

Automated 3D Wireframe Modeling of Indoor Structures from Point Clouds Using Constrained Least-Squares Adjustment for As-Built BIM

Jaehoon Jung¹; Sungchul Hong²; Sanghyun Yoon³; Jeonghyun Kim⁴; and Joon Heo⁵

Abstract: Automated three-dimensional (3D) modeling of building interiors for an as-built building information model (BIM) incurs critical difficulties because of the complex design of indoor structures and a variety of clutter from scanned point clouds. This paper proposes a scheme for automated 3D geometric modeling of indoor structures, including detailed components such as windows and open doors. Moreover, to produce a regularized model, we imposed constrained least-squares adjustment according to an assumption made with respect to typical indoor structures: walls, ceiling, floor, doors, and windows are composed of straight lines, either parallel or orthogonal. Modeling proceeds in two main phases: wall modeling and window modeling. In the wall modeling phase, the point-cloud acquisitions are projected onto a 2D binary image and the wall boundary is traced out. The boundary is regularized by means of the constrained least-squares method, and, thereby, a 2D floor boundary map can be obtained that later produces, by incorporation of a RANdom sample consensus-based height estimation, a 3D wall model. In the window modeling phase, the wall points are each projected onto a 2D inverse binary image to detect the hollow parts, after which image-based filtering removes incorrect detections. The lines, representing the four window edges, are extracted and adjusted again using the constrained least-squares method. Finally, the window models are projected back onto the original coordinates, and the complete 3D wireframe model is obtained. Two typical indoor environments, a seminar room and a corridor, both at Yonsei University, were chosen as study sites. The generated 3D wireframe models were evaluated by comparison with total station measurements. The 3D positional accuracies of the two results were 0.068 and 0.059 m, respectively, at the 90% confidence level, which indicates the effectiveness of the proposed approach. Finally, using the automatically produced wireframe model as a reference, as-built BIM creation was conducted in the BIM software. DOI: [10.1061/\(ASCE\)CP.1943-5487.0000556](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000556). © 2015 American Society of Civil Engineers.

Author keywords: Three-dimensional (3D) wireframe; As-built building information model (BIM); Indoor modeling; Point clouds; Constrained least squares adjustment.

Introduction

A building information model (BIM) is a digital representation of the physical and functional characteristics of a building (Hichri et al. 2013). However, it is common that building components are not built exactly as specified in their design BIM, either because of construction errors or undocumented changes that might occur during the construction process. After the construction stage, differences

arise from subsequent changes and renovations to a building over time. In addition, buildings often have no design BIM at all, and, in some cases, even blueprints may not be available (Tang et al. 2010; Xiong et al. 2013). For these reasons, the use of BIM requires the condition of a building to be accurately recorded after a building has been constructed (as-built) or the current condition of a building (as-is). The concept of an *as-built* BIM includes both as-built and as-is representations of a building with BIM (Dore and Murphy 2014).

As-built BIM is important for supporting different tasks during the lifecycle of buildings. For example, detection of defects during the construction phase requires timely as-built data. The need for accurate as-built information is also crucial for tasks in further phases of the project, namely, operation and maintenance, because as-designed information deviating from facilities' reality can lead to biased decisions (Liu et al. 2012; Nahangi et al. 2015). The potential benefits of as-built BIM are significant, and include reduced financial risk, location of building components, improved planning for renovation, checking of maintainability, real-time data access and monitoring, energy and space management, emergency management, and cultural heritage preservation (Klein et al. 2012; Volk et al. 2014). In addition, thanks to the recent advances in ubiquitous computing and indoor navigation, the range of application of 3D as-built BIM has been extended from outdoor to indoor environments (Costin and Teizer 2014; Hagedorn et al. 2009; Isikdag et al. 2013; Li et al. 2010). Along with this trend, a growing number of research studies aim at handling the problems of indoor as-built BIM creation involving three main phases: (1) geometric modeling

¹Dept. of Photogrammetry, Univ. of Bonn, Nussallee 15, 53115 Bonn, Germany; formerly, School of Civil and Environmental Engineering, College of Engineering, Yonsei Univ., 134 Shinchondong, Seodaemungu, Seoul 120-749, Korea.

²Korea Institute of Construction Technology, 283 Goyangdae-ro, Ilsanseo-gu, Goyang-si, Gyeonggi-do 411-712, Korea; formerly, School of Civil and Environmental Engineering, College of Engineering, Yonsei Univ., 134 Shinchondong, Seodaemungu, Seoul 120-749, Korea.

³School of Civil and Environmental Engineering, College of Engineering, Yonsei Univ., 134 Shinchondong, Seodaemungu, Seoul 120-749, Korea.

⁴School of Civil and Environmental Engineering, College of Engineering, Yonsei Univ., 134 Shinchondong, Seodaemungu, Seoul 120-749, Korea.

⁵School of Civil and Environmental Engineering, College of Engineering, Yonsei Univ., 134 Shinchondong, Seodaemungu, Seoul 120-749, Korea (corresponding author). E-mail: jheo@yonsei.ac.kr

Note. This manuscript was submitted on June 4, 2014; approved on October 2, 2015; published online on December 23, 2015. Discussion period open until May 23, 2016; separate discussions must be submitted for individual papers. This paper is part of the *Journal of Computing in Civil Engineering*, © ASCE, ISSN 0887-3801.

of components, (2) recognition and attribution of components, and (3) relationship modeling between them (Adán and Huber 2010; Hichri et al. 2013; Hong et al. 2015; Jung et al. 2014; Klein et al. 2011; Tang et al. 2010; Xiong et al. 2013). Among these, the main focus in the present study is to facilitate and enhance the 3D geometric modeling phase.

In order to create a 3D geometric model, the modeler often uses surveying measurements, such as raw point-cloud data, as input (Dore and Murphy 2014; Zhang et al. 2013). Many techniques are available for recording the information of existing structures. In particular, application of as-built BIM with laser scanning is very rapidly expanding within the architecture, engineering, and construction (AEC) industry (Randall 2011) because recent advances in 3D laser scanners allow for rapid capturing of a complex structure's geometry, which produces explicit 3D information in the form of high-density point clouds suitable for detailed 3D modeling (Han et al. 2012a; Pu and Vosselman 2007; Yeung et al. 2014). However, there are some issues with using the point-cloud data in the geometric modeling of indoor environments. First, the huge size of point-cloud data often incurs system slowdown or failure during the modeling process (Jung et al. 2014). Second, manual geometric modeling of detail-rich components is very labor intensive and error prone; in fact, even skilled modelers sometimes produce significantly different results (Xiong et al. 2013; Sepasgozar et al. 2014; Zhang et al. 2013). Third, natural indoor environments, such as offices and seminar rooms, often exhibit high levels of clutter, making it significantly harder to generate as-built models than the more well-studied problem of building shape reconstruction from outdoor scans (Mura et al. 2014). While the demand for a higher degree of automation has increased, no software has yet been developed that ensures a direct conversion from point clouds to a complete 3D as-built model (Budroni and Böhm 2009; Hichri et al. 2013; Pătrăucean et al. 2015). Therefore, the recent focus of indoor as-built creation lies in the automated and detail-rich 3D geometric modeling in the presence of clutter and occlusion.

The present study focuses on the fast and automated provision of 3D indoor wireframe model from point-cloud data that can be further utilized for as-built BIM creations. The scheme for automated 3D geometric modeling of indoor structures includes walls, ceiling, floor, windows, and open doors. Our input is registered and noise-filtered point-cloud data acquired by a laser scanner. The windows and doors are assumed to have a rectangular form with fewer point clouds, either because of specular reflections on the glass or because they remained open during the scanning. The final output is a 3D boundary representation (wireframe model) of the indoor structures, in which a shape is described by a set of straight lines. The proposed approach first models the main structures including the ceiling, floor, and walls, and then extracts the windows and doors and incorporates them into the main model. The principal processes are as follows: projection to a 2D binary image, boundary tracing, segmentation, and regularization. Particularly for the regularization phase, we applied the constrained least-squares method, which adjusts line extractions according to the assumptions that (1) indoor structures are formed primarily of straight lines and (2) all such structures are parallel or perpendicular to each other. After the adjustment, all of the line parameters are corrected together to the most probable values for the given point-cloud observations (Wolf and Ghilani 1997). It should be noted that, excepting parameter selection, the entire wireframe geometric modeling process is fully automated, though the highly detailed and semantic modeling with manual adjustment that meets the conditions specified in the as-built BIM creation are left to be handled by users. The application of the proposed approach for as-built BIM creation is beyond the scope of this paper, and will be continued in our follow-up study.

Related Works

Modeling of building interiors is difficult because of the complex design of indoor structures and various areas of clutter (Budroni and Böhm 2010; Mura et al. 2014; Wang and Cho 2014). To manage this problem, Budroni and Böhm (2010) made some assumptions according to the Manhattan-world domain: (1) horizontal floor and ceiling both orthogonal to the walls; and (2) lack of furniture or objects with irregular geometry. On the basis of these assumptions, they proposed sweep reconstruction and cell decomposition techniques to produce the 3D model. However, the latter assumption is not suitable for natural indoor environments in which clutter and occlusions occur frequently. Okorn et al. (2010) projected point clouds onto a 2D ground plane and created a point-density histogram from which the wall boundary of a room could be extracted automatically by a Hough transform. The approach proved suitable for unmodified, cluttered environments, but could only produce a 2D floor plan. Projection of 3D point clouds, followed by wall boundary tracing, has also been used by Valero et al. (2012). They made a 2D projection of point clouds in voxel space, determined the plane segments in which the points lie, and then generated a 3D boundary representation of the room from the intersecting planes. On the other hand, Mura et al. (2014) extracted candidate wall patches from input scans and projected them on the 2D x-y plane to find separate rooms. Each room polyhedron is finally created by intersecting the reconstructed wall planes. However, both research studies do not provide detailed modeling of doors and windows. Xiong et al. (2013) introduced a context-based method, which is capable of identifying and modeling the structural components without projection of point-cloud data. In addition, the proposed method uses a machine learning algorithm to intelligently estimate the shape of window and doorway openings even when partially occluded by clutter. However, their contribution is limited with simple rectangular-structured rooms.

Having reviewed the recent work, we also found that many of these algorithms extract planar patches from the point clouds to reconstruct the structural components. Although some rigorous planar patch generation methods have been proposed recently (Dimitrov and Golparvar-Fard 2015; Zhang et al. 2015), they are still computationally intensive processes compared to line extractions. Moreover, modeling based on the intersection of adjacent planes is not a mathematically rigorous estimation. Hong et al. (2015) addressed this problem by modeling the structural lines with the constrained least-squares method, but detailed modeling of doors and windows still requires manual operation. For highly complex indoor space modeling with a large amount of point-cloud acquisitions, Jung et al. (2014) proposed a method based on RANdom sample consensus (RANSAC) that proceeds through three steps: segmentation for plane extraction, refinement for removal of noisy points, and boundary tracing for outline extraction. The proposed approach is found to produce detail-rich 3D components including windows and doors, but the traced boundary lines of the 3D plane features are not clear, manifesting as irregularly zigzagged shapes as a result of imperfect segmentation.

