



Article

Indoor–Outdoor Point Cloud Alignment Using Semantic–Geometric Descriptor

Yusheng Yang , Guorun Fang, Zhonghua Miao and Yangmin Xie *

School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China

* Correspondence: xieym@shu.edu.cn; Tel.: +86-182-2152-2016

Abstract: Aligning indoor and outdoor point clouds is a challenging problem since the overlapping area is usually limited, thus resulting in a lack of correspondence features. The windows and doors can be observed from both sides and are usually utilized as shared features to make connections between indoor and outdoor models. However, the registration performance using the geometric features of windows and doors is limited due to the considerable number of extracted features and the mismatch of similar features. This paper proposed an indoor/outdoor alignment framework with a semantic feature matching method to solve the problem. After identifying the 3D window and door instances from the point clouds, a novel semantic–geometric descriptor (SGD) is proposed to describe the semantic information and the spatial distribution pattern of the instances. The best object match is identified with an improved Hungarian algorithm using indoor and outdoor SGDs. The matching method is effective even when the numbers of objects are not equal in the indoor and outdoor models, which is robust to measurement occlusions and feature outliers. The experimental results conducted in the collected dataset and the public dataset demonstrated that the proposed method could identify accurate object matches under complicated conditions, and the alignment accuracy reached the centimeter level.



Citation: Yang, Y.; Fang, G.; Miao, Z.; Xie, Y. Indoor–Outdoor Point Cloud Alignment Using Semantic–Geometric Descriptor. *Remote Sens.* **2022**, *14*, 5119. <https://doi.org/10.3390/rs14205119>

Academic Editor: Susana Lagüela López

Received: 21 September 2022

Accepted: 10 October 2022

Published: 13 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: semantic–geometric descriptor; window and door detection; improved Hungarian algorithm

1. Introduction

Three-dimensional (3D) building reconstruction has been studied for decades, and has promoted the technical development in autonomous navigation [1], augmented reality (AR) [2], virtual reality (VR) [3], etc. The integration of indoor and outdoor models is necessary for 3D building reconstruction as it exhibits the full view of the scene rather than only capturing the “surfaces” from one side [4]. A full 3D building model enables functions involving both the exterior and interior structures for the above-mentioned applications. For example, visitors can walk into a building in the VR world, rather than just wandering in the street [5]. Another example is ancient heritage building preservation, for which a complete digital model is necessary [6].

However, aligning the indoor and outdoor 3D models is challenging due to the lack of common visual correspondence [7]. Since the views from the two sides are often blocked by the walls, there are few overlapping areas, and thus insufficient corresponding features [8]. This problem is further aggravated when the captured data are sparse or incomplete. The shared objects observed from both sides, such as doors and windows, become the necessary links for the alignment problem. In earlier works, most researchers used the objects’ geometric features to build the connection, including points [9,10], lines [7,11] and planes [12–14]. However, some practical problems limited the performance of such geometric-based registration methods. On the one hand, a substantial number of geometric features can be detected, and these features are easily corrupted by measurement noises. As a result, finding a correct geometric correspondence with many outliers is difficult,

and the computational cost for the feature matching algorithms is considerable. On the other hand, the previous geometric-based methods did not consider the 3D property of objects. Consequently, the matched correspondences may actually describe different parts of an object, for example, the inside and outside edges of a window frame, which could cause registration errors.

Therefore, the semantic instance is a reasonable replacement of the geometric features in correspondence identification. Compared to geometric features, the number of semantic instances is limited, and the object recognition accuracy is much less sensitive to measurement noises [15]. The first demonstration of using semantic instances is the work in [16], where the windows were adopted as the alignment reference. They detected the window instances from RGB images and provided the alignment results by examining the shape similarity for all possible correspondence pairs. Due to the demand for colorful images, their method could not be directly applied to point cloud registration. More importantly, only the shape similarity between objects was considered, but not their distribution patterns. Without the latter constraints, the algorithm loses the global understanding of the objects' relative relationship and could fail when multiple objects shared identical or similar shapes, which is a common phenomenon in the structures of buildings.

In this paper, we proposed an indoor/outdoor point cloud alignment algorithm with a semantic feature matching method, as shown in Figure 1. The 3D window and door instances are recognized, segmented and localized from the indoor and outdoor point clouds, with a similar approach to our previous work [17]. We designed a semantic-geometric descriptor (SGD) to include both the objects' semantic information and spatial distribution pattern. The SGD is utilized to find the best instance matches between the indoor and outdoor semantic instances, even with unequal numbers of candidates in the two sets. Then, the indoor and outdoor models can be registered and connected. The major contributions of this work include:

1. A novel framework to use semantic objects in indoor–outdoor point cloud alignment tasks is proposed. It is the first work to include the objects' distribution pattern in model matching, which inherently prevents the ambiguity caused by objects' shape similarity.
2. A unique feature descriptor called the SGD is proposed to include both the semantic information and relative spatial relationship of 3D objects in a scene. The Hungarian algorithm is improved to detect the same object distribution patterns automatically and output optimal matches.
3. The algorithms are tested on both an experimental dataset and a public dataset. The results show that the SGD-based indoor–outdoor alignment method can provide robust matching results and achieve matching accuracy at the centimeter level.

