB.H_{min}$) of the columns and beams are no smaller than 0.2 m × 0.2 m (width × height) although the minimum dimension of the column is 0.25 m [54]. Finally, as the beams support the slabs, the shortest length of the beam ($B.L_{min}$) are set equal to the minimum span length of the floors and ceiling slabs and walls.

3.2. Proposed method

The paper proposes a method to automatically extract structural elements of concrete buildings at a construction stage from a terrestrial laser scanning point cloud, which consists of 4 consecutive steps to extract: (i) floors/ceiling slabs and walls, (ii) columns and (iii) primary beams and (iv) secondary beams. In this way, as the proposed method process subsets of data to extract point clouds of a single type of components, the processing complexity is significantly reduced. Moreover, as point clouds of building components are immediately deactivated after each step, the proposed method avoids exhausted computations. The proposed method is a synergy between spatial information as contained in a point cloud and contextual knowledge of structures to develop working principles of the proposed method and set parameters to extract structural components (Fig. 1). Moreover, when constructing reinforced concrete buildings, structural elements of each storey must have been inspected before moving to the next storey. This implies a point cloud of a single storey is used as input data for the proposed method rather than a point cloud of an entire building.

As a building has an arbitrary orientation in geo-space, a building orientation (wide and long directions) is needed to align to parallel to the axes of the global coordinate system before exporting data for the proposed method. To align the building, a two dimensional (2D) minimum bounding box (mBB) of the building is determined from the x- and y-coordinates of the data points. Next, a rotation matrix is estimated from the angle between the directional vector of the long edge of 2D mBB to the unit vector ($u_y = [0, 1, 0]$) of y axis. The rotated point clouds are then considered as input data for the proposed method. As the goal of the alignment is to reduce the number of cells when decomposing the building into 2D cells and to support in clustering columns in rectangular grids in the following steps, the alignment does not require to be obtained highly accurately. Notably, before estimating the 2D mBB, the point clouds of objects outside the building are manually removed. This work is often done within the proprietary software of the laser scanner before exporting x-, y- and z- coordinates of the point cloud for further processing, and often takes a couple of minutes.

3.3. Floor, ceiling slab and wall extraction

Floors, ceiling slabs, and walls of reinforced concrete buildings are often planar surfaces, but have different size and orientation. The methodology is used to extract the floors, ceiling slabs and walls that are nearly similar, which consists of three sub steps: *Step 1.1*: use a quadtree representation to decompose the point clouds into 2D cells in the xy plane, *Step 1.2*: roughly extract local surfaces of the floor and ceiling

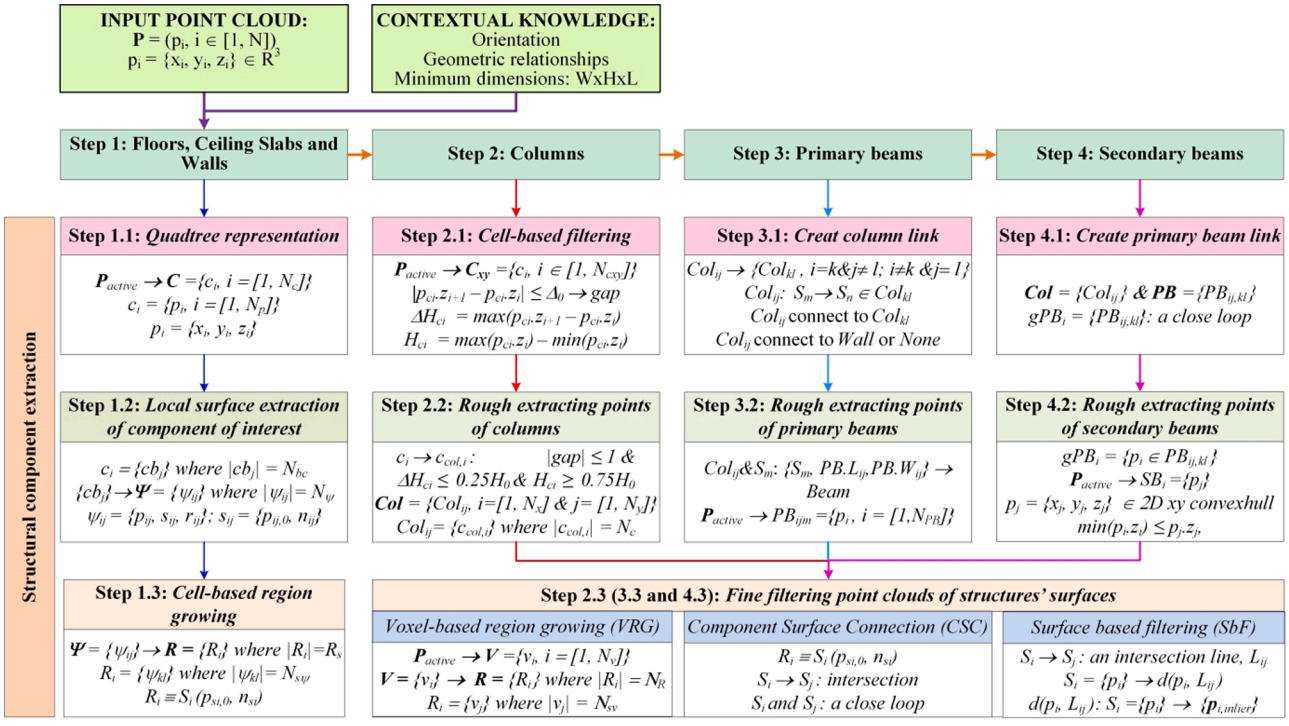


Fig. 1. Workflow for inspecting structural components in reinforced concrete buildings.

slab, and *Step 1.3*: filter the final points of the components using cell-based region growing segmentation (CRG).

Step 1.1: Quadtree representation

To reduce a complexity of data processing and to prevent intensive computations, the algorithm employs a quadtree [56] to recursively subdivide an initial 2D bounding box of the input point cloud ($\boldsymbol{P} = \{p_i = (x_i, y_i, z_i)\} \in R^3$) into 2D cells ($\boldsymbol{C} = \{c_i\}, i = [1, N_c]$) until a termination condition is reached. Several termination criteria have been widely used in research, for example, the maximum number of points within a cell [57] or the depth of the quadtree [58]. In this study, the cell size (ce_0) is considered as the termination condition, which is selected based on the minimum size of the structure surface. A cell c_i is classified as "*full*" if it contains at least a predefined minimum number of points (c_{min_ptc}); otherwise, it is an "*empty*". Moreover, during recursive subdivision, only full cells are subdivided into 4 smaller cells. The full cells on the leaf nodes of the quadtree are used in further processing. Each full cell c_i describes as a tuple consisting of two opposite corners ($c_i.x_1$, $c_i.y_1$, $c_i.x_2$, $c_i.y_2$) and a list of point indices ($c_i = \{p_i\}, i = [1, N_p]$) where N_p is the number of points within the cell. Fig. 2 shows a point cloud decomposed into 2D cells in the xy plane.

Step 1.2: Local surface extraction of component of interest

After decomposition of data points of a building into 2D cells in the xy plane, a cell c_i may contain points of multiple components (e.g., a floor, ceiling slab, beam, or column) in the elevation direction. Additionally, patches corresponding to a floor, ceiling slab and other components within the cell c_i are distributed at different locations along the vertical direction, and a large void space is often available between different horizontal components. Candidate points of the floor and ceiling slab located within the cell c_i are extracted in two sub-steps. The first sub-step decomposes the points within the cell into multiple cell blocks, which are distinguished by void space due to a large gap between the point clouds along the depth direction of the cell. For the 2D xy cell, the void space is a form of the cell area and the elevation difference between two consecutive points, where there is no data point available. The points of cell blocks are extracted according to Algorithm 1. The goal is to extract the cell blocks containing the points of the floor and slab, in which a distance between them is the approximate height of the storey (or width or length of the room for the walls). Therefore, the void space threshold vs_0 is set equal to a half of the smallest value of the room dimensions (width, length, and height), which is used to determine the void space within the data points of the cell (Fig. 3a and b). This selection is to overcome available of point clouds of non-structural element (e.g., a scaffolding).

Algorithm 1: Extracting the cell block

Input: the data points $(p_i) = (x_i, y_i, z_i) \in c_i$ and a void space threshold vs_0
Output: $c_i = (cb_j$, where $|cbj| = N_{cb}$ and $cb_j = (p_k)$
Begin
1. Sort p_i according to z coordinates, where $p.z_i < p.z_{i+1}$
2. $j = 1$;
3. $k = 1$ and block $cb_j = (p_k)$;
4. while $k < |p_i|$
if $p.z_{k+1} - p.z_k \leq vs_0$ then $p_{k+1} \rightarrow cb_j$;
else j++ and $cb_j = (p_{k+1})$;
k++;
return $cb = (cb_j)$

In the second sub-step, kernel density estimation (KDE) is implemented to identify the orientation of the surfaces within the cell block cb_j. However, as floors and ceiling slabs can be slanted, the use of KDE based on the z-coordinates of the data points $p_j \in cb_j$ may not detect these surfaces. As such, for each cell block cb_j (Fig. 4a), the points $p_j \in cb_j$ are respectively rotated around the × and y axes with a local origin at the centroid of the points p_j. In this implementation, the rotation angle is in the range [-α_{obj}, α_{obj}] with an appropriate angle interval (α_{int} = 2 degrees), in which α_{obj} is the angle between the surface normal of the component and the unit vector of the axis. As such, surfaces of the wall and column are not yet detected. For the floors and ceiling slabs, the angle α_{obj} is about 6.8 degrees equivalent to the maximum floor slope by 12% [53]. Subsequently, KDE generated from z-coordinates of rotated points p'_j of the points p_j are used to determine the optimal rotation angle (α_{opt}), as the rotation angle corresponding to the *KDE* having the largest local peak. Next, given a probability density shape (*PDS*) of the *KDE* corresponding to the optimal rotation angle α_{opt}, a valley-peak-valley pattern is exploited to extract the points of the planar surface according to Equation (1), in which the valleys and peaks of the *PDS* are determined through the second derivative of the PDS (Fig. 4a).

$$cb_j \text{®} y_{ij} = (p_{ij} | PDS.z_{valey,k} \text{£} p'_{ij}.z \text{£} PDS.z_{valey,k+1}) where p_{ij} \hat{\text{I}} p_i and p'_{ij} \text{I} p'_j \quad (1)$$

where $PDS.z_{valey,k}$ and $PDS.z_{valey\ k+1}$ are the coordinates of two consecutive valleys of the PDS.

This extraction separates the points of the cell block to a set of patches ψ_{ij} (Fig. 4b), and the points $p_i \in c_i$ are assigned to patches $\psi_{ij} \in c_i$, where $j = [1, N_\psi]$, denoted as a local surface (Fig. 3c). The patches in the cell are ordered according to decreasing elevation. Notably, by using this simple assumption, the patches ψ_{ij} may possess points of adjacent surfaces, for example the patch ψ_{i1} in Fig. 4b.

In this proposed method, patch ψ_{ij} is assumed to represent a planar surface. A plane is fitted to points $p_{ij} \in \psi_{ij}$ using *PCA* based on a covariance matrix $cov_{\psi ij}$ (Equations (2) and (3)) [59]. The normal vector

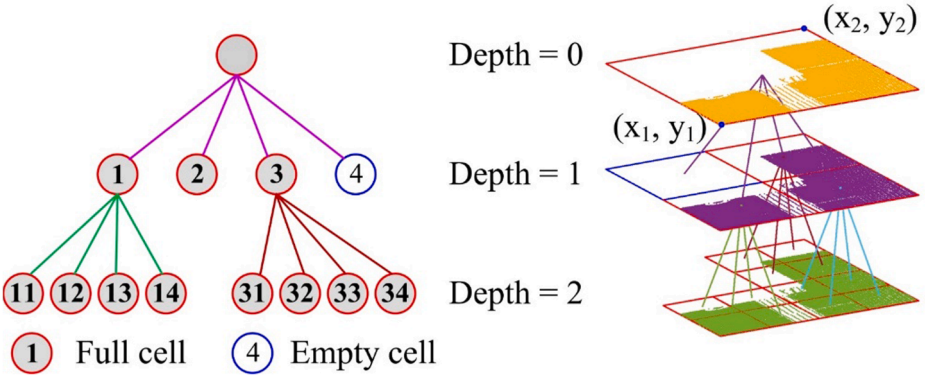


Fig. 2. Quadtree subdivision of a data set in xy plane.

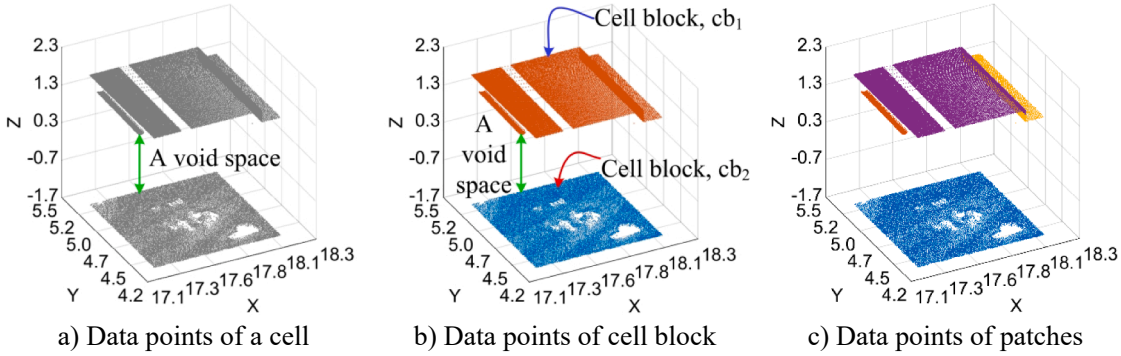


Fig. 3. Extracting data points of patches within a 2D cell.