As generating an as-built model from point clouds is becoming a common practice, many government agencies and private owners are demanding quality assessment of as-built models (Anil et al. 2013). To verify the accuracy of the model with quantitative measures, Bhatla et al. (2012) calculated the deviations of, for example, the length and width of the bridge's beam by subtracting the ground truth (obtained from the existing 2D drawings) from the modeled elements. However, it is hard to believe that the existing drawing accurately reflects the real condition. Xiong et al. (2013) extracted one side of a modeled opening (doors and windows) and calculated

the difference against the ground truth, but still it is unclear how to obtain the ground truth. In addition, they do not consider the errors of other modeled components, for example, ceiling, floor, and walls. In another case, Anil et al. (2013) proposed a deviation analysis method, which analyzes the patterns of geometric errors between the model and the original point clouds. It can comprehensively evaluate the quality of every modeled surface as long as there are corresponding point-cloud data. Dore and Murphy (2014) selected a total of 30 points on the as-built façade model; recorded x -, y -, and z -coordinates of each point; and calculated the Euclidean distance and root-mean-square error (RMSE) against the coordinates of the matched point on the original point-cloud data. However, the two studies mentioned previously do not address the inherent noise in the point-cloud data. A review of the relevant research indicates the limitation of the current evaluation metric: the absence of a reliable measuring technique for ground truth. Alternatively, Hong et al. (2015) evaluated the final as-built products using a set of highly accurate, independent total station measurements. However, an insufficient number of point measurements because of occlusions is still a problem.

The present study is a significantly improved version of our previous work in Hong et al. (2015). The methodological improvements in data structure, height estimation, segmentation and regularization of traced boundaries achieved significant reduction of processing time, while improving the accuracy; comparison results with the same data set can be found in the Experiments and Results section. Moreover, the previous work and many other research studies have focused primarily on the automation of wall modeling, whereas the present approach can provide detail-rich components, such as windows and doors. The process greatly reduces the manual input, thus allowing for a more efficient modeling of indoor environments. Finally, for the purposes of an evaluation metric, the leave-one-out cross-validation is proposed that provides an almost unbiased solution even with the lack of a sufficient data set. We performed a qualitative analysis of the proposed approach on two indoor spaces that feature more complex layouts compared to the test sites in Hong et al. (2015).

Methods

Overview

The proposed methodology is structured as two main phases (Fig. 1): wall modeling and window modeling. Wall modeling

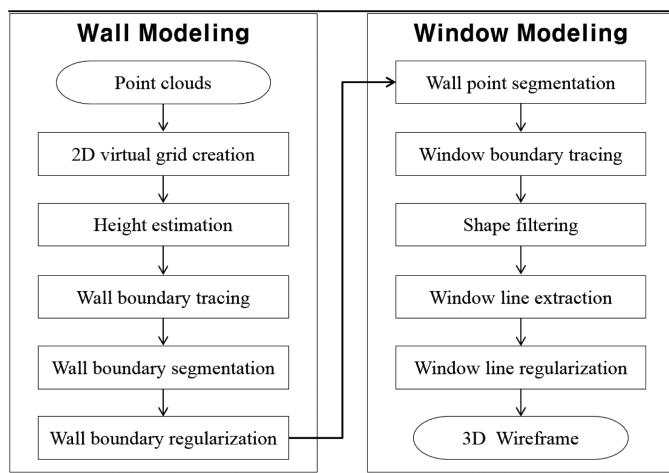


Fig. 1. Process of the automated 3D wireframe model creation

extracts the planes of the main structures (ceiling, floor, walls) and represents them as simplified lines rather than as point clouds. Initially, a 2D virtual grid structure is created to facilitate the indexing of targeted point clouds (Han et al. 2012b). In the height estimation, only the points with the maximum and minimum z -values in each grid cell are extracted, and they are used to estimate the ceiling-to-floor height. Then, all of the point clouds are projected onto a 2D binary image, from which the 2D floor boundary of the room is traced out. This process allows for extracting the structural lines even in the presence of severe clutter and occlusion. However, because of the inherent noise of point locations and low density areas, the traced boundary often has an irregular pattern; thus further refinement with segmentation and regularization is conducted. Finally, by incorporation of the height estimation into the 2D floor boundary, the 3D main structural model is created.

In window modeling, under the assumption that a wall segment's hollow parts are windows or open doors, the point clouds close to each wall are segmented and projected onto an inverse 2D binary image, thus enabling the hollow parts' boundaries to be obtained. However, occlusions caused by clutter also represent hollow parts, which later lead to modeling errors. Therefore, an image-based shape-filtering technique, assuming the rectangular form, is employed to remove incorrect detections. The window's four edges are then extracted and regularized in the same way as in wall modeling. Finally, the window models are projected back to their original 3D coordinates, and a 3D wireframe model of the room is created.

For the creation of an indoor model, Li et al. (2008) proposed five levels of detail (LOD). The lowest level, LOD0, is point-cloud data, created by laser scanners, that consist of a set of vertices with x -, y -, and z -coordinates. A LOD1 model is a surface model resulting from the segmentation of point-cloud data. LOD2 is a block model comprising building spaces, for example, a room space. LOD3 denotes architectural solid models with detailed wall structures. LOD4 completes an LOD3 model by adding 3D objects to the interior building, for example, stairs and furniture. The present approach is relevant to architectural modeling (LOD3), providing 3D wireframe models of walls, ceiling, floor, windows, and open doors. The more detailed components, such as furniture, handles on doors and windows, and ventilators, are considered as clutter, and are thus excluded from the modeling. Also, the generated geometric model does not contain the semantic and volumetric information.

Wall Modeling

2D Virtual Grid Creation

With the incremental enhancements in scanner performance, the size of a point cloud acquired in laser scanning has rapidly increased. Consequently, the process of querying points from a point cloud has become much more time consuming. The problem is that a random point's coordinate is not directly related to its physical address in a storage medium; thus, it cannot be quickly retrieved from a huge amount of point-cloud data. In order to facilitate fast point querying, the two-dimensional virtual grid (2DVG) was created (Han et al. 2012b). Given a minimum boundary rectangle (MBR) from (x_{\max}, y_{\max}) to (x_{\min}, y_{\min}) and cell sizes dx and dy in a two-dimensional coordinate system, a 2DVG is defined as

2DVG

$$= \left[D[u, v] : 0 \leq u \leq \text{INT}\left(\frac{x_{\max} - x_{\min}}{dx}\right), 0 \leq v \leq \text{INT}\left(\frac{y_{\max} - y_{\min}}{dy}\right) \right] \quad (1)$$

where $D[u, v]$ = pointer indicating the storage of a point cloud within the 2DVG; $\text{INT}(t)$ returns a rounded value of t ; and u and v = integer coordinate values. To insert a point $p(x, y, z)$ or to query points around it, the 2DVG coordinates are retrieved from $D[u, v]$, where u and v are resolved as

$$u = \text{INT}\left(\frac{x - x_{\min}}{dx}\right), \quad v = \text{INT}\left(\frac{y - y_{\min}}{dy}\right) \quad (2)$$

Owing to the reduced search space, 2DVG allows for fast point data retrieval (Han et al. 2009, 2012b). 2DVG and not 3DVG was adopted for the purposes of the present study, because with the former, point retrievals are conducted primarily in the process of wall point segmentation, and wall points in a grid cell are searched and segmented together regardless of z -value.

Height Estimation

After 2DVG creation, ceiling-to-floor height estimation is conducted. In building interiors, it is typical that ceilings and floors are mostly planar and parallel structures located at the highest and lowest heights, respectively (Valero et al. 2012). The points with the maximum and minimum z -values in each grid cell are chosen for ceiling and floor height estimation, respectively. In order to reduce the computational complexity and failure rate of plane extraction in the 3D context, line segmentation is applied only for (z, x) or (z, y) values. For line detection and segmentation, the RANSAC algorithm was adopted, owing to its robustness against outliers (Boulaassal et al. 2007; Fischler and Bolles 1981; Karsli and Kahya 2008). RANSAC includes two phases: (1) hypothesis and (2) test. In the hypothesis phase, two points are randomly selected from the set of points, and the line-model parameters are initialized. In the test phase, the error is defined according to the orthogonal distance of the other points with respect to the line parameters, and the points within the threshold distance (0.005 m) from the calculated line are considered as potential inliers. If the number of those inliers is greater than that from the previous iterations, the inliers in the current iteration become the new candidate for the final output segment. This procedure is repeated iteratively until the maximum number of iterations (500) is reached (Heo et al. 2013; Hong et al. 2015). Fig. 2 provides the segmented result, the bright dots indicating the unsegmented points, and the dark dots representing the ceiling and floor segments. Finally, the ceiling-to-floor height is estimated from the height difference between two mean z -values of each segment.

Wall Boundary Tracing

Initially, by setting the height value (z -value) to zero, all of the point clouds constituting the room are projected onto a 2D x - y plane according to a defined grid size. Next, the 2D point clouds are converted into a binary image wherein the occupied pixels represent 1 and the others 0. Assuming that the most outer lines of the occupied pixels constitute a 2D floor boundary, boundary tracing is then performed to extract numerous line segments. The tracing begins searching from the leftmost occupied pixel in the first row of a binary image. Then, the next adjacent boundary pixels are sequentially searched in the clockwise direction (Jung et al. 2014). The traced boundary is usually irregular, because of the inherent noise of point locations and low density areas; thus line regularization must follow so as to produce clear outlines.

Wall Boundary Segmentation

Prior to regularization, each line constituting the boundary should be segmented. In the present study, the Douglas-Peucker algorithm, which searches a 2D floor boundary's corner points and identifies two connecting points as a line segment matched to a wall, was used. It initially marks the first and last points to be kept. Then,

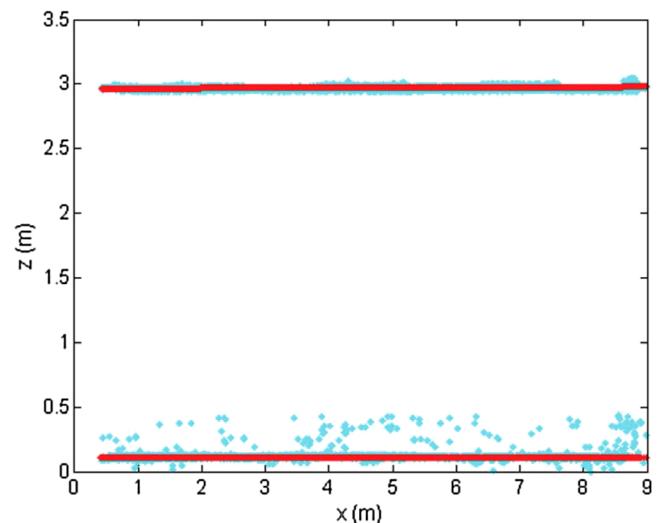


Fig. 2. Ceiling-to-floor height estimation by RANSAC algorithm

it finds the point that is furthest from the line segment with the first and last points as endpoints. If the point is greater than the threshold distance (ε) to the line segment [Fig. 3(a)], the point is kept for use as the next endpoint. Otherwise, the point closer than the threshold distance is included in the line segment [Figs. 3(b and c)]. The algorithm operates recursively until all of the points are tested and segmented. Subsequently, new output lines are produced from the points that have been kept [Fig. 3(d)] (Douglas and Peucker 1973). However, if the effect of location errors on the points is severe, particularly for the presence of noise in the wall corners, the algorithm can produce incorrect segmentation results; necessarily then, a further refinement process is followed for two line segments intersecting at a right angle: if the latter line is shorter than the threshold distance or the former line is connected to the latter at an angle of -45° , then the two line segments are merged into one. All lines are sequentially tested; finally, the last line segment is checked with the first one.

Wall Boundary Regularization

The regularization process adjusts segmented lines by means of the constrained least-squares method. As a regularization boundary, it

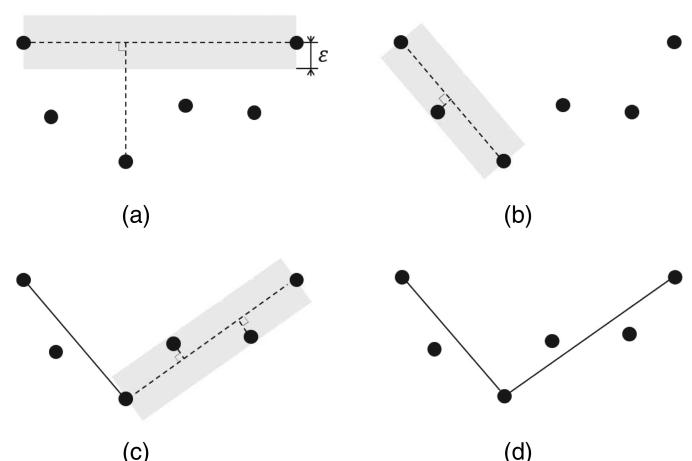


Fig. 3. Process of Douglas-Peucker algorithm: (a) selection of the furthest point; (b) integration of the furthest point into the line segment; (c) recursion of point segmentation; (d) newly segmented output lines

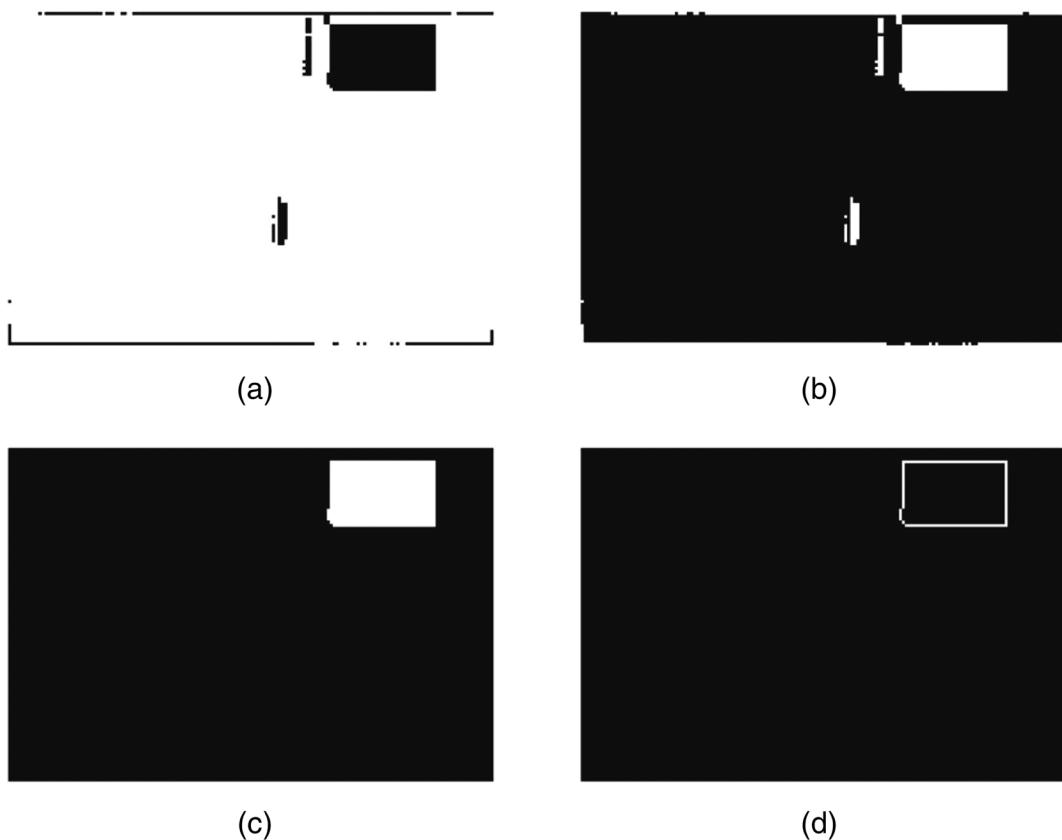


Fig. 4. Window boundary tracing: (a) projection of wall points; (b) creation of inverse binary image; (c) removal of small segments; (d) traced window lines

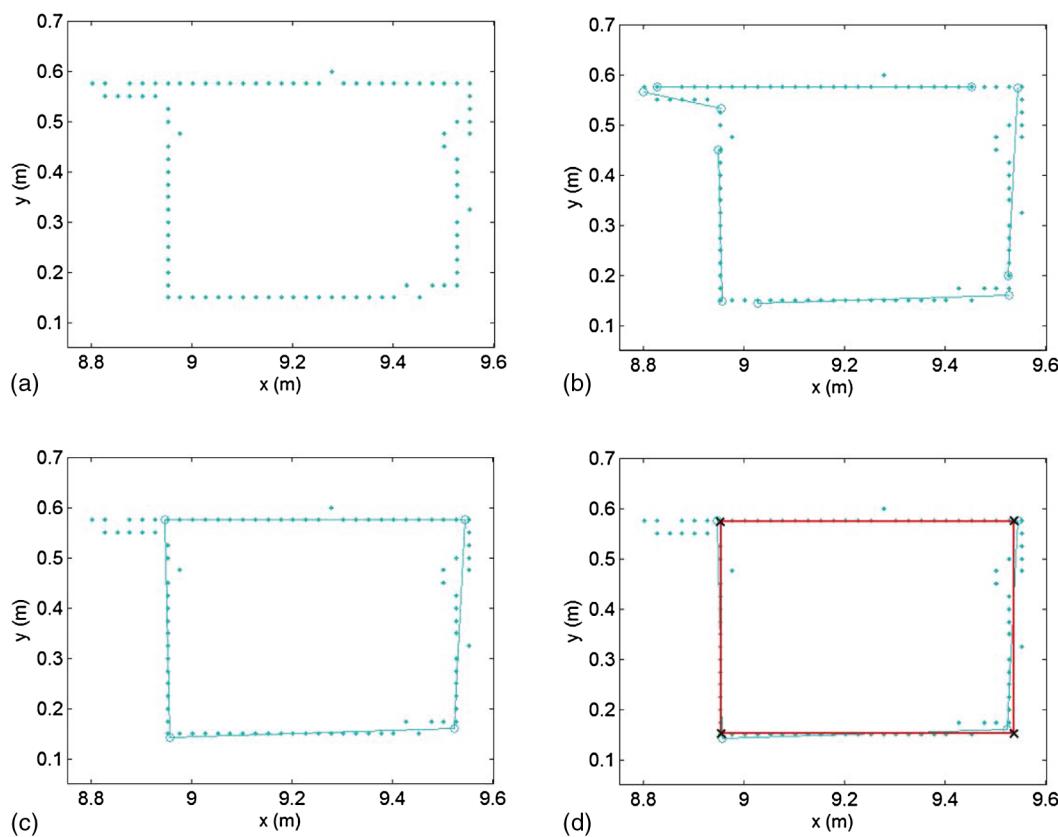


Fig. 5. Window line extraction and regularization: (a) points consisting of window boundary; (b) line extraction by incremental algorithm; (c) fitting of rectangle with largest line segments on each side; (d) adjustment of line extractions

is assumed that straight line segments satisfy geometric conditions such as orthogonality or parallelism. Commonly used in line extraction is the Hessian model (Navarro et al. 2007; Zhang et al. 2008)

$$x \cos \alpha + y \sin \alpha - r = 0 \quad (3)$$

where x and y = projected coordinates constituting the boundary and line parameters $L = [\alpha \ r]^T$ = orientation and distance from the origin, respectively. Provided that the first line L_1^r is the reference (a line including the greatest number of point clouds), the conditional rules for the other orthogonal and parallel lines L_i are defined as

$$\text{if } |\alpha_1^r - \alpha_i| - \pi/2 < \theta, \quad L_i \text{ is orthogonal} \quad i = 2, \dots, n \quad (4)$$

$$\begin{aligned} \text{if } |\alpha_1^r - \alpha_i| - \pi < \theta \quad \text{or} \quad |\alpha_1^r - \alpha_i| < \theta, \\ L_i \text{ is parallel} \quad i = 2, \dots, n \end{aligned} \quad (5)$$

where α_1^r and α_i = orientation of the reference and the other line, respectively (ranging from $-\pi$ to π) and θ = empirically determined threshold (10° was used in this study). In Eq. (5), the former condition indicates that the other line L_i is located on the same side as the reference line L_1^r , and the latter condition, vice versa.

For constrained adjustment of line parameters, the nonlinear least-squares method is used. The normal matrix and its matching constraints matrix are formed as

$$\begin{bmatrix} J^T W J & K^T \\ K & 0 \end{bmatrix} \begin{bmatrix} \hat{\xi} \\ \lambda \end{bmatrix} = \begin{bmatrix} J^T W \tau \\ \kappa_0 \end{bmatrix} \quad (6)$$

where J = Jacobian matrix that contains the coefficients of the linearized observations equations [Eq. (3)] with respect to $L = [\alpha \ r]^T$; W = weight matrix; and τ represents the observed minus the computed values. Observation equations for the constraints are included in the normal matrix as additional rows K and columns K^T , and their constants $\pm\pi/2$ (orthogonal) or $\pm\pi$ (parallel) are added to the constraints matrix as additional rows κ_0 . $\hat{\xi}$ = correction for the line parameters while $\hat{\lambda}$ = Lagrangian multiplier. Finally, the correction $\hat{\xi}$ and the dispersion $D\{\hat{\xi}\}$ (the adjusted covariance) of the line parameters are calculated as

$$\hat{\xi} = N^{-1}c + N^{-1}K^T(KN^{-1}K^T)^{-1}(\kappa_0 - KN^{-1}c) \quad (7)$$

$$D\{\hat{\xi}\} = \sigma_0^2[N^{-1} - N^{-1}K^T(KN^{-1}K^T)^{-1}KN^{-1}] \quad (8)$$

where σ_0^2 = reference variance; N = normal matrix $J^T W J$; and $c = J^T W \tau$. The process is repeated until the corrections become sufficiently small. The least-squares adjustment allows for inclusion of all of the observed points in the adjustment and simultaneously computes them, thereby ensuring the consistency of all line parameters; moreover, it produces the most probable values satisfying all of the geometric constraints for the given data set (Snow 2009; Wolf and Ghilani 1997). After the adjustment, the segmented lines should satisfy the conditional rules in Eqs. (4) and (5). Finally, the wall corners are recalculated by intersecting two adjacent lines



Fig. 6. Test sites (images by authors): (a and b) are the seminar room, with the highlighted portion in (a) indicating the conduit passing through the ceiling and floor; (c and d) are the corridor, both at Yonsei University

forming a right angle. The produced wall models will always be composed of a set of the most probable perpendicular lines to the previous and the next line segments. Still, exceptional problems can occur: if some incorrectly segmented lines are forced to be adjusted according to the parallel or orthogonal constraints, the geometry of the adjusted lines can be broken, producing a line considerably longer than its original size, primarily because of noisy points or low-density areas; thus, in each case, careful noise removal or re-scanning of the study area is required. The 2D floor boundary map with regularized lines having been obtained, then, by incorporating the height estimation, the 3D wireframe model of the main structure (floor, ceiling, walls) is produced.