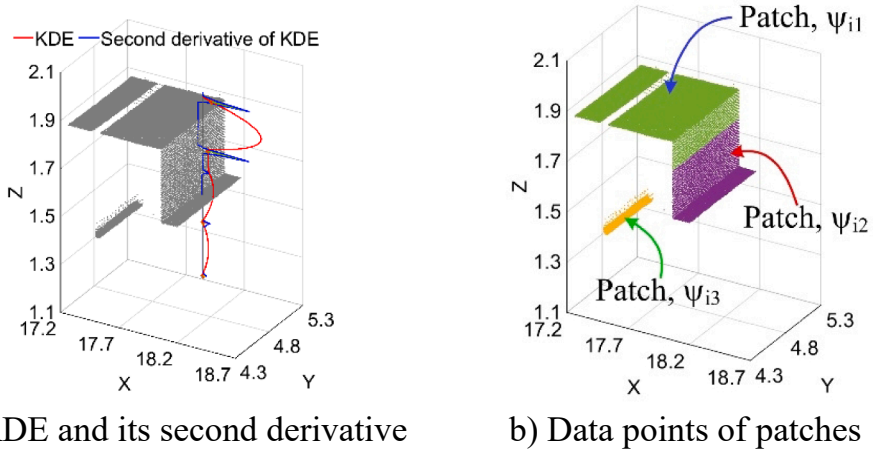


Fig. 4. Illustration of extracting points of patches in the cell block cb_2 shown in Fig. 3b.

n_{ij} of the fitted plane (s_{ij}) is the eigenvector corresponding to the smallest eigenvalue of the $cov_{\psi ij}$.

$$cov_{\psi ij} = \sum_{i=1}^{N} \left(p_{ij} - p_{ij,0}\right)\left(p_{ij} - p_{ij,0}\right)^T \tag{2}$$

$$p_{ij,0} = \frac{1}{|p_{ij}|}\sum p_{ij} \tag{3}$$

where $p_{ij,0} = (x_{ij,0}, y_{ij,0}, z_{ij,0})$ is the centroid of the points $p_{ij} \in \psi_{ij}$, and $|\,|$ denotes the cardinal of the p_{ij}.

Moreover, the residual (cr_{ij}) is defined as the root mean square distances d_{ij} from the points to the fitting plane (s_{ij}) as given in Equations (4) and (5). Finally, the patch is described by the tuple $\psi_{ij} = (p_{ij}, s_{ij}, cr_{ij})$.

$$d_{ij} = \frac{\left(x_j - x_{ij,0}\right)n_{ij}.x + \left(y_j - y_{ij,0}\right)n_{ij}.y + \left(z_j - z_{ij,0}\right)n_{ij}.z}{\sqrt{n_{ij}^2.x + n_{ij}^2.y + n_{ij}^2.z}} \tag{4}$$

$$cr_{ij} = \sqrt{\frac{1}{|d_{ij}|}\sum_{i=1}^{N} d_{ij}^2} \tag{5}$$

where d_{ij} is the Euclidean distance from the points p_{ij} to the fitting plane s_{ij}.

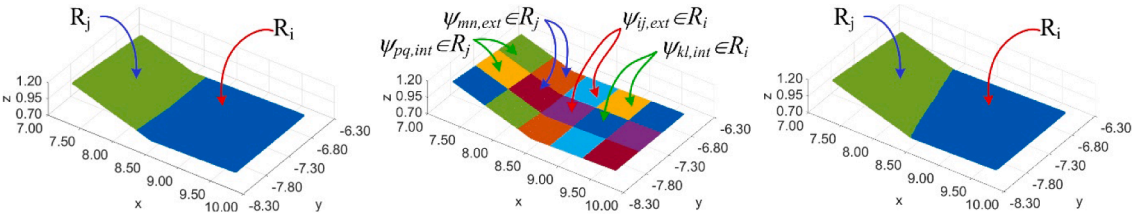
Step 1.3: Cell-based region growing (CRG)

Candidate points of the floor and ceiling slab are extracted through patches using KDE, but may include points of adjoined components. *Step 1.3* introduces a cell-based region growing (CRG) method to obtain floor and ceiling slab points. CRG consists of 3 sub-steps: patch-patch region growing (*Step 1.3.1*), patch filtering (*Step 1.3.2*), and patch-point region growing (*Step 1.3.3*).

Step 1.3.1: Patch-patch region growing (PPRG)

Intuitively, patches ψ_{ij} representing local planar surfaces of the floor and ceiling slab are connected and co-planar. A patch-patch region growing (PPRG), similar to region growing-based methods [31,60], is proposed to cluster those patches having similar salient features. The algorithm starts to filter candidate patches of the floor and ceiling slab, which are the patches ψ_{ij} having an angle between the normal vector n_{ij} and the unit vector u_z no larger than the angle α_{obj}, which is the slanted angle of the floor and ceiling slab. Next, the patch $\psi_{ij} \in c_i$ having the smallest residual value ($cr_{ij} \rightarrow$ min) is considered as the initial seeding patch to search neighbouring patches ($\psi_{kl} \in c_k$). The patch ψ_{kl} is added to a region (R_i) of the patch ψ_{ij} if the patch ψ_{kl} satisfies Equation (6) and is added to the seeding patch set for next iterations if its residual value cr_{ki} is smaller than the residual threshold (cr_0). The growing process is completed when all candidate patches are checked. Notably, in this implementation, the neighbour patch searching is done through the cell containing the patches. For example, the patches $\psi_{kj} \in c_k$ are the neighbouring patches of the patch $\psi_{ij} \in c_i$ if the cells c_k are adjoined to the cell c_i through its vertices and edges.

$$\begin{cases} \angle n_{ij}, n_{kl} \le \alpha_0 \\ d\left(p_{kl,0}, s_{ij}\right) \le d_0 \end{cases} \tag{6}$$



a) Point clouds of two adjoined regions R_i and R_j before filtering

b) Classification of patches within the regions

c) Point clouds of two adjoined regions R_i and R_j after filtering

Fig. 5. Illustration of patch filtering.

where $d(p_{kl,0}, s_{ij})$ is Euclidean distances between the centroid of the patches ψ_{kl} to the fitting plane $s_{ij}(p_{ij,0}, n_{ij})$ of the patch ψ_{ij} , and α_0 and d_0 are the angle and distance thresholds, respectively.

PPRG groups the patches into a set of the regions $\boldsymbol{R} = \{R_i, i = [1, N_R]\}$. However, as mentioned before, patches may contain data points of other adjoined components, for example, walls, columns, or beams (Fig. 5a). Those points must be assigned to correct segments through *Step 1.3.2* and *Step 1.3.3*, respectively.

Step 1.3.2: Patch filtering

Goal of a patch filtering is to re-assign points within patches to correct regions representing surfaces of the structural components [60]. The filtering algorithm classifies the patches $\psi_{ij} \in R_i$ into exterior ($\psi_{ij,ext}$) and interior patches ($\psi_{ij,int}$) (Fig. 5a and b). The patch ψ_{ij} is an interior patch $\psi_{ij,int}$ if all its neighbouring patches belong to the region R_i; otherwise, it is an exterior patch $\psi_{ij,ext}$. Next, the points $p_{ij} \in \psi_{ij}$ are classified as inlier ($p_{ij,inlier}$) and/or outlier ($p_{ij,outlier}$) points based on the distances from the points p_{ij} to the fitting plane $s_{\psi ij,local}$ (Equation (7)). Notably, the filtering algorithm also applies to the interior patches $\psi_{ij,int}$ because these patches may also include the points of non-structural objects (e.g., components of mechanical, electrical and plumbing - MEP), and the filtering process is only applied to the R_i having both the exterior and interior patches.

$$p_{ij} \rightarrow \begin{cases} p_{ij,inlier} if d(p_{ij}, s_{\psi ij,local}) \le d_0 \\ otherwise p_{ij,outlier} \end{cases} \tag{7}$$

where $d\left(p_{ij}, s_{\psi ij,local}\right)$ is Euclidean distances from the points p_{ij} to the local surface $s_{\psi ij,local}$, and d_0 is the distance threshold.

The local plane $s_{\psi ij,local}$ is determined from the points p_{kl} of the interior patches ($\psi_{kl,int} \in R_i$) derived from the neighbour patches ψ_{kl} of the patch ψ_{ij} using the robust PCA (*rPCA*) [25]. rPCA is similar to PCA in *Step 1.2* but the weight ($w(p)$) of each point is included in the covariance matrix (Equation (3)). The weight is expressed in Equation (8). This weighting scheme allows to minimize the impact of outliers on estimating the local plane $s_{\psi ij,local}$.

$$w(p) = exp\left(\frac{-d^2}{\sigma^2}\right) \tag{8}$$

where σ is the standard deviation of the Gaussian kernel of the distance (d) from all points to its fitting plane, in which $\sigma = 1/3d$ is used in this work.

If a patch ψ_{ij} is an interior patch $\psi_{ij,int}$, the outlier points $p_{ij,outlier}$ are discarded. However, if the patch ψ_{ij} is an exterior patch, the outlier points $p_{ij,outlier}$ can subsequently be assigned to the adjacent regions R_j (Fig. 5b). For this case, the interior patches $\psi_{pq,int} \in R_j$ adjoined to the patch $\psi_{mn,ext}$ are retrieved, in which the patch $\psi_{mn,ext}$ is the neighbouring patches of the patch $\psi_{ij,ext}$. Subsequently, the local plane $s_{\psi mn,local}$ of the region R_j at the patch $\psi_{mn,ext}$ is estimated (Equations (2) and (3)). A score voting is used to assign the points $p_{ij,outlier}$ to the region R_j (Equation (9)). Notably, the second condition is applied in the case of the number of the region R_j larger than 1. For example, Fig. 5c shows a result of the patch filtering.

$$p_{ij,outlier} \rightarrow R_j if \begin{cases} d(p_{ij,outlier}, s_{\psi mn,local}) \le d_0 \\ d(p_{ij,outlier}, s_{\psi mn,local}) \rightarrow min \end{cases} \tag{9}$$

Step 1.3.3: Patch-point region growing

Goal of this step is to re-assign those points of the exterior patches, which have not been checked in *Step 1.3.2* to correct regions. The patch-point region growing algorithm involves two consecutive sub-steps. In the first sub-step, the process starts with exterior patches $\psi_{ij,ext} \in c_i \in R_i$ to search adjacent patches $\psi_{kl} \notin R_i$. The points $p_{kl} \in \psi_{kl}$ are candidate points $p_{kl,cand}$ to merge to the region R_i if the points satisfy the first condition in Equation (9), where the local plane $s_{\psi ij,local}$ is replaced by the fitting plane s_{ij} of the patch ψ_{ij}. The points $p_{kl,cand}$ may belong to multiple, adjacent, parallel segments. For example, the patch occupies the points of two adjacent slabs or the walls, in which the slabs or walls can be distinguished by a beam or column. As such, in the second sub-step, an incremental slicing method is proposed to search new points of the region R_i from the candidate points $p_{kl,cand}$. In this implementation, the points $p_{ij,proj}$ and $p_{kl,cand,proj}$ are the projection of the points $p_{ij} \in \psi_{ij,ext}$ and $p_{kl,cand} \in \psi_{kl}$ onto the fitting plane s_{ij} of the patch ψ_{ij}. A 2D polygon (2DPp) of the patch ψ_{ij} is created from vertices of the 2D mBB determined from the points $p_{ij,proj}$. The searching incrementally enlarges the polygon 2DPp with an interval step (δ_0) to search points $p_{kl,cand,proj}$ located within the polygon, and these points are added to the region R_i. The searching is terminated when no point is added to the region R_i. In this study, the interval step is set equal to a 1/4 of the minimum cross-section of the column and beam (width - $CB.W_{min}$ and length - $CB.H_{min}$).

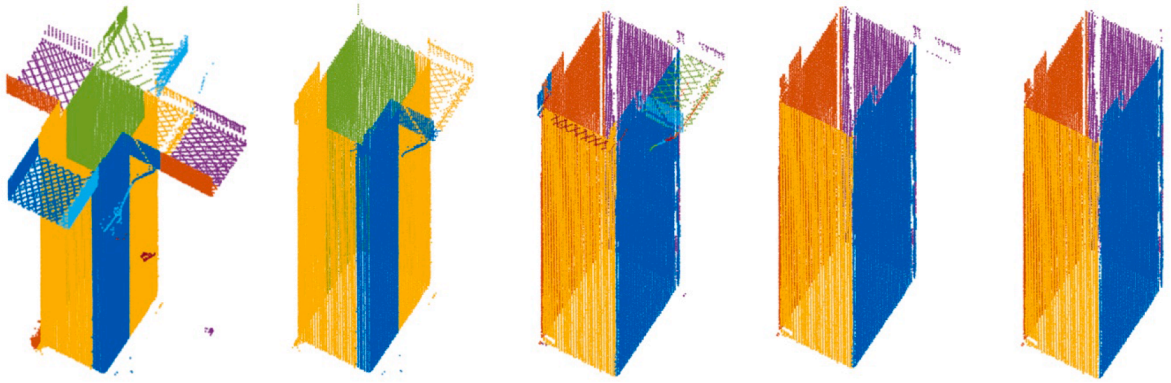
Additionally, width and length ($R_i.W$ and $R_i.L$) of the region R_i are determined as dimensions of a 2D mBB estimated from final points of the region R_i projected onto the fitting plane, which is estimated by using PCA (Equations (2) and (3)). The region R_i is considered as the floors, ceiling slabs, or walls if the region dimensions satisfy Equation (10).

$$R_i = (R_i.W, R_i.L) \rightarrow \begin{cases} if R_i.W \le S.W_{min} \wedge R_i.L \le S.L_{min} \rightarrow realfloor, ceilingorwalls \\ otherwise \rightarrow unrealfloor, ceiling, orwalls \end{cases} \tag{10}$$

Finally, once the data points ($\boldsymbol{P}_{floor}$ and $\boldsymbol{P}_{slab}$) of the floor and ceiling slab are extracted and deactivated, the active points ($\boldsymbol{P}_{\text{active}} = \boldsymbol{P} \setminus (\boldsymbol{P}_{floor} \cup \boldsymbol{P}_{slab})$) are subsequently used to extract walls using a similar process. The active points $\boldsymbol{P}_{\text{active}}$ decompose into 2D cells in the yz and xz planes to extract walls in these planes, respectively. However, as surfaces of walls are often perfectly vertical, the point clouds within the cell block only rotate around the z axis to determine the optimal rotation angle (α_{opt}). Finally, the CRG is employed to extract the data points of each wall.