Window Modeling

Wall Point Segmentation

Assuming vertical walls to be of a rectangular form, each line segment, which is composed of two adjacent corner points (c_i, c_{i+1}) in the regularized 2D floor boundary map, can be used for wall point segmentation. First, walls smaller than a minimum window size are removed by calculating the Euclidean distance between two corner points constituting a line segment. Then, to classify the scanned points belonging to each line segment, the conditional rules are defined as

$$d_{p_i} \leq \delta \quad \& \quad \text{len}(c_j, c_{j+1}) \geq [\text{len}(f_{p_i}, c_j) + \text{len}(f_{p_i}, c_{j+1})] \quad (9)$$

where the former condition calculates the orthogonal distance (d_{p_i}) of the i th scanned point (P_i) with respect to the j th line segment and then determines whether the scanned point is closer to the line segment than the predefined threshold distance (δ), and the latter condition determines whether the perpendicular foot (f_{p_i}) of the i th scanned point lies on the line segment by calculating the length (len) of it with respect to the line segment's corner points. Only scanned points satisfying both conditions are classified to a wall point group. In order to determine the threshold distance, the modeller needs to have some background knowledge of the indoor environments: for precise extraction of window boundaries, the distance should be greater than the thickness of the window frames, and to avoid the occlusion, it should be shorter than the minimum distance to the nearest clutter.

Window Boundary Tracing

After wall point segmentation, window boundary tracing is conducted. Initially, each segmented wall point-cloud data set was rotated along the z -axis, and then was rotated again along the y -axis. Consequently, the segmented wall point-cloud data could be projected onto a 2D x - y plane according to a defined grid size [Fig. 4(a)]; however, in this case, an inverse binary image is produced that represents a plane segment's hollow parts (e.g., windows and open doors) as 1 and the originally occupied-pixel parts as 0 [Fig. 4(b)]. Then, the small segments are removed [Fig. 4(c)], and boundary tracing of the remaining hollow segments is conducted in the same way as in wall modeling [Fig. 4(d)].

Shape Filtering

The window detections in the boundary tracing phase are verified according to the property of the ratio between the square of the perimeter and the area (Peri2Area). Any particular shape has a unique parameter value that is a constant for that shape (Admuthe and Bendre 2012). For example, if the value is between 9 and 11.75, the shape is considered to be octagonal; for values from 11.8 to 14, circular; from 14.1 to 15.77, pentagonal; from 15.78 to 19.14, rectangular; and from 19.15 to 23, triangular. Any value less than 9 or greater than 23 is considered an indication of an

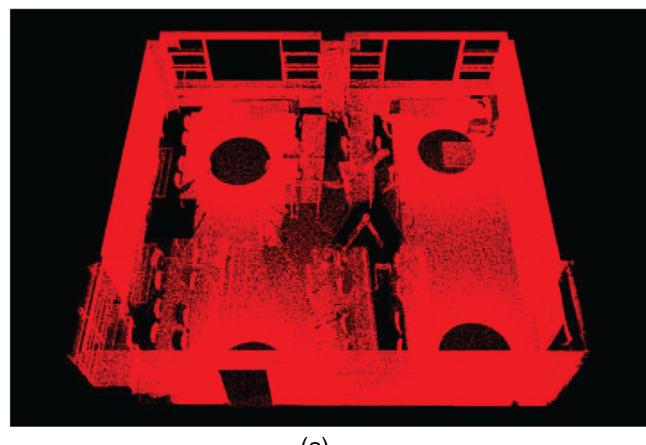
irregular shape (Khan et al. 2011). Assuming windows and doors to be of the rectangular form, Peri2Area filtering finds rectangular objects in the segmented wall.

Window Line Extraction and Regularization

An incremental algorithm is utilized to extract the four lines of rectangular windows and doors. It begins by constructing a line with the first three points using a least-squares method for line fitting. It

Table 1. FARO Focus 3D Specifications

Categories	Specifications
Measurement type	Phase difference
Laser wavelength	905 nm
Max scan rate	976,000 pps
FOV (V/H)	305°/360°
Range accuracy	±2 mm @25 m
Angular accuracy	0.015°



(a)



(b)

Fig. 7. Point cloud acquisitions: (a) the seminar room; (b) the corridor

Table 2. Test-Computer Specifications

Categories	Specifications
CPU	Intel® Core™ i5-2500 CPU @ 3.30GHz
Memory	8.00 GB
OS	Windows 7 64-bit
Tools	MATLAB 8.0, Autodesk Recap

then adds the next points to the current line model, and recomputes the line parameters (typically, more than three points are added to accelerate the incremental process). The consistency between current line segment L_i and new line segment L_j is calculated according to the Mahalanobis distance

$$\chi_{\gamma,n}^2 \geq (L_i - L_j)^T (C_i + C_j)^{-1} (L_i - L_j) \quad (10)$$

where L_i and L_j = line parameters (orientation and distance); C_i and C_j = respective covariance matrices of the lines; and $\chi_{\gamma,n}^2$ = number taken from a χ^2 distribution with $n = 2$ degrees of freedom

Table 3. Time Lapses for 2DVG Creation and Modeling Operation according to Different Virtual Grid Cell Sizes

Cell size (m)	Seminar room		Corridor	
	2DVG creation (s)	Modeling (s)	2DVG creation (s)	Modeling (s)
0.1	425.2	14.3	647.7	26.5
0.2	143.5	9.3	230.2	17.9
0.3	93.1	7.0	150.0	13.9
0.4	75.3	6.7	121.2	13.1
0.5	67.1	6.5	107.9	12.6
0.6	62.8	6.9	98.7	12.6
0.7	59.1	6.7	93.6	12.6
0.8	57.4	6.7	90.3	12.3
0.9	55.8	6.6	90.6	12.5
1.0	55.0	6.8	88.2	12.2

and probability level γ of rejection of the hypothesis of pairing correctness (Arras et al. 2001). If the condition is satisfied, the algorithm continues to add new points, or it puts back the last points and computes each point's distance to the current line model. If the distances of three continuous points are each greater than a predefined threshold, a new line starts with those points; otherwise, they are removed as outliers (Nguyen et al. 2007; Siadat et al. 1997). Fig. 5(a) shows the original points constituting a window boundary. By employing incremental line extractions, five lines are initially extracted, as indicated in Fig. 5(b). The window is then fitted to a rectangle by choosing the longest line segments on each side and finding the four intersecting corner points, as in Fig. 5(c). Finally, the lines are adjusted using the constrained least-squares method described in "Wall Boundary Regularization," and a window consisting of regularized lines can be obtained [Fig. 5(d)].

Experiments and Results

Study Sites and Data Acquisitions

We conducted a pilot implementation of the proposed approach for a seminar room [Figs. 6(a and b)] and a corridor [Figs. 6(c and d)] at Yonsei University. The seminar room is a relatively small and simple structure including much clutter in the form of desks and chairs, whereas the corridor is more complex because of several pillars and a corner, though there is less clutter. In the scanning process, FARO Focus 3D (FARO, Lake Mary, Florida), a

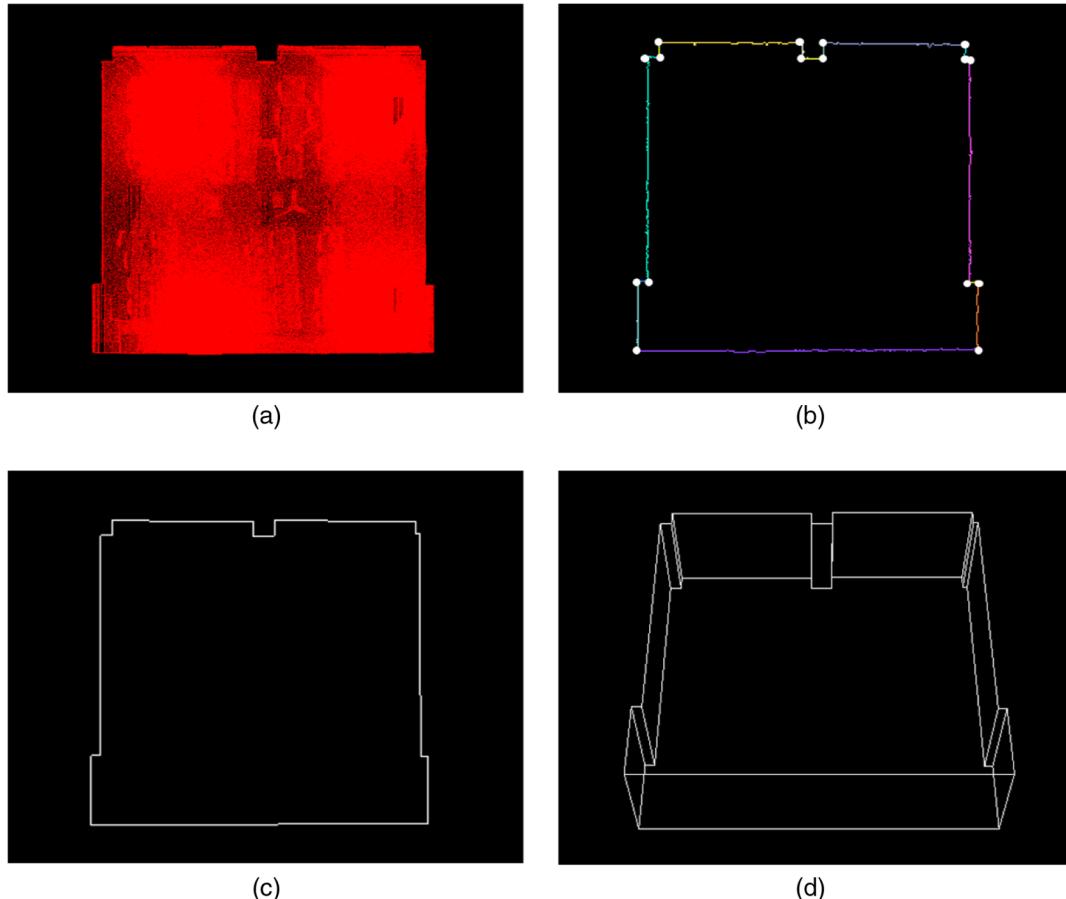


Fig. 8. Wall modeling process: (a) 2D projection of point clouds; (b) 2D floor boundary tracing and segmentation; (c) regularization by the constrained least-squares method; (d) 3D wireframe wall model creation

phase-based laser scanner, was employed to collect point-cloud data. The specifications of the FARO Focus 3D are listed in Table 1. The laser scanner was set up at several locations; nonetheless, the point-cloud acquisitions inevitably included occlusions attributable to clutter. After mosaicking individual point clouds, approximately 10 million points remained for the seminar room [Fig. 7(a)] and 15 million points for the corridor [Fig. 7(b)]. For the development of the algorithm and the removal of noisy points, *MATLAB 8.0* and *Autodesk Recap* software were used, respectively. The test computer specifications are listed in Table 2.

Details of Wall Modeling

Initially, the point-cloud acquisitions of two test sites were stored and organized in the 2DVG to facilitate point retrieval. Specifically, owing to 2DVG's reduction of the search space, the retrieval operation could be accelerated. Then, since the virtual grid cell size directly influences the performance with respect to specific 2DVG grids, 2DVG was evaluated according to different cell sizes ranging from 0.1 to 1.0 m (at 0.1-m intervals) (Table 3). At the beginning, as the cell size increased, the time lapse for 2DVG data structure creation and retrieval in the modeling drastically decreased. With a cell size of 0.5 m, however, the performance slowed down, because the concentration of more points into a smaller number of cells increased the operations on each cell (Han et al. 2012b). Thereafter, larger cell sizes produced fewer maximum and minimum points for height estimation. Accordingly, we adopted the cell size of 0.5 m, which showed the peak performance in both sets of test data.

Fig. 8 illustrates the wall modeling process for the seminar room. The projection of the 3D point clouds [Fig. 8(a)] produces a 2D binary image in which each pixel is occupied by at least one or more 3D points. In the wall boundary tracing, a small tracing grid size produces a precise boundary, but it is greatly influenced by noisy points and, as such, requires excessive processing time. Conversely, a larger tracing grid size produces simpler lines within a shorter time, but a too-large tracing grid size leads to large distortions resulting from the inability to classify small segments. In order to select the optimal tracing grid size, Heo et al. (2013) and Jung et al. (2014) proposed the halving of the size of detectable objects: at each test site, the minimum line was measured, similarly, to 0.13 m for the seminar room and 0.14 m for the corridor; thus, the tracing grid size would be recommended to be less than 0.065 and 0.07 m, respectively. Moreover, given the present study's purpose to precisely model the test sites in as short a time as possible (less than 1 min), the tracing grid size of 0.02 m finally was chosen for both test sites, which proved less susceptible to noisy points while preserving real detail.

Since the initial floor boundary extraction was likely to have irregular patterns, further refinement, in the form of segmentation and regularization, followed. In segmentation by the Douglas-Peucker algorithm, a shorter threshold distance is appropriate for a complex floor boundary segmentation, whereas a too-short distance can produce an irregularly segmented pattern that later incurs failure in the refinement process. In addition, because of the effect of the tracing grid size on line-pattern irregularity, a threshold distance greater

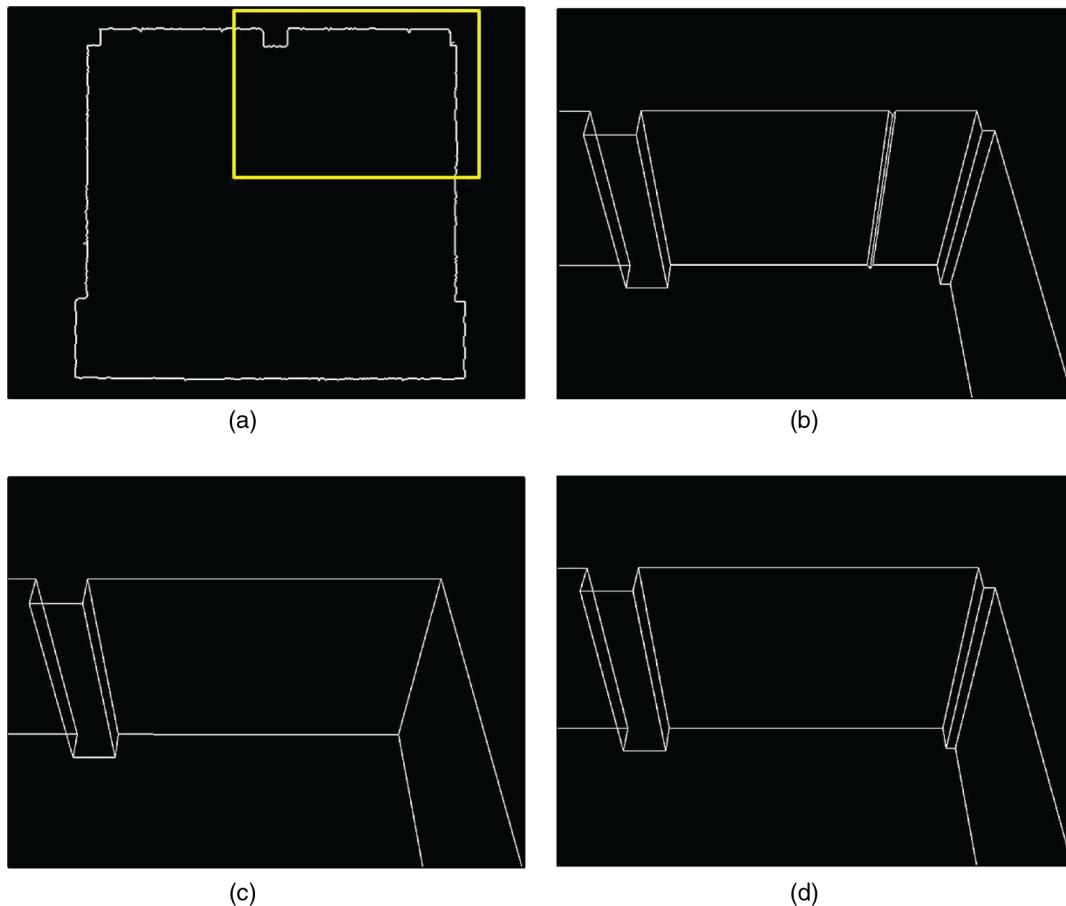


Fig. 9. Examples of the modeled walls with different threshold distances in Douglas-Peucker segmentation: (a) traced boundary of the seminar room, with the highlighted portion modeled by (b) too-short threshold distance (0.07 m); (c) too-wide threshold distance (0.13 m); (d) optimal threshold

than the tracing grid size is desired. A wider threshold produces a simpler boundary without failure, but if the threshold is equal to or greater than the minimum size of the line components, the floor boundary will be oversimplified (Hong et al. 2015). Therefore, the optimal threshold distance is found between the minimum size of the detectable line component and the tracing grid size we defined. Fig. 9(a) illustrates an example of traced boundary, and Figs. 9(b–d), which correspond to the highlighted portion in Fig. 9(a), show examples of the modeled walls with different threshold distances. If the effect of noisy points is greater than the

threshold distance, the shape of wall model can be distorted as illustrated in Fig. 9(b). Conversely, Fig. 9(c) depicts a case in which the shape of wall model is oversimplified with too-wide threshold distance. Fig. 9(d) shows an example of the properly modeled wall with an optimal threshold distance. At each test site, the minimum line was measured, similarly, to 0.13 m for the seminar room and 0.14 m for the corridor, and the same tracing grid size of 0.02 m was used for both test sites; however, in consideration of the effects of noisy points and oversimplification, different threshold distances, 0.1 m for the seminar room and 0.07 m for the complex corridor, were finally chosen. Fig. 8(b) provides an example of the traced and segmented 2D floor boundary of the seminar room. A white dot represents each endpoint of the segmented boundary. During the seminar room modeling, the main factor leading to incorrect floor boundary tracing was the white conduit passing through the ceiling and floor [the red circle in Fig. 6(a)], which was thicker than the threshold distance (0.1 m) in Douglas-Peucker segmentation and thus traced out together as a column structure. As for the corridor modeling, no such objects were encountered.

The regularization process was then employed for the line segments by means of the constrained least-squares method. Assuming a typical indoor environment, the parallel or orthogonal constraint between line segments was imposed. The weight was computed according to the number of points in each segmented line, which permitted longer lines to have a dominant influence on the line parameter adjustment. Subsequently, the intersecting points of two adjacent lines forming a right angle were extracted and connected as in Fig. 8(c). Finally, the ceiling-to-floor height, as estimated by RANSAC segmentation, was incorporated into the

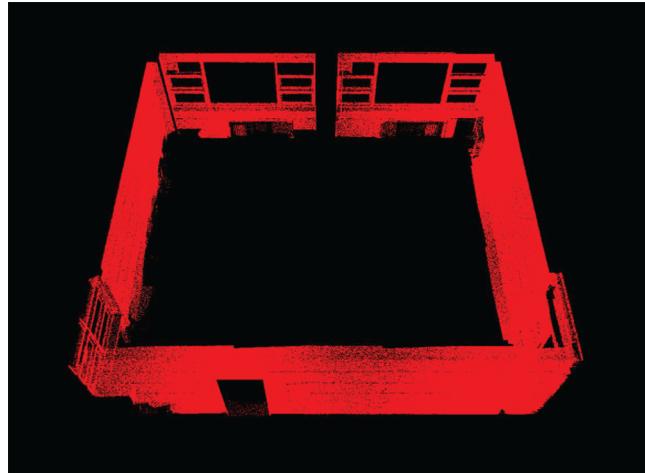


Fig. 10. Wall point segmentation result for the seminar room

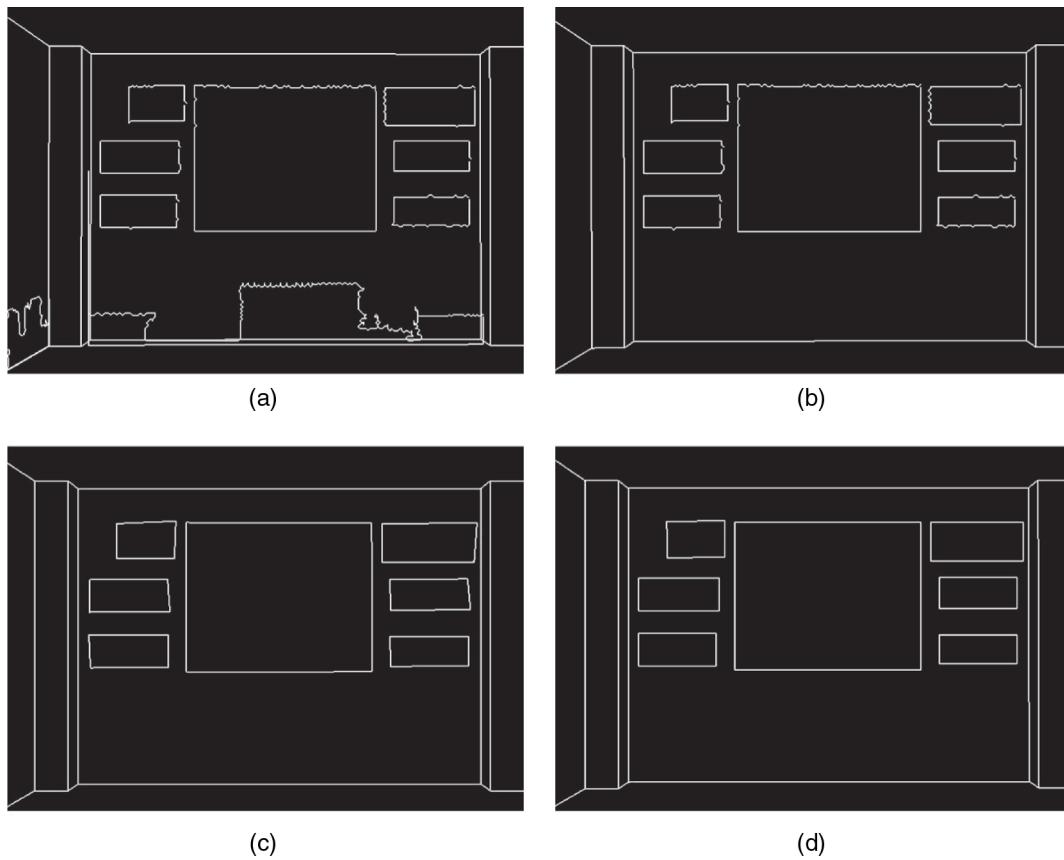


Fig. 11. Window modeling process: (a) wall point segmentation; (b) boundary tracing of windows and doors; (c) remaining windows after shape filtering; (d) filtering of irregular window model after adjustment

regularized 2D floor boundary map. The 3D wireframe wall model of the seminar room was created as shown in Fig. 8(d).