3.4. Column extraction

In concrete buildings, columns are often vertical elements



a) 2D xy cells around the column | b) Candidate points of the columns | c) Resulting surfaces from VRG | d) Column surfaces from CSC | e) Final surfaces from SbF

Fig. 6. Illustration of column extraction.

connecting the floor and ceiling slab. The majority of concrete columns have symmetric, rectangular or squared cross-sections. Moreover, the columns are often distributed in rectangular grids and every grid must have at least two columns. After extracting the floors, ceilings and walls, the point clouds of these components are deactivated. 2D cells in the *xy* plane are generated from the active data points $\boldsymbol{P}_{active}$, where the cell size by 1/4 ce_0 is used to control a subdivision (Fig. 6a). The column extraction algorithm has three main steps (Fig. 1).

Step 2.1: Cell-based filtering

Theoretically, the points in the column continuously distribute from the floor to the ceiling slab. However, in practice, occlusions or missing data often occur. In *Step 2.1*, for each cell c_i, three features that are the number of void spaces, the maximum gap (ΔH_{ci}) and the cell height (H_{ci}) are computed to extract the cells representing a column (Equation (11)). A cell c_i is considered as a column cell $c_{col,i}$ if its features satisfy Equation (11). The criteria are selected to overcome incompleteness of the data. However, cells of non-structural components (e.g., door frames) have the point distribution in the vertical direction similar to that of the column cell, which cannot be eliminated by the criteria in Equation (12). This type of the cells can be removed at the column level (*Step 2.2*).

$$\begin{cases} |p_{ci}.z_{i+1} - p_{ci}.z_i| \le \Delta_z \rightarrow gap \\ \Delta H_{ci} = max(|p_{ci}.z_{i+1} - p_{ci}.z_i|) \\ H_{ci} = max(p_{ci}.z_i) - min(p_{ci}.z_i) \end{cases} \tag{11}$$

$$c_i \rightarrow c_{col,i} if \begin{cases} |gap| \le 1 \\ \Delta H_{ci} \le 0.25 \Delta H_0 \\ H_{ci} \ge 0.75 H_0 \end{cases} \tag{12}$$

where $p_{ci}.z_i$ is z coordinate of the point $p_{ci} \in c_i$, Δ_z is the gap threshold, which is empirically selected equal to 10 %H_0, and H_0 is the distance between the floor and slab, which is the average distance between patches of the floor and ceiling slab.

Step 2.2: Rough extracting points of columns

A cell connectivity, similar to connected component labelling algorithm [61], is proposed to cluster adjoined column cells $c_{col,i}$, which are possibly describe the column (Fig. 3b). The method starts with a random column cell $c_{col,i}$ to search the adjoined column cells $c_{col,j}$ and subsequently add these cells to a region of the column cell $c_{col,i}$. This process iteratively groups the column cells into the cluster and is completed when all column cells are examined. The cell connectivity returns a set of the clusters $\boldsymbol{CL} = (cl_i, i = [1, N_{cl}]$ and $cl_i = \{c_{col,j}\})$, where each cl_i may represent a column.

As mentioned above, the clusters ***CL*** may contain unreal columns because cells of non-structural elements have similar features to ones of the column**.** Next, the unreal columns are eliminated based on *Feature 2* that the columns distribute in rectangular grids. The process starts to compute the cluster center ($p_{cli,0}$) from the data points of the cluster. Next, KDE is generated from x- and y- coordinates of the cluster center $p_{cli,0}$ and a valley-peak-valley pattern (Equation (1)) is used to extract the clusters on the same grid along × and y directions. In this implementation, the bandwidth equals to 1/4 the shortest distance between two adjacent columns, known as the shortest length of the beam ($B.L_{min}$). Finally, the cluster is considered as the unreal column and removed out the cluster ***CL,*** if the cluster belongs to the grid with only one cluster. Thus, a resulting filtering generates a set of the columns $\boldsymbol{COL} = (Col_{ij}, i = [1, N_{col,x}], j = [1, N_{col,y}]$ and $Col_{ij} = \{c_{col,j}\})$, where $N_{col,x}$ and $N_{col,y}$ are column grids in × and y directions. The data points ($\boldsymbol{P}_{col,ij}$) in each column are used as input data in a next step to filter the final points of the column (Fig. 6b).

Step 2.3: Fine filtering points of columns

This step aims to extract the final points of the column through 3 sub-steps: (1) voxel-based growing segmentation (VRG), (2) a connected surface component (CSC) filtering and (3) surface-based filtering (SbF).

Step 2.3.1: Voxel-based growing segmentation (VRG)

In the first sub-step, the octree-based region growing developed by Vo et al. [32] is adopted to segment points of the column surfaces. The pure octree representation [62] is used to subdivide the point cloud of the column into 3D voxels, in which the voxel size (ve_0) equals to 1/4 of the surface size or the minimum cross-section of the column, is selected as a termination condition. Notably, this is different from the work of Vo et al. [32] where the residual threshold and voxel size are used to control a subdivision process. The voxel is labelled as "*full*" if the number of the points within the voxel is no smaller than the predefined minimum number of points (v_{min_ptc}); otherwise, it is "*empty*". Additionally, data points of each full voxel (v_i) on the leaf nodes are assumed to represent the plane, in which the plane parameters are estimated by using PCA (Equations (2) and (3)) and the residual vr_i is also computed according to Equation (5). Next, similar PPRG, the VRG groups the voxels in a set of the regions $\boldsymbol{R} = (R_i, i = [1, N_v]$ and $R_i = \{v_i\})$ based on a region growing methodology [32].

However, as data points are decomposed into 3D voxels, which implies the voxel may contain data points of multiple surfaces. Under- and

over-segmentations occur in the octree-based region growing [32], which requires to refine voxels on the region boundary. The voting-based distance is used to overcome over- and under-segmentation into two sub-steps applied to (i) exterior voxels of the region and (ii) un-segmented voxels. The process starts to classify the voxels $v_i \in R_i$ into interior and exterior voxels ($v_{i,int}$ and $v_{i,ext}$), in which the exterior voxel $v_{i,ext}$ is the voxel connected to less than 4 voxels of the region R_i or at least one voxel of the adjacent region R_j. In the first sub-step, for each exterior voxel $v_{i,ext} \in R_i$, its neighbour voxels $v_{j,ext} \in R_j$ are retrieved. Next, the interior voxel $v_{k,int} \in R_i$ and $v_{l,int} \in R_j$ connected to the exterior voxel $v_{i,ext}$ and $v_{j,ext}$ are respectively extracted. Subsequently, the local planes ($s_i = (p_{i,0}, n_i)$ and $s_j = (p_{j,0}, n_j)$) of the regions R_i and R_j are respectively estimated based on points $p_k \in v_{k,int}$ and $p_l \in v_{l,int}$ by using rPCA (*Step 1.3.2*). The points $p_i \in v_{i,ext}$ are re-assigned to corresponding the segment R_i or R_j based on Equation (13).

$$v_{i,ext} = (p_i) \rightarrow \begin{cases} R_i if d(p_i, s_i) \rightarrow min \\ R_j if d(p_i, s_j) \rightarrow min \end{cases} \quad (13)$$

where $d(p_i, s_i)$ and $d(p_i, s_j)$ are Euclidean distances from the point p_i to the local planes s_i and s_j.

In practice, due to missing data, noisy data, or surface defects, the VRG can segment a real surface of a structural component as multiple regions, which are needed to merge. The merging process starts with the exterior voxel $v_{i,ext} \in R_i$ and $v_{j,ext} \in R_j$ to respectively search their neighbour voxels $v_l \in R_j$ and $v_k \in R_i$ using the searching window (*sw*). The voxels are known as the neighbourhood of the given voxel if the distances between their centres are no larger than the searching window *sw*. In this study, the *sw* equals to $3ve_0$ is used, which translates the smallest gap between the region R_i and R_j is about $2ve_0$. Notably, this value can be adjusted based on the quality and quantity of the data. Next, the local planes s_k, s_l and s_{kl} are respectively fitted through the points $p_j \in v_k \in R_i$ and $p_j \in v_l \in R_j$, and $p_j \cup p_k$ using rPCA (*Step 1.3.2*). The regions R_i and R_j are considered as parts of the structure surface if distributions of the points p_j on the local planes s_k and s_{kl} and of the points p_k on the local planes s_l and s_{kl} are similar. Correlations between $d(p_i, s_k)$ and $d(p_i, s_{kl})$, and between $d(p_j, s_l)$ and $d(p_j, s_{kl})$ are used to measure this similarity. The regions R_i and R_j are merged if the correlations are no less than 0.9.

After merging the region R_j to the region R_i, if dimensions (width - $R_i.W$ and length – $R_i.L$) of the region R_i satisfies Equation (14), this region is considered as a surface of a column, which is an input data for *Step 2.3.2*.

$$R_i = (R_i.W, R_i.L) \rightarrow acolumnsurfaceifCB.W_{min} - S.T_{min} \leq R_i.W \quad (14)$$

where $CB.W_{min} - S.T_{min}$ is a smallest width of a visible surface of a column, which is for a case the column embedded to the wall (the smallest wall thickness – $S.T_{min}$), and $R_i.W$ and $R_i.L$ are dimensions of 2D mBB of the region data points projected onto the fitting surface of the region R_i, in which the fitting surface is estimated from the region data points using PCA (Equations (2) and (3)).

Step 2.3.2: A connected surface component (CSC) filer

Step 2.3.1 segments candidate points of a column in a set of the regions $\boldsymbol{R} = (R_i, i = [1, N_v])$ involving column and un-column surfaces (Fig. 6c). This step proposes a connected surface component (CSC) filter to determine real surfaces of a column from the regions $\boldsymbol{R}$ (Fig. 6d). The method is based on the hypothesis that the component surfaces of the column connect together in a form of a close loop (Fig. 7). The CSC algorithm starts with the initial reference region R_i to search the adjacent regions R_j. The regions R_i and R_j are connected if their surfaces satisfied the condition in Equation (15).

$$\begin{cases} \angle n_i, n_j > \dfrac{\pi}{2} - 2\alpha_0 \\ |P_{j1}P_{i2}| \geq 0.5 R_i.L \end{cases} \quad (15)$$

where n_i and n_j are respectively normal vectors of fitting planes S_i and S_j of the regions R_i and R_j, $|P_{j1}P_{i2}|$ is the overlap length between lines $P_{i1}P_{i2}$ and $P_{j1}P_{j2}$ (Fig. 7a), α_0 is the angle threshold, and $R_i.L$ is the edge length of the region R_i where two regions R_i and R_j intersect. The fitting planes S_i and S_j are estimated applying PCA (Equations (2) and (3)) based on the points of the regions R_i and R_j. The lines $P_{i1}P_{i2}$ and $P_{j1}P_{j2}$ are respectively line segments generated from projected points of the points $p'_i \in R_i$ and $p'_j \in R_j$ on the intersection line L_{ij}. sub-set points $p'_i \subseteqq p_i \in R_i$ and $p'_j \subseteqq p_j \in R_j$ are determined based on Equation (16).

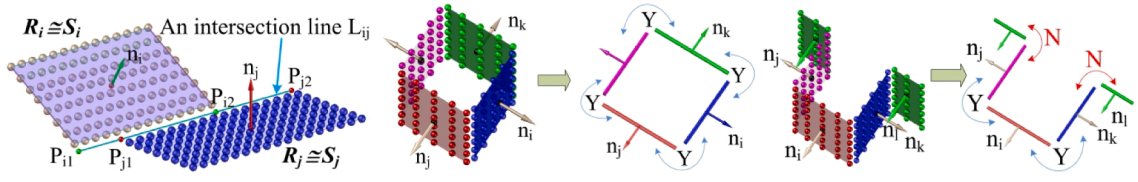
$$\begin{cases} \{p_i \rightarrow p'_i | d(p'_i, L_{ij}) \leq R_i.w \\ \{p_j \rightarrow p'_j | d(p'_j, L_{ij}) \leq R_j.w \end{cases} \quad (16)$$

where $d(p'_i, L_{ij})$ and $d(p'_j, L_{ij})$ are Euclidean distances from the points p'_i and p'_j to the line segment L_{ij}, and $R_i.w$ and $R_j.w$ are widths of the regions R_i and R_j. The region dimensions including witdh $R_i.w$ and length $R_i.L$ are edge lengths of 2D mBB determined from the projection of the points $p_i \in R_i$ onto the fitting plane S_i.

Once the connected regions are established, the regions R_i and R_j are considered as parts of the column, if all regions R_j are located in one side of the region R_i. For example, the R_i and a pair of R_j and R_k are parts of the column (Fig. 7b and c); however, in Fig. 7c, the R_k and a pair of R_i and R_j are not parts of the column surfaces. Additionally, the region R_j is used as the reference surface for a next iteration, and the process is terminated if no additional surface of the column is recognized. The column is considered as the real column if they have at least two surfaces. This criterion is used to obtain the column at the building corners.

Step 2.3.3: Surface-based filtering (SbF)

A surface-based filtering (SbF) proposes to remove outlier points of the region or surface. These points can be available due to data errors or non-column objects adjoined to the column. The filtering is based on an observation that points of a surface bounded by the adjoined surfaces. Any points out of this bound is considered as an outlier point. To eliminate outlier points of the surface, an intersection line L_{ij} between the surface $S_i \equiv R_i$ and $S_j \equiv R_j$ divides the points $p_i \in S_i$ into inlier ($p_{i,inlier}$) and outlier ($p_{i,outlier}$) groups, which is determined based on signs of sign distances $d(p_i, L_{ij})$. The outlier point group has small number of the points ($|p_{i,outlier}| < |p_{i,inlier}|$), and are then discarded (Fig. 6e).



a) Determine surface connection b) Case 1: all planes accepted c) Case 2: 2 planes rejected

Fig. 7. Illustration of the CSC filter.