Details of Window Modeling

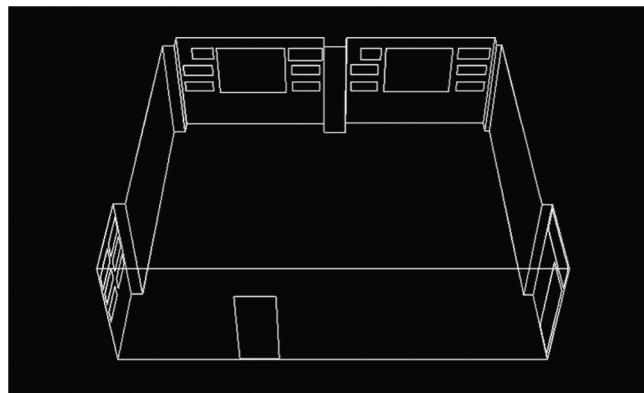
Prior to the window modeling, wall point segmentation was conducted. Assuming a vertical and rectangular wall, each wall was defined as a line segment consisting of two intersecting corner points in the regularized 2D floor boundary map. The scanned points, which were closer to the line segment than the predefined threshold distance (0.3 m), were then retrieved and saved for the window modeling. The 2DVG was especially useful in this phase, because it enables a reduced retrieval time. As indicated in Fig. 10, only the segmented wall points remained after the other points, representing the ceiling, floor, and pillars, were removed.

The wall points were then projected and converted to an inverse binary image to trace the hollow areas. If the tracing grid size is smaller than the scanned points' interval, the connectivity can be broken, resulting in the exclusion of certain parts of a window from the traced boundary. Hence, considering the lowest-density areas of the two test sites, the tracing grid size of 0.02 m was chosen for the window modeling. Fig. 11(a) illustrates the traced boundaries of windows in a wall. The figure also includes many irregular components attributable to clutter and occlusions. Since irregular components typically have a high perimeter-to-area ratio because of their complex boundaries, they thus can be filtered by employing an image-based shape-filtering algorithm. In the experiment, the total number of pixels was counted as the area, and the sum of the distances between adjacent pixels in the boundary was counted as

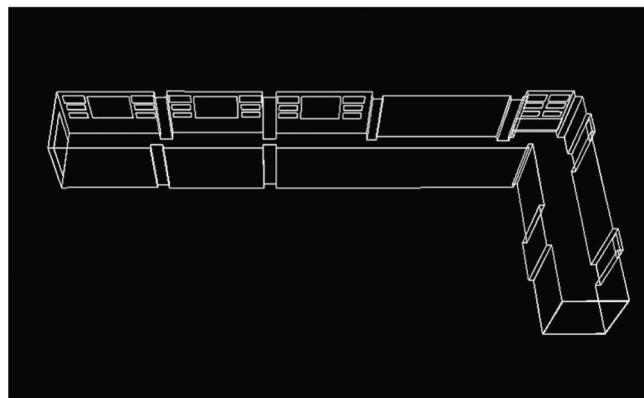
the perimeter of the window. The properly filtered result is shown in Fig. 11(b). However, as was the case with the wall modeling, the traced boundary still showed a zigzag pattern. Further refinement of the projected boundary coordinates by line extraction and regularization followed.

In the line extraction by the incremental algorithm, at each step, the consistency of the current and new line segments was tested using the Mahalanobis distance at the 95% confidence level. When the new line segment did not satisfy the condition, the corresponding points were each tested to determine if their distances to the current line model were greater than the predefined threshold (the window tracing grid size was used). Although two points are sufficient to construct a line model, we limited the number of minimum continuous points to three so as to ensure correct line extraction; otherwise, the points were removed as outliers. Thus, only the longest line segments on each side of a window were selected and intersected, as shown in Fig. 11(c).

Finally, the extracted lines were regularized by imposing the parallel or orthogonal constraint [Fig. 11(d)]. During the adjustment, the windows that did not satisfy the right-angle condition for two adjacent lines were filtered out as noise. Then, the adjusted and filtered window models were projected back to their original 3D coordinates, and the 3D wireframe model of the seminar room and the corridor were created as shown in Fig. 12. We also overlapped the generated 3D wireframe models on the corresponding point clouds, as indicated in Fig. 13 (seminar room) and Fig. 14 (corridor). Note that since the laser scanner only collected the



(a)



(b)

Fig. 12. Final 3D wireframe models of (a) the seminar room; (b) the corridor

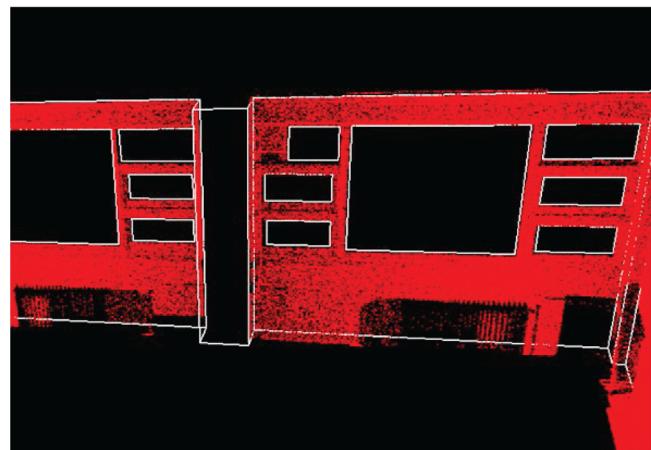
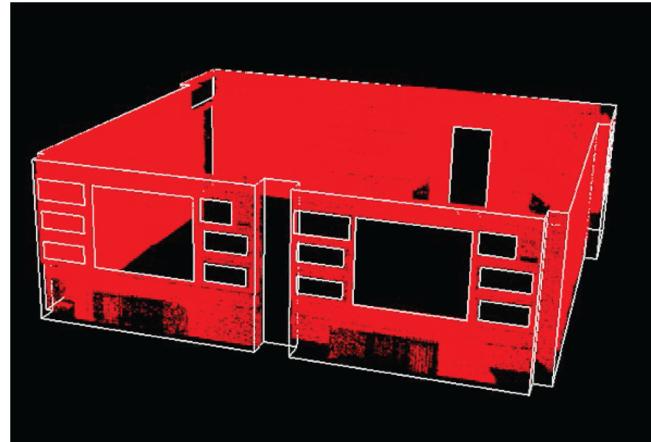


Fig. 13. 3D wireframe model of the seminar room overlaid with the original point clouds

point-cloud data of the surface of the lockers, the generated wireframe presented the lockers as the wall and did not represent the wall behind them.

Table 4 lists each modeling phase with the corresponding time consumption. Two people were involved in operating the scanner, whereas only one person was involved in all of the other phases.

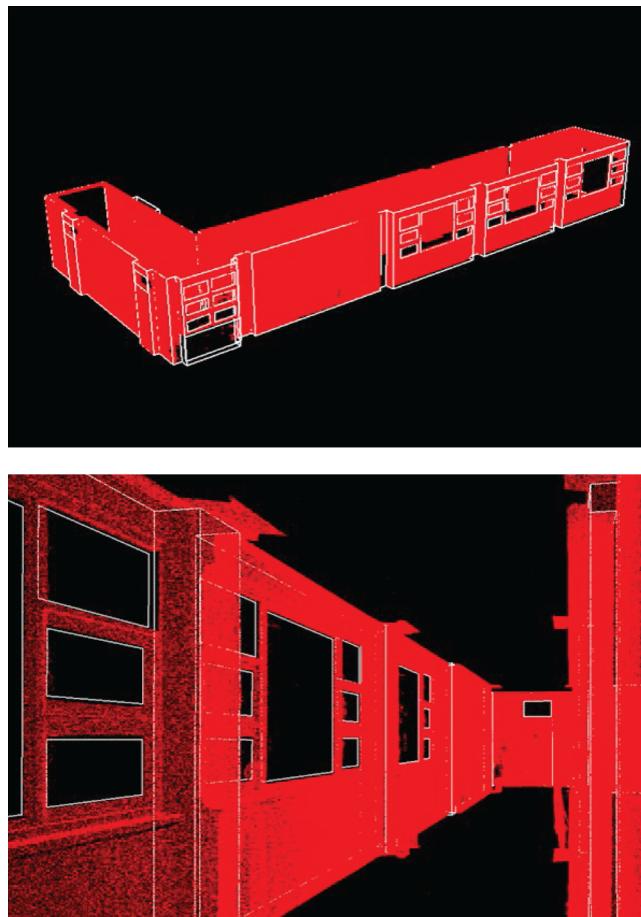


Fig. 14. 3D wireframe model of the corridor overlaid with the original point clouds

The following three phases were manually carried out: (1) acquisition of point clouds using a laser scanner; (2) registration of the scanned points in *FaroScene* software; (3) removal of noisy points—primarily attributable to instrumental errors (e.g., in distance or angle measurement) or natural constraints (e.g., surface properties and angles of incidence) (Schulz et al. 2005); and (4) as-built BIM creation using the 3D wireframe model in *AutoCAD Recap* software—this will be discussed in detail in the later section, “As-Built BIM Creation.” Overall, acquisition and preprocessing of the point-cloud data consumed most of the time. By contrast, automated 3D modeling required only less than 2 min in each case. Compared to the processing time (12.6 min for the seminar room and 18.9 min for the corridor) achieved by our previous algorithm (Hong et al. 2015), the present approach, even though a further process of automated window modeling is incorporated, showed a great time reduction rate: to 9.5 and 6.3% of the previous work, respectively. The enhancement in the processing time was primarily achieved because of the fast point data retrieval by the 2D virtual grid and the simpler line segmentation for the height estimation. The metric quality of the proposed approach also was assessed, as discussed in the next section.