3.5. Primary beam extraction

In concrete building structures, beams can consist of primary and secondary beams, in which the primary beam (PB) connects two columns, or a column and a wall, while the secondary beam (SB) connects two primary beams. This section focuses on extracting the primary beams, while the secondary beams present in the following section. The primary beam (PB) is extracted through three steps: *Step 3.1* – Create column link, *Step 3.2* – Rough extraction of primary beam and *Step 3.3* – Fine filter point clouds of the primary beam. Notably, *Step 3.3* is similar to *Step 2.3* in Section 3.2

Step 3.1: Create column link

This step is to establish a link between a column to column and a column and wall, where the primary beam may connect them. In a concrete building, the columns are mostly symmetric, rectangular, or square sections and distributed in rectangular grids (Fig. 8a). A primary beam intersects to the column through its surface, which is used to indicate the primary beam. Moreover, due to complexity of a structure and/or scene, where the column is embedded to the wall, all column surfaces may not be extracted. There are 3 cases that the primary beam can connect: *Case 1*: column Col_{ij} and Col_{kl} indicated by two surfaces $S_m \in Col_{ij}$ and $S_n \in Col_{kl}$, *Case 2*: column Col_{ij} and Col_{kl} indicated the surface $S_m \in Col_{ij}$ only when the surface S_n is not available, and *Case 3*: column Col_{ij} and a wall indicated by $S_m \in Col_{ij}$. Notably, the columns Col_{ij} and Col_{kl} are on the same rectangular grids, where $i = k$ and $j \neq l$, or $i \neq k$ and $j = l$ (Fig. 8b).

The algorithm starts to estimate the fitting planes of sides of all columns using PCA (Equations (2) and (3)), in which surface normals are set outward the column (Fig. 8c). Next, from a column Col_{ij}, the closest columns Col_{kl} on the same × and y grids are retrieved. Case 1 is available if there is an opposite surface $S_n \in Col_{kl}$ of the surface $S_m \in Col_{ij}$. The surface S_n is determined based two conditions (Equation (17)): (1) to determine the opposite and parallel surface S_n of the surface S_m, (2) to ensure that the surfaces S_m and S_n are on the same side regarding to the column center. For example, in Fig. 6c, when searching the opposite surface for $S_{ij,1} \in Col_{ji}$, $S_{i-1j,3} \in Col_{i-1j}$, $S_{ij+1,3} \in Col_{ij+1}$, and $S_{ij-1,3} \in Col_{ij-1}$ satisfy the first condition, while only $S_{ij+1,3} \in Col_{ij+1}$ satisfies the second condition. Thus, the primary beam between the columns Col_{ij} and Col_{ij+1} is through the surfaces $S_{ij,1}$ and $S_{ij+1,3}$.

$$\begin{cases} cos(n'_m, n'_n) \leq 0 and \angle n'_m, n'_n \leq \alpha_1 \\ 0 < d(p_{n,0}, S_m) \end{cases} \quad (17)$$

where n'_m and n'_n are projection of the normal vectors of the surface S_m and S_n onto the xy plane, $d(p_{n,0}, S_m)$ is a sign distance from the point $p_{n,0}$ on the surface S_n to the surface S_m, and $\alpha_1 = 10$ degrees is the angle threshold.

If the surface S_n is not available, for example when searching the opposite surface for the surface $S_{ij,4} \in Col_{ji}$ and $S_{ij,2} \in Col_{ji}$ (Fig. 8c), the primary beam can connect the surface $S_m \in Col_{ij}$ to the column Col_{kl} (Case 2) or the wall (Case 3). For Case 2, the column Col_{kl} is located on the same side of the surface S_m regarding to the column Col_{ij} (Equation (18)).

$$\begin{cases} 0 \leq cos(n'_m, t'_{ki}) and \angle n'_{ij}, t'_{ki} \leq \alpha_1 \\ 0 \leq d(p_{col,kl}, S_m) \end{cases} \quad (18)$$

where t'_{ki} is a projected vector of the vector between the centers of the columns Col_{ij} and Col_{kl} on the xy plane, $d(p_{col,kl}, S_m)$ is the sign distance between the point $p_{col,kl}$ and the surface S_{ij}, and $p_{col,kl}$ is the center of the column Col_{kl} computed as average of the points of the column.

Finally, if the column Col_{kl} cannot identify, the primary beam from the column Col_{ij} connect to the wall. For example, the primary beam connects to the Col_{ij} through the $S_{ij,2}$ (Fig. 8c). Once the type of the connection of the primary beam is determined, the parameters including the surface $S_m(p_{m,0}, n_m)$, the estimated length $(PB_{ijm}.L)$ and width $(PB_{ijm}.W)$ are established to extract the candidate points of the primary beam. The length $PB_{ijm}.L$ is set equal to $d(p_{m,0}, S_m)$ for Case 1, $d(p_{col,kl}, S_m)$ for Case 2 and infinity for Case 3, while the estimated width $PB_{ijm}.W$ is the surface width $(S_i.w)$.

Step 3.2 – Rough extracting points of primary beams

Candidate points $p_{il} \in PB_{ijm}$ connect the columns Col_{ij} and Col_{kl} (Case 1 and 2) or the column Col_{ij} and the wall (Case 3) are extracted from the active points $\boldsymbol{P}_{active}$ (Equation. (19)), which is based on three conditions: (1) the points between two columns (or the column and wall), (2) points within the column's width, and (3) points on the top part of the column (Fig. 9a).

$$\boldsymbol{P}_{active} \rightarrow PB_{ijm} = (p_i) if \begin{cases} 0 \leq d(p_i, S_m) \leq PB_{ijm}.L \\ d(p'_i, L_{ijm}) \leq 2PB_{ijm}.W \\ mean(p_m.z) \leq p_i.z \end{cases} \quad (19)$$

where $d(p_i, S_m)$ is a sign distance from the points p_i to the surface S_m, $d(p_i, L_{ijm})$ is a Euclidean distance from the points p_i to the line L_{ijm}, p'_i are the projection of the points p_i on the xy plane, $p_m.z$ and $p_i.z$ are z-coordinates of the points of the surface S_m and of the candidate primary beam, respectively. The line L_{ijm} is defined by two points $p'_{m,0}$ and a directional vector n'_m.

Step 3.3 – Fine filtering points of primary beams

Once the candidate points of the primary beam are extracted, similar

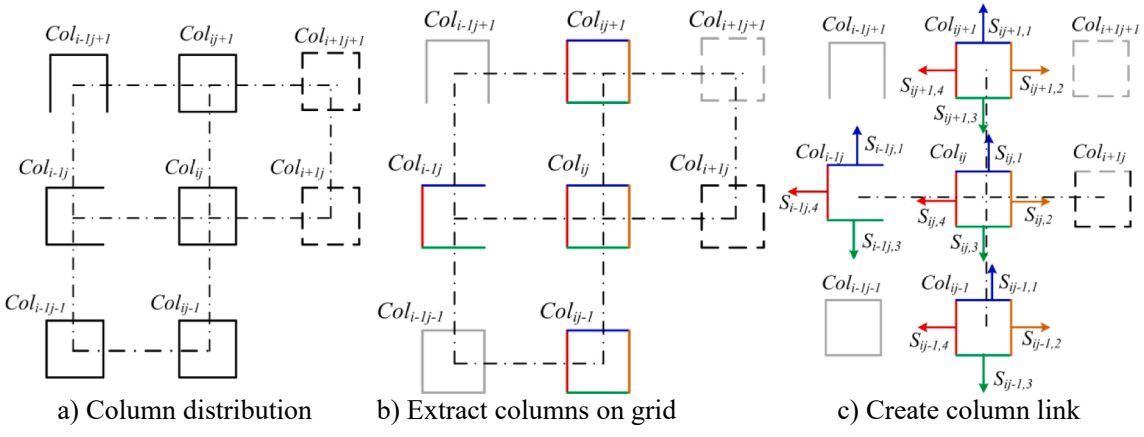
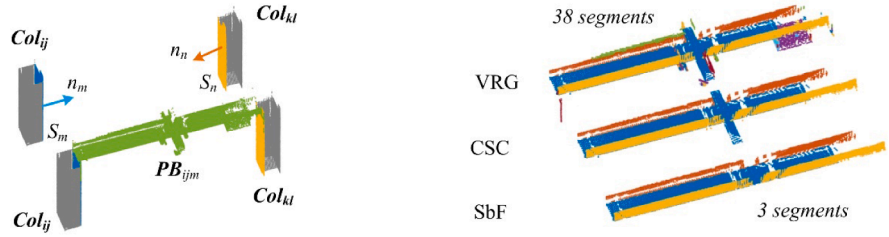


Fig. 8. Create column link to extract the candidate points of the primary beams(*). (*) dash boxes show invisible columns.



a) Rough extracting candidate points b) Fine filtering points on surfaces of a primary beam

Fig. 9. Illustration of extracting the point clouds of surfaces of a primary beam.

to *Step 2.3* in the column extraction, VRG is employed to extract points of planar surfaces, $\boldsymbol{R} = (R_i, i = [1, N_R])$. Similar to extract point clouds of columns, the region R_i represents to a surface of the primary beam if its dimensions (width - $R_i.W$ and length – $R_i.L$) satisfies Equation (20).

$$R_i = (R_i.W, R_i.L) \rightarrow a surface of a primary beam if \begin{cases} CB.W_{min} - S.T_{min} \leq R_i.W \\ CB.L_{min} \leq R_i.L \end{cases} \quad (20)$$

where $CB.W_{min} - S.T_{min}$ is a smallest width of a visible surface of a primary beam, which is for a case the beam embedded to the ceiling slab with the smallest wall thickness ($S.T_{min}$), and $R_i.W$ and $R_i.L$ are dimensions of 2D mBB of the region data points projected onto the fitting surface of the region R_i, in which the fitting surface is estimated from the region data points using PCA (Equations (2) and (3)).

Additionally, the CSC filter is to determine the final surfaces of the primary beam (Fig. 9b). Notably, the initial reference surface for the CSC filter is the bottom surface of the primary beam, which is the surface having the largest area, and the smallest angle between its normal vector and the unit vector u_z. Finally, SbF is applied to filter outlier points (Fig. 9b). These thresholds used in VRG and CSC are similar ones in the column extraction (Section 4.2).

3.6. Secondary beam

As mentioned above, a secondary beam connects two primary beams. The proposed method to extract surfaces of the secondary beam consists of three steps: *Step 4.1* – Create primary beam link, *Step 4.2* – Rough extraction of points of the secondary beam and *Step 4.3* - Fine filtering of points of the secondary beam similar to *Step 2.3*.

Step 4.1 – Create primary beam link

In the concrete building, the primary and secondary beams are on the boundary of a slab, and the secondary beam connects two primary beams. This implies that the points of the secondary beam are bounded by a set of the primary beams in the form of a closed loop. To obtain this, the proposed method starts with the reference beam as the primary beam PB_{kln} connected to the column Col_{kl} and Col_{pq} to search the primary beams PB_{ijm} and PB_{pql}, which respectively connect to the columns Col_{ij} and Col_{pq}. Three primary beams PB_{ijm}, PB_{kln} and PB_{pql} are part of the closed loop if the primary beams PB_{ijm} and PB_{pql} locate on one side of the primary beam PB_{kl}. This is similar to the CSC filter (*Step 2.3.2* in *Section 4.2*) applied for the columns. Next, the primary beams PB_{ijm} and PB_{kln} are set as the reference beams in next iterations to search for a new primary beam of the group. The searching can assign primary beams PB to the group $gPB_i = (PB_{ijm})$ in the form of a closed loop.

Step 4.2 – Rough extracting points of secondary beams

For each primary beam group $gPB_i = (PB_{ijm})$, the 2D convexhull is generated from the x- and -y coordinates of the points of these primary beams. Candidate points of secondary beams (SB_i) are the points from the remaining points $\boldsymbol{P}_{active}$ satisfying: (1) the x- and y- coordinates located within the 2D convexhull and (2) the z coordinates of these points no smaller than the minimum z-coordinates of the points of these primary beams.

Step 4.3 – Fine filtering points of secondary beams

In addition, once the candidate points of the secondary beams are extracted, the final points of the primary beams can be achieved by using the same procedure used in extracting the final points of the primary beams. This consists of flowing steps: (1) VRG to extract the points of surfaces, (2) the CSC filter to determine real surfaces of the beam, and (3) the SbF to eliminate outlier points. Notably, before applying the CSC filter, the resulting region $R_i \in \boldsymbol{R}$ is examined if it is a candidate surface of a secondary beam by using Equation (20).

4. Experiments and results

A goal of experimental tests was to evaluate the proposed method in extracting building components from a construction site. To this end, two concrete buildings with different sizes of the building and its components, functionalities and layouts were used in this evaluation. Resulting point clouds of surfaces of building components from the proposed method were compared to ones from ground truth through a level of location deviation and shape similarity. For the level of local deviation, object- and point-based evaluations were used to compare extracted surfaces from the proposed method and ground truth. Moreover, for shape similarity, differences of area and overlap area of the building component surfaces derived from the proposed method and ground truth were also measured. Finally, executing-time for different data sets were reported to highlight an efficiency of the proposed method.

4.1. Data description

Two buildings: (1) Building 1 - an office building and (2) Building 2 - a residential building were used to demonstrate the proposed method. As the goal of this study is to extract structural components for inspection, the basement and ground storey of Building 1 and 2 were respectively used. Building 1 is a concrete building and has overall dimensions of about 18.5 m wide × 29.5 m long × 3.45 m high. It was scanned by a Trimble TX8 with a maximum scanning range at 120 m and an angular accuracy of 8 μrad in both vertical and horizontal directions [63]. A point spacing of 11.3 mm at a range of 30 m and a total of 11 scanning stations were established to capture the interior storey with maximal data coverage (Fig. 10a). The point clouds were registered by Trimble RealWork software v11.2 [16] with a registration error about 1.57 mm. Finally, 23.5 million points with x-, y- and z- coordinates were exported as input data for the proposed method. Building 2 is a concrete frame and brick walls, and has an overall size of approximate 7.6 m wide × 9.6 m long × 3.32 m high. The building was scanned by a Leica RTC 360 with a sampling step 6 mm at a measure range of 10 m from 4 scanning stations [64]. The point clouds were registered by Leica Cyclone [65]

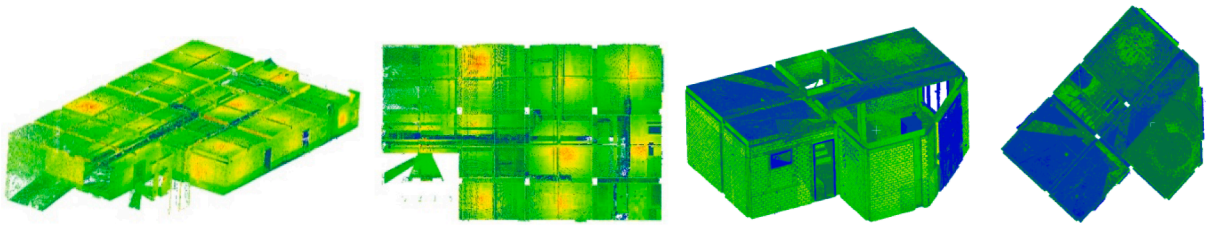


Fig. 10. Point clouds of buildings as input data for evaluation of the proposed method.

with a registration error about 5 mm. As Building 2 was captured with high density point clouds, down-sampling was applied with a sampling step of 5 mm, which leads to 5.94 million points. This down-sampled point clouds was used as input data for the proposed method (Fig. 10b). Notably, during data acquisition, parts of a MEP system were installed in Building 1, which obstructes some structural elements (e.g., slabs and beams), and doors and windows were also installed in Building 2.

4.2. Input parameter selection

As the proposed method deploys both spatial point cloud information and contextual knowledge of building structure derived from building design specifications and practice, input parameters shown in Table 2 are selected based on: (1) geometric information of structural components of the building (Table 1) and (2) data quality.

After decomposing a point cloud of a building into 2D cells, cell blocks containing data points of a floor and ceiling slab, or walls of a 2D cell in the xy or xz/yz planes are extractedbased on a void space defined as a distance between two consecutive points along a depth direction of the cell. As mentioned in *Step 1.2*, the void space threshold vs_0 is selected equal to a half of the minimum dimensions of the room, which translates to $vs_0 = 0.5S.W_{min} = 0.6\ m$ used in this study. Moreover, as KDE is used to identify where high propability distribution of data points to distinguish adjacent objects, the bandwidth (*bw*) is set equal to a 1/4 smallest distance between two adjacent objects. , which is $S.T_{min}$ = 0.1 m (Table 1) for the floor, ceiling slab and wall, and $B.L_{min}$ = 1.2 m (Table 1) for the columns.

To extract a surface by using either CRG or VRG, the algorithm needs at least one cell or voxel located within the surface. This translates that the cell size (ce_0) and voxel size (ve_0) should be no larger than 1/2 the smallest dimension of the surface. As such, the ce_0 equals to 1.0 m and 0.5 m for extracting the floor, ceiling slab and wall in Building 1 and 2, respectively. Moreover, as the columns and beams are embeded to the walls and slabs, the dimensions of the columns and beams derived from the interior scaning point cloud are smaller than those of the designed column and beam. Thus, the ve_0 = 0.05 m is set equal to 1/4 cross-sections of the column and beam ($CB.W_{min} = CB.H_{min}$ = 0.20 m).

In the CRG and VRG, the patches or voxels on the boundaries of the surfaces can occupy the points of two adjoined surfaces. In this case, the deviation angle between the normal vectors of the patches and voxels surrounding the boundary is small although the adoined surfaces, for example the surfaces of the ceiling slabs and beams, are often perpendicular. Moreover, the concrete buildings are objective in this study, where the surfaces of structural components are mostly smooth. Based on an observation above, the thresholds α_0 = 5 degrees, d_0 = 10 mm, r_0 = 10 mm are used in the CRG and VRG. This selection is to prevent over- or under-segmentation, and to minimize the outlier points in the patches and voxels.

Table 2
Input parameters for processing point clouds.

Name	Notation	Values
Bandwidth (m)	*bw*	0.025/0.25(**)
Void space (m)	vs_0	0.6
Cell size (m)	ce_0	1.0/0.5(**)
Voxel size (m)	ve_0	0.05
Angle threshold (degrees)	$\alpha_c = \alpha_v$	5
Distance threshold (mm)	$d_c = d_v$	10
Residual threshold (mm)	$cr = vr$	10
Minimum number of points (points)	c_{min_ptc}/v_{min_ptc}	10/5(***)

(*) Values used for KDE in extracting floor/slab and wall, and identifying the column grids.
(**) *Values used for Building 1 and 2 respectively.*
(***) *Values used for the 2D cell and 3D voxel representation respectively.*

4.3. Results

For extracting the floors and ceiling slabs, an input point cloud was decomposed into 2D xy cells in *Step 1.1,* while *Step 1.2* extracted the data points within the patches describing potential surfaces of the floor and ceiling slab (Fig. 11a and b). As mentioned above, the floors and ceiling slabs can be non-horizontal surface, candidate patches of their components are filtered by comparing the angle between the normal vectors n_{ij} of the patches and the unit vector u_z to the slanted angle α_{obj} (Fig. 11c). The filtered patches were subsequently used as input patches for *Step 1.3.1.* Moreover, other patches in the cells were used in *Step 1.3.2* and *1.3.3* to obtain the final points of the floors and ceiling slabs (Fig. 11d).

A similar process was applied to extract the wall in the yz plane, in which an input data was the active points $\boldsymbol{P}_{active} = \boldsymbol{P} \setminus \boldsymbol{P}_{floor/slab}$. The 2D cells in the yz plane, candidate patches for the first step of the CRG (Fig. 12a), and final resulting points of surfaces of the yz walls from *Step 1.3.1* and *Step 1.3.3* were respectively shown in Fig. 12b and c.

Once the surfaces of the yz walls were extracted, these data points were immediately deactivated. Remain active points $\boldsymbol{P}_{active} = \boldsymbol{P} \setminus (\boldsymbol{P}_{floor}\ U\ \boldsymbol{P}_{yzwall})$ were considered as input data for extracting surfaces of the xz walls, in which input patches and results of *Step 1.3.1* and *Step 1.3.3* are illustrated in Fig. 13.

Once the points of the floors, ceiling slabs and walls are deactivated. Remain active points $(\boldsymbol{P}_{active} = \boldsymbol{P} \setminus (\boldsymbol{P}_{floor/slab}\ U\ \boldsymbol{P}_{xzwall}\ U\ \boldsymbol{P}_{yzwall}))$ were used to extract columns (Fig. 14a). The quadtree was used to decompose the points $\boldsymbol{P}_{active}$ into 2D cells in the xy plane, where the cell size of $1/4ce_0$ was used as a terminated condition of a subdivision. Subsequently, cells representing to columns and candidate points of columns were extracted by using *Step 2.1* and *Step 2.2,* in which resulting candidate points were illustrated in Fig. 14b. Finally, *Step 2.3* is employed VRG, CSC filter and SbF to obtain the point clouds of surfaces of the columns as shown in Fig. 14c. The final points of the columns were immediately deactivated.

Additionally, from resulting column extraction, a link between columns was established (*Step 3.1*). From the active points $\boldsymbol{P}_{active}$ (Fig. 15a), *Step 3.2* extracted candidate points of each primary beam based on the column link, which were shown in Fig. 15b. Fig. 15c illustrates extraction of the primary beams using the fine filtering (*Step 3.3*).

A combination between the columns and the primary beams was

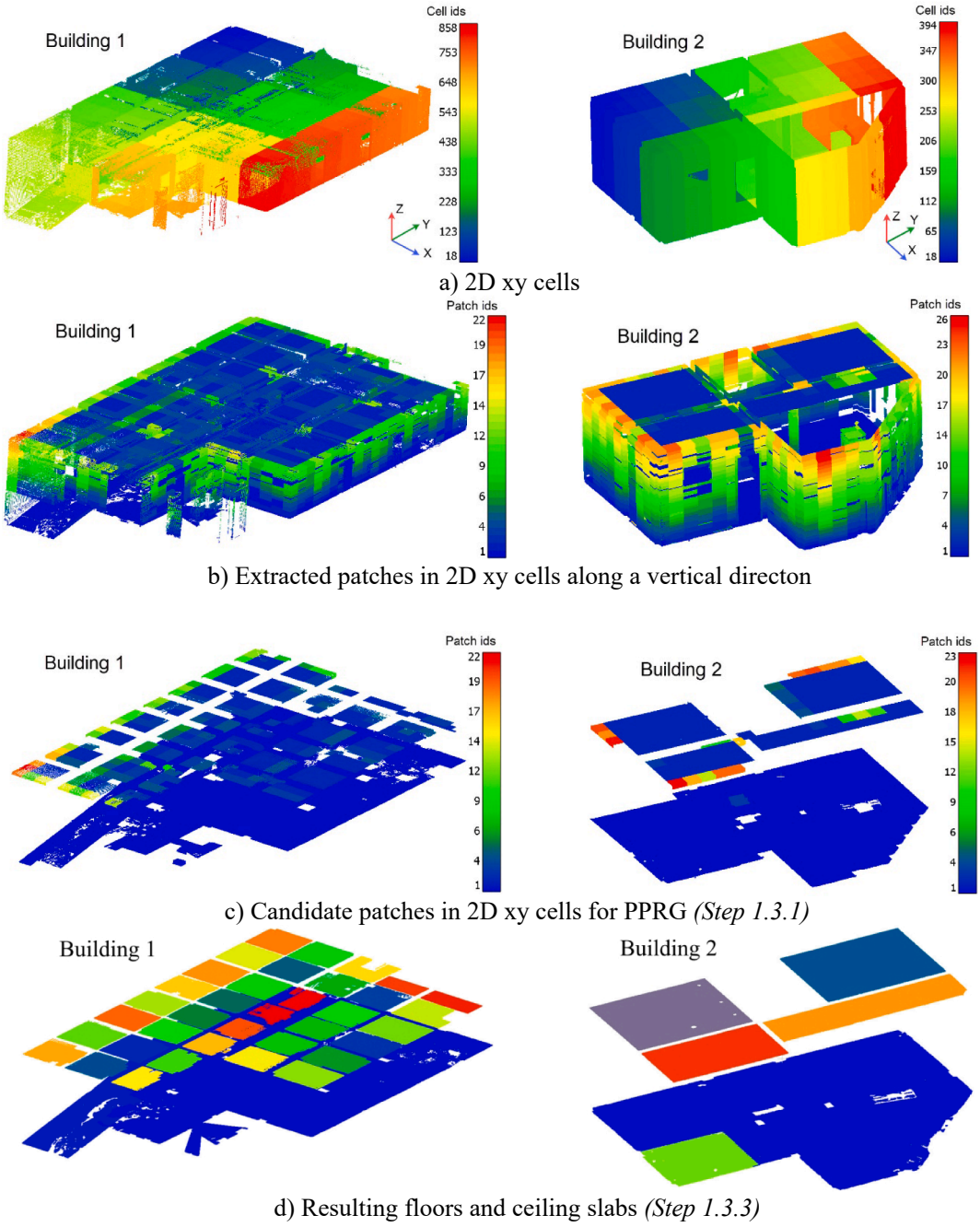


Fig. 11. Extracting point clouds of the floors and ceiling slabs.

used to create the link between the primary beams to make a close loop. from the remain active points $\boldsymbol{P}_{active}$ (Fig. 16a), *Step 4.2* was employed to extract candidate points of secondary beams using the primary beam link (Fig. 16b). Subsequently, the fine filtering (*Step 4.3*) was applied to get the final points of the secondary beams (Fig. 16c). Notably, the secondary beam is not available in Building 2.

4.4. Evaluation

A goal of this study is to extract point clouds of surfaces of structural elements (floors, ceiling slabs, walls, columns and beams). A complete evaluation must be done based on the final model and/or applications to give insight into the performance of the proposed method, but this is out of scope of this study. As such, in this study, a performance of the proposed method is evaluated through a level of location deviation and shape similarity of extracted surfaces with respect to the ground truth [66]. The level of local deviation is to determine a match rate between extracted surfaces from the proposed method (Fig. 17) and ones of ground truth, while the shape similarity is to measure what extent extracted surfaces is similar to the ground truth regarding to properties of the surfaces. This evaluated strategy can partially express impact on 3D modelling based on the extracted surfaces of the building components. In this evaluation, the ground truth was manually extracted from the points of the buildings.

In the level of local deviation, extracted surfaces were evaluated in terms of object and point levels. At the object level, the component was known to be a successful extraction if the overlap between the extracted component derived from the proposed method (PM) and the ground truth (GT) was larger than a predefined threshold of 90% to be empirically selected in this study. For the floors, ceiling slabs and walls, this

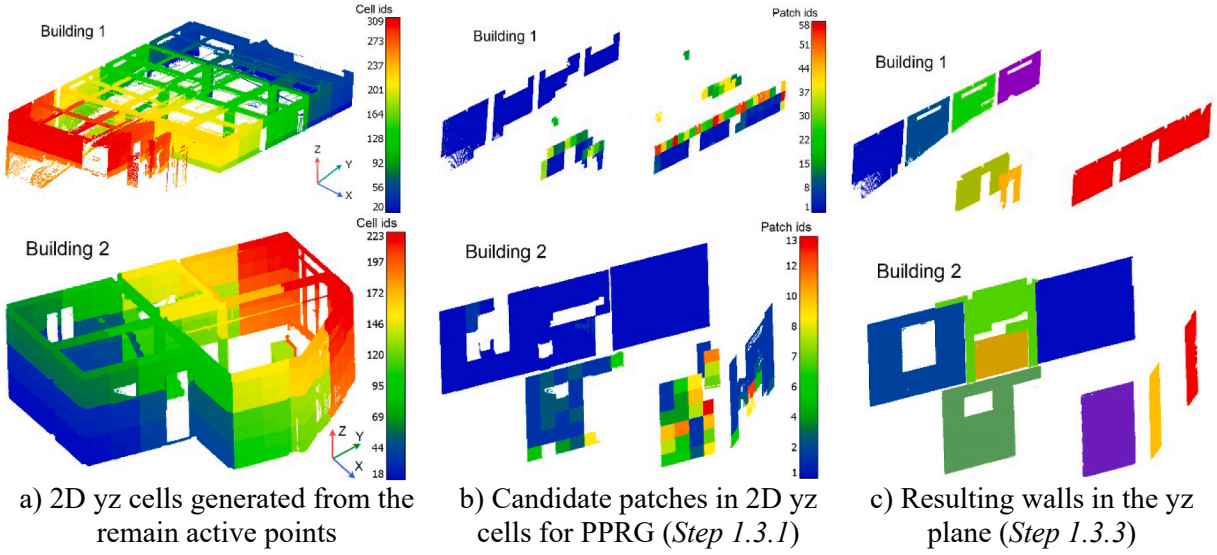


Fig. 12. Resulting point clouds of walls in the yz plane.