Accuracy Assessments

The metric quality of the two 3D wireframe models was validated on the basis of measurements acquired by a total station. The evaluation was scored on the basis of well-distributed and distinct points such as the corners of walls, doors, and windows. First, the accuracy of the model was assessed according to the Euclidean average distance error (δ_{avg})

$$\delta_{\text{avg}} = \frac{1}{n} \sum_{i=1}^n |Ra_i - T - b_i| \quad (11)$$

where a_i = i th point in the 3D wireframe model; b_i = corresponding point measured by the total station; n = number of point measurements; and R and T = rotation and translation parameters, respectively, for 3D Helmert transformation (the scale was not considered in this comparison) (Reit 1998). In practice, except for some corner points which were blocked by the window screen or the mosquito bar, a total of 73 points were extracted from the seminar room, and 94 points were extracted from the corridor. Instead of splitting the

Table 4. Modeling Phases with Time Consumption

Phase	Time (min)		Notes
	Seminar room	Corridor	
Manual processing			
Scanning of structures	38	60	This phase included selection of the areas to be scanned, location of the scanner and targets (for registration), and scanning of the areas
Data registration	12	15	Multiple point-cloud acquisitions were registered as single coordinates by the commercial software <i>FaroScene 4.8</i>
Noise removal	34	51	The noisy points were removed manually in the commercial software <i>AutoCAD Recap</i>
As-built BIM creation	24	36	By using the 3D wireframe model as input, as-built BIM was created in the commercial software <i>AutoCAD Revit Architecture 2014</i>
Time consumed for manual process	108	162	—
Automatic processing			
2DVG creation	1.1	1.8	The 2D virtual grid data structure was created in this phase
Automated 3D wireframe modeling	0.1	0.2	The 3D wireframe model of the structure was created and exported in the CAD-oriented file format
Time consumed for automatic process	1.2	2.0	—
Total time consumed	109.2	164	—

Table 5. Accuracy Assessment Results for the Seminar Room

Error type	x (m)	y (m)	z (m)	Total (m)
Average error	0.025	0.014	0.019	0.037
RMSE	0.036	0.024	0.021	0.048
SAS	—	—	—	0.068

Table 6. Accuracy Assessment Results for the Corridor

Error type	x (m)	y (m)	z (m)	Total (m)
Average error	0.017	0.023	0.013	0.036
RMSE	0.021	0.035	0.015	0.043
SAS	—	—	—	0.059

extracted points into parameter estimation and validation, the leave-one-out cross-validation, which provides an unbiased and reliable solution, is applied (Cawley and Talbot 2004; Varma and Simon 2006). In the process, one point was retained for validation, and the remaining points were used for parameter estimation. The error vectors in the x , y , and z directions of the transformed point were then calculated for validation. This process was repeated n times until all points were validated, and the results are listed in Table 5 (seminar room) and Table 6 (corridor). The average error indicates that the overall accuracies for the seminar room and corridor were 0.037 and 0.036 m, showing significant improvements compared to those (0.174 m for the seminar room and 0.154 m for the corridor) achieved by our previous algorithm in Hong et al. (2015), which failed to model some structural components because of the complexity of the test sites. The quality of the two 3D wireframe models was additionally evaluated by the root-mean-square error (RMSE) and the spherical accuracy standard (SAS). The RMSE was calculated as

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (a_i^t - b_i)^2} \quad (12)$$

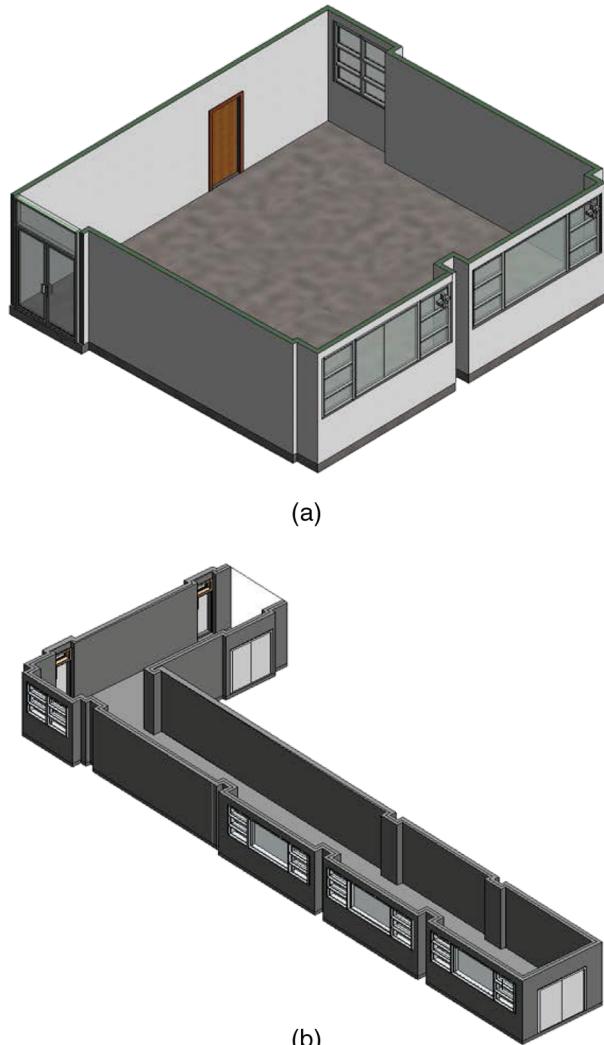
where a_i^t = point transformed to the coordinates of the total station. The RMSEs for each direction (x , y , and z) are also listed in Tables 5 and 6. The SAS, defined as the spherical radius of a 90% probability sphere (Greenwalt and Schultz 1968), was computed as

$$\text{SAS} = 2.5 \times 0.3333 \times (\text{RMSE}_x + \text{RMSE}_y + \text{RMSE}_z) \quad (13)$$

This represents a positional accuracy of the two generated 3D wireframe models at the 90% confidence level. The calculated SAS values were 0.068 m for the seminar room and 0.059 m for the corridor.

As-Built BIM Creation

Finally, the as-built BIMs of the seminar room and the corridor were created using the 3D wireframe model as a reference input. The 3D wireframe models were converted to a CAD-oriented DXF format and imported into the commercial BIM software *AutoCAD Revit Architecture 2014*. Since the present study focused on geometric modeling, we assumed the material properties for the walls, ceiling, floor, and other structures. The level of detail in the as-built BIM reflected only the modeling of the structural-related components (i.e., walls, windows, and doors, excluding furniture). In the BIM software, the 3D wireframe input helped the modeler to easily identify the object positions and their relations to the other objects. The wireframe model, moreover, with its simplified and regularized

**Fig. 15.** 3D as-built BIMs of (a) the seminar room; (b) the corridor

lines, was also very efficient with respect to the creation of object components. At the same time, system slowdown attributable to the huge data size of point clouds was not an issue, thus showing that the proposed approach is an effective means of indoor as-built modeling. Fig. 15 illustrates the generated as-built BIM products of the seminar room and the corridor. Although the present study assumed the materials for the structural components, as Wang et al. (2015) pointed out, not all the semantic information that BIM contains about a building is essential for engineering analysis tools. The building geometry information can be used for various purposes, for example, energy simulation, acoustic and fire propagation simulation, design decisions concerning building sustainability, progressive construction monitoring, and defect detection.

Conclusions and Discussion

In this paper, an automated methodology for creation of a 3D wireframe model of indoor surface structures with point-cloud data has been introduced. The major contributions of the present research are as follows: First, the proposed projection and the boundary-tracing processes automatically produce a complete 3D indoor wireframe model including detail-rich components such as windows and open doors in the presence of clutter and occlusion.

Second, in order to reduce the effects of the inherent noise of point locations and low density areas, the constrained least-squares method was proposed that provides regularized and geometrically consistent structural lines satisfying the parallel and orthogonal conditions, which holds valid for a majority of indoor environments. This is particularly important because BIM has been primarily used for the design of new buildings; commercial software is currently not suitable for modeling nonideal geometries that often occur in existing buildings because of deformation and damages over time (Dore and Murphy 2014). Third, for the purpose of a robust evaluation metric, we proposed the leave-one-out cross-validation based on the total station measurements, which provides a reliable and unbiased solution even when lacking a sufficient data set.

However, there are several issues requiring further investigation in future work. Although the proposed point-cloud processing and regularization techniques help to reduce the effects of clutter and noise, they provide only the straight-line models of walls, ceiling, floor, windows, and open doors, satisfying the parallel or orthogonal assumptions. Thus, surface anomalies or construction details are not reflected in the final product. In addition, the parallel and orthogonal assumptions could be invalid when faced with highly noisy measurements (Xiong et al. 2013) or irregularly shaped indoor environments. To manage this problem, we are investigating new techniques for fitting of nonlinear structures. The method proposed by Turner and Zakhor (2012) uses the RANSAC algorithm, which incorporates the circle model to fit the curved segment. Dimitrov and Golparvar-Fard (2014) proposed nonuniform B-spline (NURBS) curves to fit the presegmented point-cloud data. This could potentially extend our approach to model the indoor space including curved surfaces. Accordingly, the present regularization phase, which is limited to straight boundary and right-angled constraints, also should be modified to enable refinement of new environments including curved components.

Noise removal in the preprocessing step, which potentially will improve modeling accuracy, also demands consideration. Particularly in the wall modeling phase, a clutter passing through the ceiling and floor [for example, the white conduit in Fig. 6(a)] can lead to incorrect wall boundary tracing, which later results in modeling errors. In the window modeling phase, direct occlusions on the window frames caused by clutter or moving objects may lead to errors for window detections. Both problems could be prevented by careful noise removal. For this reason, automated recognition and elimination of such noise factors is highly demanded. In the long term, our research should explore direct scan-to-BIM creation; therefore, we are currently investigating methods for extending our approach to semantic and volumetric modeling. In addition, exporting the modeled product in a standard BIM file format is necessary to ensure that the model can be identified and is accessible for various applications. Within the AEC industry, several exchange schemas, such as industry foundation classes (IFC) and model view definitions (MVD), have been specifically designed to support interoperability (Belsky et al. 2015), and they are definitely worth further study in future works.

Acknowledgments

This research was supported by a grant (11 High-tech G11) from the Architecture & Urban Development Research Program funded by the Korean Ministry of Land, Infrastructure and Transport. Also, the authors would like to thank Prof. Burcin Becerik-Gerber and Prof. Vineet R. Kamat for insightful comments that improved the quality of this manuscript.