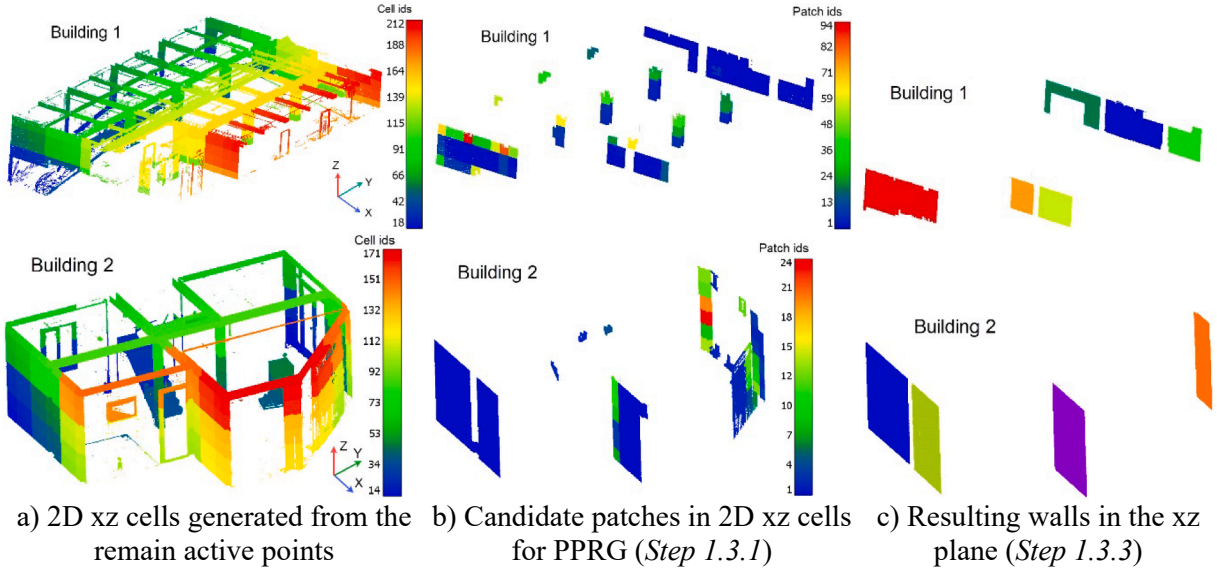


Fig. 13. Resulting point clouds of walls in the xz plane.

measured through an overlap of the surfaces underlying 2D mBB while the 3D mBB was used for columns and beams. Results of the object-based evaluation were shown in Table 3. It can be seen that the proposed method is successfully to extract structures with large size in Building 1 and 2, but the extraction accuracy was reduced for the small components. For example, in Building 2, an accuracy for the wall extraction was about 80.0% because the building has several small walls.

In addition, in the point-based evaluation, the quantity indicators including a true positive (*TP*), false positive (*FP*) and false negative (*FN*) were used to identify a difference between data points of structural elements derived from the proposed method and the ground truth. Next, evaluation quantities can be interpreted through completeness (*Comp.*), correctness (*Corr.*), and quality (*Qual.*). Definition of these quantities and the methodology to compute them can refer to Laefer and Truong-Hong [66]. A resulting evaluation was shown in Table 4, in which the PM can extract the points of the structural components with completeness, correctness and quality no smaller than 96.0%, 96.9% and 93.0% for Building 1, and 80.5%, 95.2% and 78.8% for Building 2. Moreover, the proposed method was given high accuracy in extracting the floor, ceiling slab and wall, but it is seemly difficult to extract the small components, for example the columns and beams, in which the quality of the extracted secondary beams in Building 1 is 86.8% and that of the columns in Building 2 is 78.8%.

In addition, a shape similarity was proposed to measure difference between building components derived from PM and GT, which was interpreted through difference of surface areas and overlap rate measured as a ratio between an area of the surface derived from PM and one from GT [66]. In this evaluation, the floor, ceiling slab and wall were selected to evaluate because these components are in a large size and easily extracted from data sets to create the ground truth. On another hand, other building components (e.g., columns and beams) do not interest because it is difficult to get individual surfaces of the small objects and the ground truth has potentially high bias. To compute the comparison quantity, a point cloud of a structure surface ($p_i \in S_i$) of interest (e.g., floors, ceiling slabs, and walls) derived from PM and GT were projected onto a fitting plane of the ground truth (Fig. 18a). Next, the alpha shape algorithm [67] was employed to extract boundary points ($p_{ext,i}$) of the surface from the projected points, in which the radius

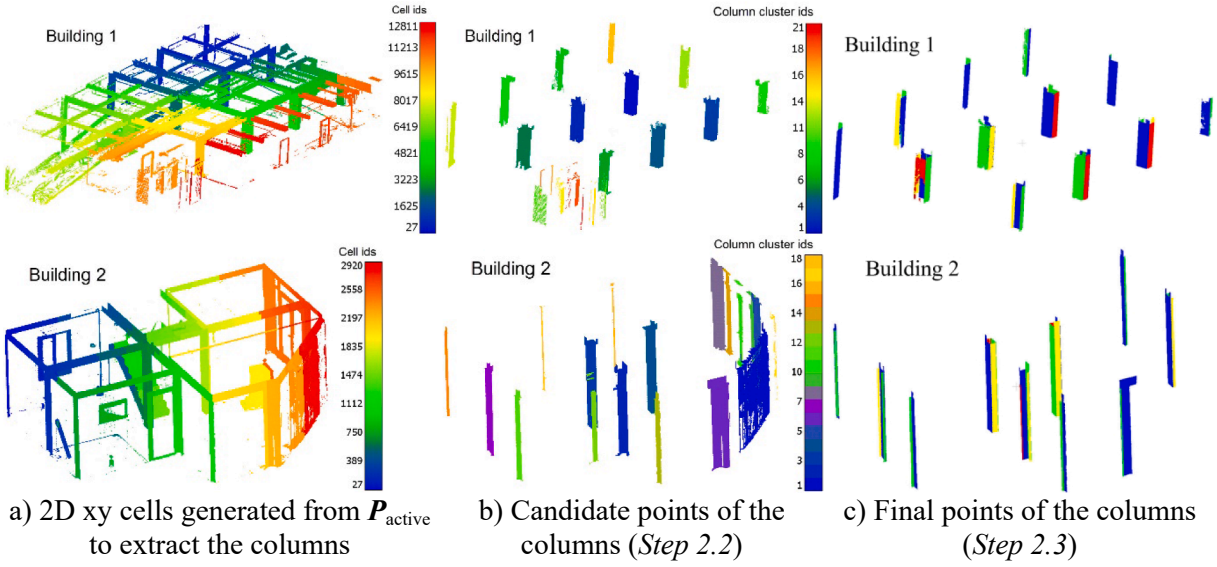


Fig. 14. Extracting data points of the columns.

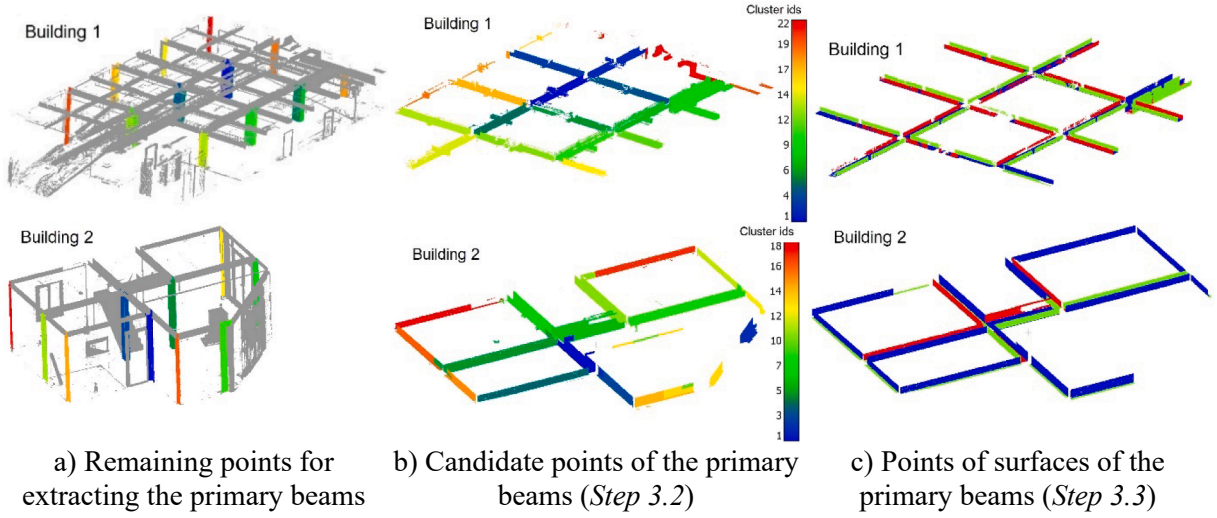


Fig. 15. Extracting data points of surfaces of the primary beams.

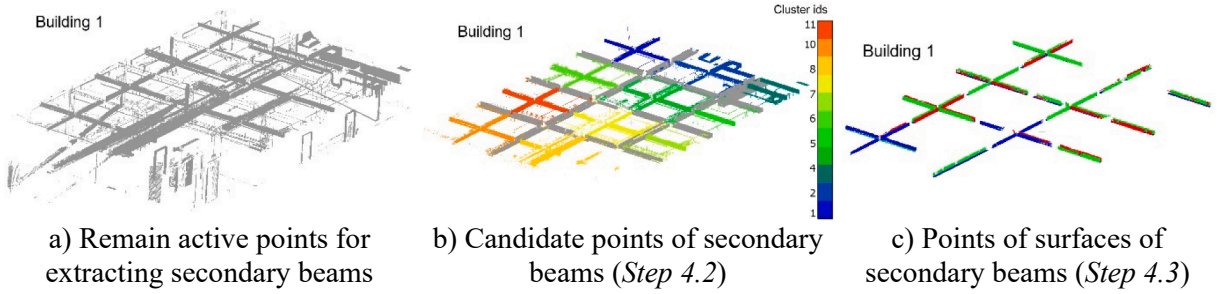
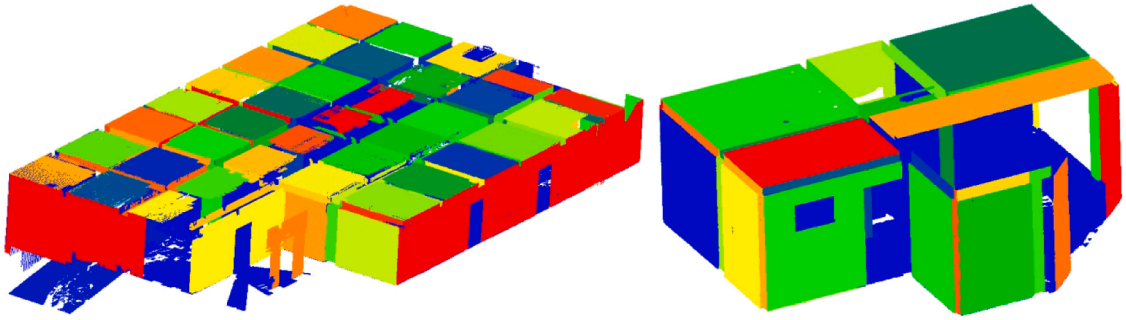


Fig. 16. Extract the data points of surfaces of the secondary beams.

threshold of $0.2ce_0$ ($0.2ce_0 = 0.2$ m for Building 1 and $0.2ce_0 = 0.1$ m for Building), is used (Fig. 18b). Subsequently, the polygon bounded the surface was created from the boundary points $p_{ext,i}$ (Fig. 18c). Finally, the surface area was computed from the polygon, and overlap area was determined as an intersection area between the polygons bounded surfaces of the components from the ground truth and proposed method. A shape similarity from the floors, ceiling slabs and walls was shown in Table 5. Results show that the proposed method extracts the components (e.g., floors, ceilings, and walls) differing ones from the ground truth no more than 5% for Building 1 and 8% for Building 2, in which the lowest overlap rates are 95.3% for the ceiling slab in Building 1 and 92.5% for the wall in Building 2, respectively.



a) Building 1 b) Building 2

Fig. 17. Point clouds of extracted surfaces of individual components from the proposed method.

Table 3
Summarized object-based evaluation metric.