References

- Adán, A., and Huber, D. (2010). "Reconstruction of wall surfaces under occlusion and clutter in 3D indoor environments." *CMU-RI-TR-10-12*, Robotics Institute, Carnegie Mellon Univ., Pittsburgh.
- Admuthe, L. S., and Bendre, P. G. (2012). "Road sign detection using image processing and recognition." *Int. J. Eng. Sci. Res. Technol.*, 1(10), 592–597.
- Anil, E. B., Tang, P., Akinci, B., and Huber, D. (2013). "Deviation analysis method for the assessment of the quality of the as-is building information models generated from point cloud data." *Autom. Constr.*, 35, 507–516.
- Arras, K. O., Tomatis, N., Jensen, B. T., and Siegwart, R. (2001). "Multi-sensor on-the-fly localization: Precision and reliability for applications." *Rob. Auton. Syst.*, 34(2), 131–143.
- Autodesk Recap* [Computer software]. Autodesk, San Rafael, CA.
- Belsky, M., Sacks, R., and Brilakis, I. (2015). "Semantic enrichment for building information modeling." *Comput.-Aided Civ. Infrastruct. Eng.*, 1–14.
- Bhatla, A., Choe, S. Y., Fierro, O., and Leite, F. (2012). "Evaluation of accuracy of as-built 3D modeling from photos taken by handheld digital cameras." *Autom. Constr.*, 28, 116–127.
- Boulaassal, H., Landes, T., Grussenmeyer, P., and Tarsha-Kurdi, F. (2007). "Automatic segmentation of building facades using terrestrial laser data." *Int. Archiv. Photogramm. Remote Sens. Spatial Inform. Syst.*, 36(3), 65–70.
- Budroni, A., and Böhm, J. (2009). "Toward automatic reconstruction of interiors from laser data." *Proc., 3D Virtual Reconstruction and Visualization of Complex Architectures (3D-Arch)*, International Society for Photogrammetry and Remote Sensing (ISPRS), Netherlands.
- Budroni, A., and Böhm, J. (2010). "Automatic 3D modelling of indoor Manhattan-world scenes from laser data." *Proc., ISPRS Commission V Mid-Term Symp.: Close Range Image Measurement Techniques*, International Society for Photogrammetry and Remote Sensing (ISPRS), Netherlands, 115–120.
- Cawley, G. C., and Talbot, N. L. (2004). "Fast exact leave-one-out cross-validation of sparse least-squares support vector machines." *Neural Networks*, 17(10), 1467–1475.
- Costin, A., and Teizer, J. (2014). "Utilizing BIM for real-time visualization and indoor localization of resources." *Proc., Int. Conf. on Computing in Civil and Building Engineering (ICCCBE)*, ASCE, Reston, VA, 649–656.
- Dimitrov, A., and Golparvar-Fard, M. (2014). "Robust NURBS surface fitting from unorganized 3D point clouds for infrastructure as-built modeling." *Proc., Computing in Civil and Building Engineering*, ASCE, Reston, VA, 81–88.
- Dimitrov, A., and Golparvar-Fard, M. (2015). "Segmentation of building point cloud models including detailed architectural/structural features and MEP systems." *Autom. Constr.*, 51, 32–45.
- Dore, C., and Murphy, M. (2014). "Semi-automatic generation of as-built BIM façade geometry from laser and image data." *J. Inform. Technol. Constr.*, 19, 20–46.
- Douglas, D. H., and Peucker, T. K. (1973). "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." *Cartographica: Int. J. Geog. Inform. Geovisualization*, 10(2), 112–122.
- Fischler, M. A., and Bolles, R. C. (1981). "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography." *Commun. ACM*, 24(6), 381–395.
- Greenwalt, C., and Schultz, M. (1968). "Principles of error theory and cartographic applications." *ACIC Technical Rep. No. 96*, Aeronautical Chart and Information Center, U.S. Air Force, St. Louis.
- Hagedorn, B., Trapp, M., Glander, T., and Dollner, J. (2009). "Towards an indoor level-of-detail model for route visualization." *10th Int. Conf. on Mobile Data Management: Systems, Services and Middleware*, IEEE, 692–697.
- Han, S., Cho, H., Kim, S., Jung, J., and Heo, J. (2012a). "Automated and efficient method for extraction of tunnel cross sections using terrestrial laser scanned data." *J. Comput. Civ. Eng.*, 10.1061/(ASCE)CP.1943-5487.0000211, 274–281.

- Han, S. H., Heo, J., Sohn, H. G., and Yu, K. (2009). "Parallel processing method for airborne laser scanning data using a PC cluster and a virtual grid." *Sensors*, **9**(4), 2555–2573.
- Han, S. H., Kim, S. M., Jung, J. H., Kim, C. J., Yu, K. Y., and Heo, J. (2012b). "Development of a hashing-based data structure for the fast retrieval of 3D terrestrial laser scanned data." *Comput. Geosci.*, **39**, 1–10.
- Heo, J., et al. (2013). "Productive high-complexity 3D city modeling with point clouds collected from terrestrial LiDAR." *Comput. Environ. Urban Syst.*, **41**, 26–38.
- Hichri, N., Stefani, C., De Luca, L., and Veron, P. (2013). "Review of the 'as-built BIM' approaches." *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, **XL-5/W1**, 107–112.
- Hong, S., Jung, J., Kim, S., Cho, H., Lee, J., and Heo, J. (2015). "Semi-automated approach to indoor mapping for 3D as-built building information modeling." *Comput. Environ. Urban Syst.*, **51**, 34–46.
- Isikdag, U., Zlatanova, S., and Underwood, J. (2013). "A BIM-oriented model for supporting indoor navigation requirements." *Comput. Environ. Urban Syst.*, **41**, 112–123.
- Jung, J., et al. (2014). "Productive modeling for development of as-built BIM of existing indoor structures." *Autom. Constr.*, **42**, 68–77.
- Karsli, F., and Kahya, O. (2008). "Building extraction from laser scanning data." *Proc., Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS Archives)*, Beijing.
- Khan, J. F., Bhuiyan, S. M., and Adhami, R. R. (2011). "Image segmentation and shape analysis for road-sign detection." *Int. Transp. Syst., IEEE Trans. on*, **12**(1), 83–96.
- Klein, L., Li, N., and Becerik-Gerber, B. (2011). "Comparison of image-based and manual field survey methods for indoor as-built documentation assessment." *Computing in Civil Engineering*, ASCE, Reston, VA, 59–66.
- Klein, L., Li, N., and Becerik-Gerber, B. (2012). "Imaged-based verification of as-built documentation of operational buildings." *Autom. Constr.*, **21**, 161–171.
- Li, S., Isele, J., and Bretthauer, G. (2008). "Proposed methodology for generation of building information model with laserscanning." *Tsinghua Sci. Technol.*, **13**(S1), 138–144.
- Li, X., Claramunt, C., and Ray, C. (2010). "A grid graph-based model for the analysis of 2D indoor spaces." *Comput. Environ. Urban Syst.*, **34**(6), 532–540.
- Liu, X., Eybpoosh, M., and Akinci, B. (2012). "Developing as-built building information model using construction process history captured by a laser scanner and a camera." *Proc., Construction Research Congress*, ASCE, Reston, VA, 1232–1241.
- MATLAB version 8.0* [Computer software]. MathWorks, Natick, MA.
- Mura, C., Mattausch, O., Villanueva, A. J., Gobbetti, E., and Pajarola, R. (2014). "Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts." *Comput. Graphics*, **44**, 20–32.
- Nahangi, M., Haas, C. T., West, J., and Walbridge, S. (2015). "Automatic realignment of defective assemblies using an inverse kinematics analogy." *J. Comput. Civ. Eng.*, **10.1061/(ASCE)CP.1943-5487.0000477**, 04015008.
- Navarro, D., Benet, G., and Martínez, M. (2007). "Line based robot localization using a rotary sonar." *Proc., IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)*, IEEE, Patras, Greece, 896–899.
- Nguyen, V., Gächter, S., Martinelli, A., Tomatis, N., and Siegwart, R. (2007). "A comparison of line extraction algorithms using 2D range data for indoor mobile robotics." *Auton. Rob.*, **23**(2), 97–111.
- Okorn, B., Xiong, X., Akinci, B., and Huber, D. (2010). "Toward automated modeling of floor plans." *Proc., Symp. on 3D Data Processing, Visualization and Transmission*, Paris.
- Pătrăucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., and Haas, C. (2015). "State of research in automatic as-built modelling." *Adv. Eng. Inform.*, **29**(2), 162–171.
- Pu, S., and Vosselman, G. (2007). "Extracting windows from terrestrial laser scanning." *Int. Archiv. Photogramm. Remote Sens. Spatial Inform. Sci.*, **XXXVI**, 320–325.
- Randall, T. (2011). "Construction engineering requirements for integrating laser scanning technology and building information modeling." *J. Constr. Eng. Manage.*, **10.1061/(ASCE)CO.1943-7862.0000322**, 797–805.
- Reit, B. (1998). "The 7-parameter transformation to a horizontal geodetic datum." *Survey Rev.*, **34**(268), 400–404.
- Schulz, T., Ingensand, H., and Steiner, M. (2005). "Laser scanning and noise reduction applied to 3D road surface analysis." *Proc., 7th Optical 3-D Measurement Techniques Conf.*, Technische Universität Berlin, Germany.
- Sepasgozar, S. M. E., Lim, S., Shirowzhan, S., and Kim, Y. M. (2014). "Implementation of as-built information modelling using mobile and terrestrial lidar systems." *Proc., Int. Symp. on Automation and Robotics in Construction (ISARC)*, Citeseer, Sydney, Australia.
- Siadat, A., Kaske, A., Klausmann, S., Dufaut, M., and Husson, R. (1997). "An optimized segmentation method for a 2D laser-scanner applied to mobile robot navigation." *Proc., 3rd IFAC Symp. on Intelligent Components and Instruments for Control Applications*, Citeseer, Annecy, France, 153–158.
- Snow, K. (2009). "Adjustment computation notes." Ohio State Univ., OH, 92.
- Tang, P., Huber, D., Akinci, B., Lipman, R., and Lytle, A. (2010). "Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques." *Autom. Constr.*, **19**(7), 829–843.
- Turner, E., and Zakhor, A. (2012). "Watertight as-built architectural floor plans generated from laser range data." *2nd Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, IEEE, 316–323.
- Valero, E., Adan, A., and Cerrada, C. (2012). "Automatic method for building indoor boundary models from dense point clouds collected by laser scanners." *Sensors*, **12**(12), 16099–16115.
- Varma, S., and Simon, R. (2006). "Bias in error estimation when using cross-validation for model selection." *BMC Bioinform.*, **7**(1), 91.
- Volk, R., Stengel, J., and Schultmann, F. (2014). "Building information modeling (BIM) for existing buildings—Literature review and future needs." *Autom. Constr.*, **38**, 109–127.
- Wang, C., and Cho, Y. K. (2014). "Automatic as-is 3D building models creation from unorganized point clouds." *Proc., Construction Research Congress (CRC)*, ASCE, Atlanta, 917–924.
- Wang, C., Cho, Y. K., and Kim, C. (2015). "Automatic BIM component extraction from point clouds of existing buildings for sustainability applications." *Autom. Constr.*, **56**, 1–13.
- Wolf, P. R., and Ghilani, C. D. (1997). *Adjustment computations: Statistics and least squares in surveying and GIS*, Wiley, New York.
- Xiong, X., Adan, A., Akinci, B., and Huber, D. (2013). "Automatic creation of semantically rich 3D building models from laser scanner data." *Autom. Constr.*, **31**, 325–337.
- Yeung, J., Nahangi, M., Shahtaheri, Y., Haas, C., Walbridge, S., and West, J. (2014). "Comparison of methods used for detecting unknown structural elements in three-dimensional point clouds." *Construction Research Congress 2014@ sConstruction in a Global Network*, ASCE, Reston, VA, 945–954.
- Zhang, G., Vela, P., and Brilakis, I. (2013). "Detecting, fitting, and classifying surface primitives for infrastructure point cloud data." *Computing in Civil Engineering*, ASCE, Reston, VA, 589–596.
- Zhang, G., Vela, P. A., Karasev, P., and Brilakis, I. (2015). "A sparsity-inducing optimization-based algorithm for planar patches extraction from noisy point-cloud data." *Comput.-Aided Civ. Infrastruct. Eng.*, **30**(2), 85–102.
- Zhang, X., Rad, A. B., and Wong, Y.-K. (2008). "A robust regression model for simultaneous localization and mapping in autonomous mobile robot." *J. Intell. Rob. Syst.*, **53**(2), 183–202.