Component	Building 1			Building 2		
	Ground truth	Proposed method	Relative accuracy	Ground truth	Proposed method	Relative accuracy
Floors	2	2	100.0%	2	2	100.0%
Ceiling slabs	36	36	100.0%	4	4	100.0%
Walls	11	11	100.0%	15	12	80.0%
Columns	12	12	100.0%	12	10	83.3%
Primary Beams	16	16	100.0%	17	15	88.2%
Secondary beams	19	17	89.5%	NA	NA	NA

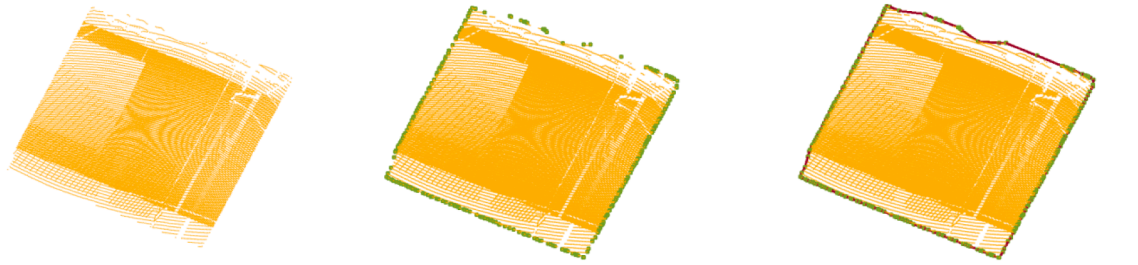
Table 4
Summarized point-based evaluation metric.

Component	Building 1			Building 2		
	Comp.	*Corr.*	*Qual.*	*Comp.*	*Corr.*	*Qual.*
Floors	98.9%	99.7%	98.6%	98.9%	99.1%	98.0%
Ceiling slabs	96.0%	96.9%	93.0%	99.7%	98.8%	98.5%
Walls	99.0%	99.3%	98.3%	96.5%	95.2%	92.0%
Columns	97.6%	99.0%	96.7%	80.5%	97.5%	78.8%
Primary Beams	97.8%	98.6%	96.5%	85.4%	98.9%	84.6%
Secondary beams	88.2%	98.2%	86.8%	NA	NA	NA

Table 6
Summarized processing time of the proposed method (in seconds).

Component	Building 1	Building 2
Floor and ceiling slab	153.7	80.4
Wall	42.1	87.1
Column	72.4	31.6
Beam	203.9(*)	38.7
Total	472.1	237.8

(*) Running time for the primary and secondary beams were 66.6 s and 137.1 s.



a) Data points of a surface b) Boundary points of a surface c) Polygons describing a surface

Fig. 18. Creating a polygon describing the element's surface.

Table 5
Summarized shape similarity.

Component	Area of a component surface (m^2)			Overlap	Area of a component surface (m^2)			Overlap
	Ground truth	Proposed method	Overlap Area	Rate	Ground truth	Proposed method	Overlap Area	rate
Floor	496.20	481.74	479.59	96.7%	60.80	61.38	60.69	99.8%
Ceiling	373.96	368.57	356.26	95.3%	35.87	36.05	35.87	100.0%
Walls	329.16	325.44	323.38	98.2%	72.10	69.70	66.69	92.5%

Additionally, the proposed method is not only successful to extract the building components in high accuracy, but also efficiently accommodates a large data set. The processing time is 472.1 s for 23.5 million points of Building 1, and 237.8 s for 5.94 million points of Building 2 (Table 6). The efficiency is achieved as the proposed method is only processed on a sub-data set when extracting surfaces of a component of interest. Moreover, as point clouds of the buildings were respectively decomposed into 2D cells for extracting the floors, ceiling slabs and walls, and 3D voxels for the columns and beams, the processing time mainly depends on the number of cells and voxels. It can be seen through the running time for the floor and ceiling slab in Building 1 and 2 were respectively 153.7 s and 80.4 s for 858 and 394 cells although the building area is 18.5 m × 29.5 m for Building 1 and 7.6 m × 9.6 m for Building 2. The performance was also a function of the number of points to process. For example, the running time is 72.4 s and 31.6 s although the proposed method processed 22 and 18 column clusters for Building 1 and 2 to obtain results of 12 and 10 columns, respectively. In addition, the processing time for the beam in Building 1 was significantly larger than that of other components because candidate points of secondary beams distributed in a large region (Fig. 16b), which requires VRG processed in the large number of voxels. VRG to obtain the final points of the component surfaces. Thus, the cell and voxel size are a key factor impacting the processing time, and selecting values of them are trade-offs between the smallest size of components to be extracted and running time. Notably, the performance was herein reported based on an implementation of the proposed method in MATLAB 2019b [68] and processing on Dell Precision Workstation with a main system configuration as follows: Intel(R) Xeon(R) W-2123 CPU @ 3.6 GHz with 32 GB RAM.

With running time is about 20.1 s per million points for Building 1 and 40.0 s per million points for Building 2, the performance of this study is arguably better than that of the work of Son and Kim [49], which required about 282 s to process 1 million points to extract and reconstruct structural elements from a point cloud of a construction site. Moreover, the performance of the proposed method is comparable with the work of Poux et al. [69] who used unsupervised segmentation for indoor point cloud, when extracting floors, ceiling slabs, and walls are considered, in which the executing time is respectively 8.3 s per 1 million points and about 6 s per 1 million points for this proposed method (Building 1) and [69]. However, the proposed method in this paper extracts the building components better than the work of Poux et al. [69]. Notably, performances of existing methods used in this comparison is derived from reports of these studies, in which these algorithms were implemented in different programming languages and run-on different computer configuration from one of this proposed method. As such, the comparison is herein to demonstrate a level of an efficiency of the methods rather than a benchmarking.

4.5. Discussions

The proposed method extracted each building component though two steps: rough extracting candidate points and fine filtering points of the surfaces. Importantly, the method also implemented contextual knowledge of building structures to eliminate unreal surfaces of structural components. Through experimental tests of two concrete buildings, the proposed method automatically extracted the structural elements with a high success rate. However, the complexity of the structural elements in terms of size, shape, and arrangement, and of the scene relating to quantity and quality of data can prevent the components fulfilling recognition.

Since the structures are designed to optimize load bearing requirement and life-cycle cost, dimensions of structures vary through the building. In the proposed method, surfaces of components are only extracted if the surface size is larger than the cell size (ce_0) or voxel size (ve_0) (Table 2), and assigned the surfaces as the component surfaces if their dimensions are no smaller than the minimum dimensions of the structure (Table 1). As such, three wall surfaces of Building 2 are failed to extract because the smallest dimensions of these surfaces vary from 0.125 m to 0.3 m smaller than the cell size $ce_0 = 0.5$ m used in the CRG (Fig. 19a). However, although reducing the cell size would help to successfully extract these walls, surfaces of other components (e.g., columns) were also extracted to be recognized as the wall surface.

In addition, as the proposed method uses spatial points cloud information and contextual knowledge, structural components were successfully extracted when distinguished geometric features between these components are available. However, when surfaces of adjoined components are co-planar, no distinguished geometric features in terms of geometry are available, these components are failed to extract. For example, Fig. 19b showed that as the surface of Wall 1, columns and primary beams are co-planar, and only one surface is extracted while two columns were missed. Moreover, recognition of the structural component can be also failed if a configuration of the structure differs from contextual knowledge implemented. For example, two primary beams in Building 2 are failed to obtain because a connection of them differ from contextual knowledge (Fig. 19c). This case imposes the limitation of using contextual knowledge, which is not applicable for a universal configuration. In summary, to overcome those issues, the additional filtering would be applied for the small surfaces after extracting all members satisfied the predefined thresholds in Table 1 and 2. Additional contextual knowledge must be included, for example, the relationship between the floors and ceiling slabs and the beam, and the walls and columns.

On another hand, as a point cloud of the building is decomposed into the cells and voxels and they are only considered in further steps to extract the component surfaces, when they are full cells and voxels. A sparse point cloud causes an increase to the number of empty cells and voxels when the cell and voxel sizes are decreased, which implies results from CRG and VRG can return a real surface of a structure in multiple segments. Particularly, the incremental slicing method in *Step 1.2.3* can be failed to search additional points of the region because there is a large gap between adjacent points due to sparse data. Fig. 20a showed a MEP system available is an obstruction causing sparse data points of the

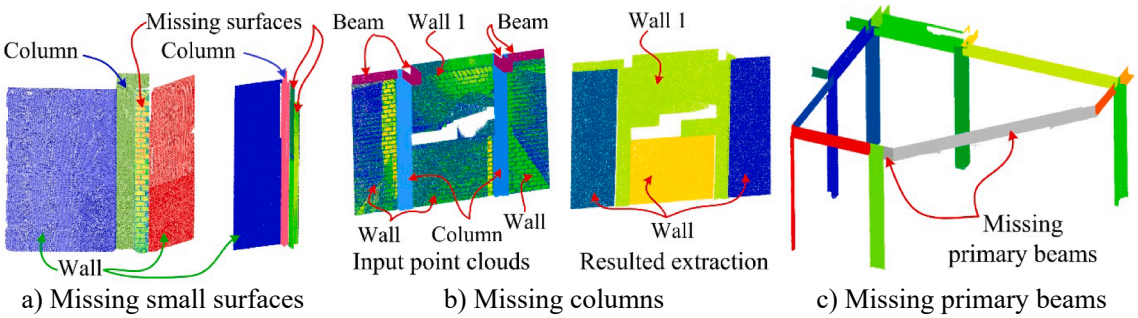
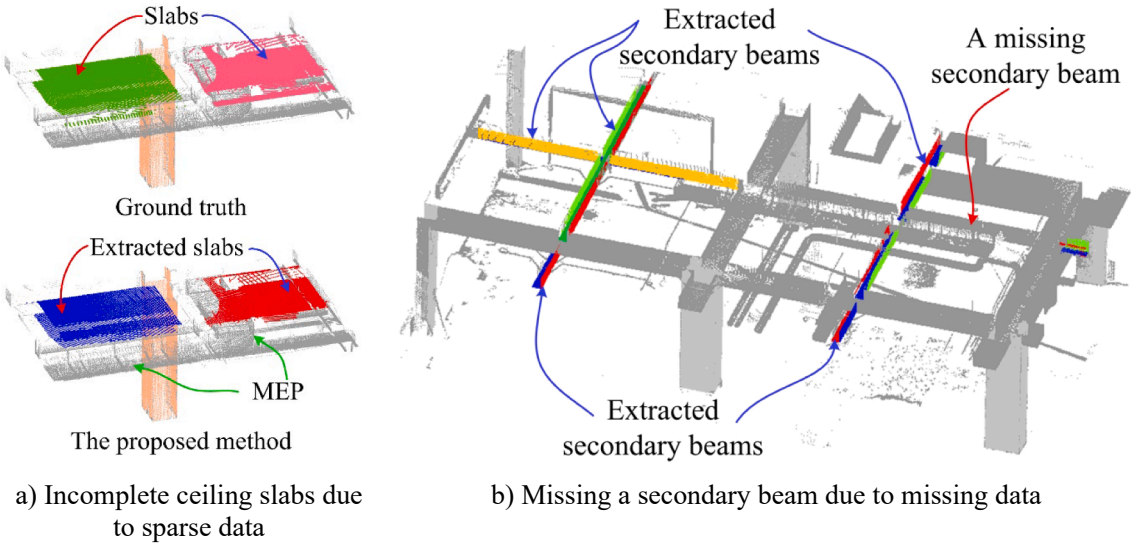


Fig. 19. Extraction failure due to complexity of the structure surfaces.



a) Incomplete ceiling slabs due to sparse data

b) Missing a secondary beam due to missing data

Fig. 20. Incomplete extraction of structural components due to a scene complexity.

ceiling slabs. In addition, missing data of the beam, especially the secondary beam, is restricted to recognize the beam. For example, Fig. 20b shows that the proposed method fails to extract the secondary beam because of missing points of the bottom surface of the beam (Fig. 16b). Thus, the quality and quantity of the data points must be improved.

Finally, in this study, as input data for the proposed method must be a point cloud within a building perimeter, manual work is required to eliminate point clouds of unexpected objects outside a building. This step is often done within a proprietary software of a laser scanner, which often takes a couple of minutes. Moreover, as a complexity of a construction site, input data can contain data points of clusters, non-structural elements (e.g., scaffolding) and moving objects (e.g., construction machines), which can affect resulting segmentation. To mitigate negative impact of such unexpected data points, a prerequisite step would be integrated into the workflow of the proposed method to remove these points of the clusters [70], of non-structural elements through point cloud classification based on construction materials [71], and of construction equipment by using deep learning [55].

5. Conclusions

This paper presents an efficient, automatic method to extract structural components of a construction project of a reinforced concrete building from a construction site. Subsequently, point clouds of surfaces of the structural element can be used for reconstructing 3D models but also for identifying defects at a construction phase. The proposed method deploys spatial information of a point cloud and contextual knowledge about the structures to automatically extract the building components in a sequential order: floors, ceiling slabs, walls, columns, and beams (primary and secondary beams). Additionally, experimental tests showed that the proposed method successfully extracted structural components of two different types of concrete buildings with a high success rate. The evaluation-based points showed that the completeness, correctness, and quality were respectively no smaller than 96.0%, 96.9% and 92.0%, except for the quality of extraction about 78.8% and 84.6% for the column and primary beam in Building 2. This lower success rate was due to a complexity of Building 2 in terms of the size and arrangement of the building elements. One of advantages of this method was that each structural component was extracted through two steps: roughly extracting candidate points and fine filtering the final points. That implied the proposed method only processes a less complex, small sub- set to extract the components, which made the method efficient to a big data encountered in practice. A performance reported executing time about 472.1 s and 237.8 s for 23.5 million points and 5.95 million points.

Even though a high success rate in automatically extracting concrete structural components was achieved based on a combination of spatial point cloud information and contextual knowledge, complexity of structural elements and a scene can prevent extracting all components completely. That is because the proposed method cannot discriminate adjoined surfaces or components having similar geometric features. Additional contextual knowledge derived from a geometric relationship between adjacent components must be included. Moreover, due to the complexity of the scene, the point cloud is subjected to sparse and missing, and parts of the components cannot be detected. To mitigate those impacts, the captured data must be improved not only for data coverage but also data quality. That would improve reliability and accuracy of inspection results based on extracted surfaces. Those are parts of future work of this study. Finally, resulting point clouds of each surface of a building component can be used to quickly report construction defects in terms of geometric quality control, deformation and damaged surfaces, and inspection results would be integrated to a BIM model. The proposed method could be extended for as-built BIM reconstructions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This study was funded by the generous support of the European Commission through H2020 MSCA-IF, "*BridgeScan: Laser Scanning for Automatic Bridge Assessment*", Grant 799149. The first author is grateful for this support. The authors also thank Dat Hop Company Limited, Ceotic., JSC and GeoInstinct Vietnam (https://www.geoinstinct.com), Viet Nam for their providing the laser scanning data.

References

[1] P.E.D. Love, Influence of Project Type and Procurement Method on Rework Costs in Building Construction Projects, J. Constr. Eng. M. 128 (2002) 18–29.
[2] J.L. Burati, J.J. Farrington, Construction Industry Institute–Costs of Quality Deviations in Design and Construction, Clemson University, Source Document, 29 (1987).

[3] M.-K. Kim, Q. Wang, H. Li, Non-contact sensing based geometric quality assessment of buildings and civil structures: A review, Autom. Constr. 100 (2019) 163–179.
[4] M.-K. Kim, H. Sohn, C.-C. Chang, Localization and Quantification of Concrete Spalling Defects Using Terrestrial Laser Scanning, J. Comput. Civil Eng. 29 (2015) 04014086.
[5] M. Cabaleiro, R. Lindenbergh, W.F. Gard, P. Arias, J.W.G. van de Kuilen, Algorithm for automatic detection and analysis of cracks in timber beams from LiDAR data, Constr. Build. Mater. 130 (2017) 41–53.
[6] I. Anagnostopoulos, V. Pătrăucean, I. Brilakis, P. Vela, Detection of Walls Floors, and Ceilings in Point Cloud Data, Construct. Res. Cong. 2016 (2016) 2302–2311.
[7] L. Truong-Hong, D.F. Laefer, T. Hinks, H. Carr, Combining an Angle Criterion with Voxelization and the Flying Voxel Method in Reconstructing Building Models from LiDAR Data, Comput.-Aided Civ. Infrastruct. Eng. 28 (2013) 112–129.
[8] I. Puente, R. Lindenbergh, A. Van Natijne, R. Esposito, R. Schipper, Monitoring of progressive damage in buildings using laser scan data, (2018).
[9] H. Son, C. Kim, 3D structural component recognition and modeling method using color and 3D data for construction progress monitoring, Autom. Constr. 19 (2010) 844–854.
[10] F. Bosché, Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction, Adv. Eng. Inf. 24 (2010) 107–118.
[11] Y. Turkan, Tracking of secondary and temporary objects in structural concrete work, Construct. Innov. 14 (2014) 145–167.
[12] J. Jung, C. Stachniss, S. Ju, J. Heo, Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM, Adv. Eng. Inf. 38 (2018) 811–825.
[13] E. Agapaki, G. Miatt, I. Brilakis, Prioritizing object types for modelling existing industrial facilities, Autom. Constr. 96 (2018) 211–223.
[14] Faro, Faro SCENE Software, Accessed by: 15th April, 2021, Available at, https://www.faro.com/en/Products/Software/SCENE-Software.
[15] PointFuse, PointFuse, Accessed by: 15th April, 2021, Available at, https://pointfuse.com/software/.
[16] Trimble, Trimble RealWorks v11.2, Accessed by: 15th April, 2021, Available at, https://geospatial.trimble.com/products-and-solutions/trimble-realworks.
[17] M.E. Esfahani, C. Rausch, M.M. Sharif, Q. Chen, C. Haas, B.T. Adey, Quantitative investigation on the accuracy and precision of Scan-to-BIM under different modelling scenarios, Autom. Constr. 126 (2021), 103686.
[18] L. Díaz-Vilariño, B. Conde, S. Lagüela, H. Lorenzo, Automatic Detection and Segmentation of Columns in As-Built Buildings from Point Clouds, Remote Sensing 7 (11) (2015) 15651–15667.
[19] R. Maalek, D.D. Lichti, J.Y. Ruwanpura, Automatic recognition of common structural elements from point clouds for automated progress monitoring and dimensional quality control in reinforced concrete construction, Remote Sensing 11 (2019) 1102.
[20] H. Fang, F. Lafarge, C. Pan, H. Huang, Floorplan generation from 3D point clouds: A space partitioning approach, ISPRS J. Photogramm. Remote Sens. 175 (2021) 44–55.
[21] X. Xiong, A. Adan, B. Akinci, D. Huber, Automatic creation of semantically rich 3D building models from laser scanner data, Autom. Constr. 31 (2013) 325–337.
[22] M. Bassier, M. Vergauwen, Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data, Autom. Constr. 120 (2020), 103338.
[23] B. Okorn, X. Xiong, B. Akinci, D. Huber, Toward automated modeling of floor plans. Proceedings of the symposium on 3D data processing, visualization and transmission, 2010.
[24] R. Romero-Jarén, J.J. Arranz, Automatic segmentation and classification of BIM elements from point clouds, Autom. Constr. 124 (2021), 103576.
[25] D.F. Laefer, L. Truong-Hong, Toward automatic generation of 3D steel structures for building information modelling, Autom. Constr. 74 (2017) 66–77.
[26] M.-K. Kim, J.P.P. Thedja, H.-L. Chi, D.-E. Lee, Automated rebar diameter classification using point cloud data based machine learning, Autom. Constr. 122 (2021), 103476.
[27] R. Maalek, D. Lichti, J. Ruwanpura, Robust Segmentation of Planar and Linear Features of Terrestrial Laser Scanner Point Clouds Acquired from Construction Sites, Sensors 18 (3) (2018) 819, https://doi.org/10.3390/s18030819.
[28] F. Bosché, E. Guenet, Automating surface flatness control using terrestrial laser scanning and building information models, Autom. Constr. 44 (2014) 212–226.
[29] P.V. Hough, Machine analysis of bubble chamber pictures, Conf. Proc. (1959) 554–558.
[30] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for Point-Cloud Shape Detection, Comput. Graphics Forum 26 (2007) 214–226.
[31] T. Rabbani, F.v.d. Heuvel, G. Vosselmann, Segmentation of point clouds using smoothness constraint, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 36 (2006) 248–253.
[32] A.-V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, ISPRS J. Photogramm. Remote Sens 104 (2015) 88–100.
[33] A. Costin, A. Adibfar, H. Hu, S.S. Chen, Building Information Modeling (BIM) for transportation infrastructure – Literature review, applications, challenges, and recommendations, Autom. Constr. 94 (2018) 257–281.
[34] H. Son, F. Bosché, C. Kim, As-built data acquisition and its use in production monitoring and automated layout of civil infrastructure: A survey, Adv. Eng. Inf. 29 (2) (2015) 172–183.
[35] C. Thomson, J. Boehm, Automatic Geometry Generation from Point Clouds for BIM, Remote Sensing 7 (9) (2015) 11753–11775.
[36] K. Khoshelham, L. Díaz-Vilariño, 3D modelling of interior spaces: Learning the language of indoor architecture, The International Archives of Photogrammetry, Remote Sensing and Spatial, Inf. Sci. 40 (2014) 321.
[37] R. Hulik, M. Spanel, P. Smrz, Z. Materna, Continuous plane detection in point-cloud data based on 3D Hough Transform, J. Vis. Commun. Image Represent. 25 (2014) 86–97.
[38] J. Jung, S. Hong, S. Jeong, S. Kim, H. Cho, S. Hong, J. Heo, Productive modeling for development of as-built BIM of existing indoor structures, Autom. Constr. 42 (2014) 68–77.
[39] G. Zhang, P.A. Vela, P. Karasev, I. Brilakis, A Sparsity-Inducing Optimization-Based Algorithm for Planar Patches Extraction from Noisy Point-Cloud Data, Comput.-Aided Civ. Infrastruct. Eng. 30 (2015) 85–102.
[40] P.H.S. Torr, A. Zisserman, MLESAC: A New Robust Estimator with Application to Estimating Image Geometry, Comput. Vis. Image Underst. 78 (2000) 138–156.
[41] S. Murali, P. Speciale, M.R. Oswald, M. Pollefeys, Indoor Scan2BIM: Building information models of house interiors, IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) 2017 (2017) 6126–6133.
[42] M. Bassier, B. Van Genechten, M. Vergauwen, Classification of sensor independent point cloud data of building objects using random forests, Journal of Building, Engineering 21 (2019) 468–477.
[43] I. Brilakis, Y. Pan, A. Braun, A. Borrmann, Void-Growing: A Novel Scan-to-BIM Method For Manhattan World Buildings From Point Cloud, European Conference on Computing in ConstructionEuropean Conference on Computing in ConstructionRhodes, Greece, 2021, pp. 9.
[44] M.-K. Kim, H. Sohn, C.-C. Chang, Automated dimensional quality assessment of precast concrete panels using terrestrial laser scanning, Autom. Constr. 45 (2014) 163–177.
[45] D. Li, J. Liu, L. Feng, Y. Zhou, P. Liu, Y.F. Chen, Terrestrial laser scanning assisted flatness quality assessment for two different types of concrete surfaces, Measurement 154 (2020), 107436.
[46] Q. Wang, J.C.P. Cheng, H. Sohn, Automated Estimation of Reinforced Precast Concrete Rebar Positions Using Colored Laser Scan Data, Comput.-Aided Civ. Infrastruct. Eng. 32 (2017) 787–802.
[47] R.P. de Souza, C.A. Sierra-Franco, P.I.N. Santos, M. Polonia Rios, D.L. de Mattos Nascimento, A. Barbosa Raposo, Automatic Deformation Detection and Analysis Visualization of 3D Steel Structures in As-Built Point Clouds, in: M. Kurosu (Ed.) Human-Computer Interaction. Design and User Experience, Springer International Publishing, Cham, 2020, pp. 635–654.
[48] J. Chen, Y.K. Cho, Point-to-point comparison method for automated scan-vs-bim deviation detection. 17th International Conference on Computing in Civil and Building EngineeringTampere, 2018.
[49] H. Son, C. Kim, Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity, Adv. Eng. Inf. 34 (2017) 114–124.
[50] Y. Xu, Z. Ye, R. Huang, L. Hoegner, U. Stilla, Robust segmentation and localization of structural planes from photogrammetric point clouds in construction sites, Autom. Constr. 117 (2020), 103206.
[51] The Construction, Minimum height and size standards for rooms in buildings, Accessed by: 25 April, 2021, Available at, https://theconstructor.org/building/rooms-minimum-height-size-standards/5116/.
[52] ACI, ACI 318-08: Building code requirements for structural concrete, ACI Farmington Hills, MI, USA, 2008.
[53] ACI Committee 330, Guide for Design and Construction of Conrete Parking Lots, in: A.-. 330R-8 (Ed.), ACI, UAS, 2008.
[54] ACI Committee 314, ACI 314R-11: Guide to Simplified Design for Reinforced Concrete Buildings, ACI Farmington Hills, MI, USA, 2017.
[55] H. Kim, H. Kim, W. Hong Yong, H. Byun, Detecting Construction Equipment Using a Region-Based Fully Convolutional Network and Transfer Learning, J. Comput. Civil Eng. 32 (2018) 04017082.
[56] H. Samet, R.E. Webber, Hierarchical data structures and algorithms for computer graphics. I. Fundamentals, IEEE Comput. Graph. Appl. 8 (3) (1988) 48–68.
[57] J. Wang, M.M. Oliveira, H. Xie, A.E. Kaufman, Surface reconstruction using oriented charges, Int. Comput. Graph. 2005 (2005) 122–128.
[58] K. Pulli, T. Duchamp, H. Hoppe, J. McDonald, L. Shapiro, W. Stuetzle, Robust meshes from multiple range maps, Proceedings. International Conference on Recent Advances in 3-D Digital Imaging and Modeling (Cat. No.97TB100134), 1997, pp. 205–211.
[59] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, ACM SIGGRAPH Comput. Graph. 26 (2) (1992) 71–78.
[60] L. Truong-Hong, S. Chen, V.L. Cao, D.F. Laefer, Automatic Bridge Deck Damage Using Low Cost UAV-based Images, (2018).
[61] R.M. Haralick, L.G. Shapiro, Computer and robot vision, Addison-wesley Reading1992.
[62] D. Meagher, Geometric modeling using octree encoding, Comput. Graph. Image Process. 19 (1982) 129–147.

[63] Trimble, Trimble TX8 LASER SCANNER, Accessed by: 15th April, 2021, Available at, https://geospatial.trimble.com/products-and-solutions/trimble-tx8.
[64] Leica Geosystems, Leica RTC360 3D Laser Scanner, Accessed by: 24th November, 2020, Available at, https://leica-geosystems.com/products/laser-scanners/scanners/leica-rtc360.
[65] Leica Geosystems, Leica Cyclone REGISTER 360, Accessed by: 24th November, 2020, Available at, https://leica-geosystems.com/products/laser-scanners/software/leica-cyclone/leica-cyclone-register-360.
[66] L. Truong-Hong, D.F. Laefer, Quantitative evaluation strategies for urban 3D model generation from remote sensing data, Comput. Graph. 49 (2015) 82–91.
[67] H. Edelsbrunner, D. Kirkpatrick, R. Seidel, On the shape of a set of points in the plane, IEEE Trans. Inf. Theory 29 (1983) 551–559.
[68] MathWorks, MATLAB Function Reference., 2019b.
[69] F. Poux, C. Mattes, L. Kobbelt, Unsupervised segmentation of indoor 3D point cloud: application to object-based classification, International Archives of the Photogrammetry, Remote Sensing and Spatial, Inf. Sci. 44 (2020) 111–118.
[70] J. Lee, S. Yoo, S. Hong, M.G. Farkoushi, J. Bae, I. Park, H.-G. Sohn, Automated Algorithm for Removing Clutter Objects in MMS Point Cloud for 3D Road Mapping, Sensors 20 (15) (2020) 4076, https://doi.org/10.3390/s20154076.
[71] H. Son, C. Kim, N. Hwang, C. Kim, Y. Kang, Classification of major construction materials in construction environments using ensemble classifiers, Adv. Eng. Inf. 28 (2014) 1–10.