The rest of the paper is organized as follows. In Section 2, related works about indoor and outdoor registration are introduced. The window and door detection method and the procedure of SGD generation are proposed in Section 3, followed by the SGD matching algorithm in Section 4. The experiential results in Section 5 demonstrate the effectiveness of the proposed method and the paper is concluded in Section 6.

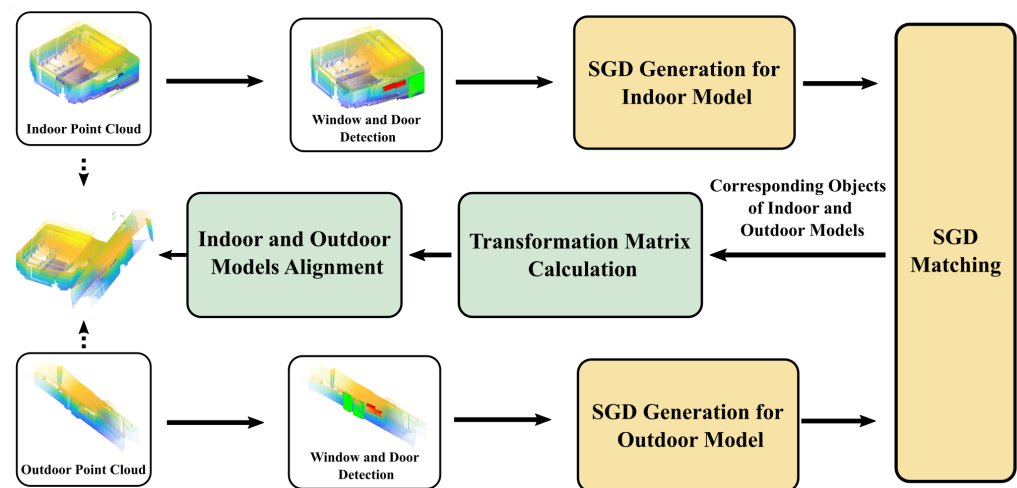


Figure 1. The overview of the proposed method. By matching the SGDs of indoor and outdoor point clouds, the corresponding objects can be recognized and the transformation between indoor and outdoor point clouds can be calculated accordingly.

2. Related Works

Registering the 3D point clouds of a building's interior and exterior models requires finding common features from both sides [18]. Previous studies have developed two types of features for this purpose: geometric-type and semantic-type features. They are all related to the doors and windows on the wall, the most commonly seen objects shared by views from the two sides.

Geometric-type features, such as points, lines and planes, can be extracted abundantly in the point clouds of a building. Muhammad Imanullah et al. [19] proposed a 2D SIFT (scale invariant feature transform) keypoint-based registration approach by extracting the SIFT features from the RGBD images. Rami Assi et al. [9] directly detected the 3D SIFT features from the point clouds. By point cloud voxelization, Biao Xiong et al. [20] caught the keypoints from the voxel grid and registered point clouds with the four-point congruent set technology. Compared to the point features above, line features, which can be extracted by sliding a sectioning plane through the point cloud [21], are more robust since they capture quadrilateral structures of the windows or doors [22]. Chenglu Wen et al. [22] used the iterative closest point (ICP) algorithm on the extracted 3D line segments. A patch-based classifier was adopted on the point cloud to identify the plane first, and the line structures were extracted from the plane. Another line-based method by Tobias Koch et al. [7] also assumed that lines located on the window plane. Rather than using ICP, they found the transformation between indoor and outdoor models by minimizing the perpendicular length of line segments. For point and line features, some researchers took plane polygons from the point clouds to restrict their positions and limit the number of potential matches, thus reducing the computational complexity. For example, Rahima Djahel et al. [12] proposed an algorithm matching the planar polygons by clustering polygons based on their normal direction and the offset in the normal direction. A plane-based descriptor that characterized the interrelation among nonparallel plane/lines was proposed by Songlin Chen et al. [23]. The difficulties of using geometric-type features in indoor–outdoor alignment mainly lie in two aspects. However, there are potentially two limits. First, it is hard to exclude geometric features that do not belong to windows or doors. As a result, there are many outliers in the feature detection. Second, plenty of geometric-type features can usually be identified in indoor and outdoor models. The large number of features decreases the matching efficiency, especially for buildings with complex structures.

With the development of deep learning technology, object recognition and segmentation technology have been fully developed. Therefore, semantic objects, such as windows and doors, can be used as prior information to find entity correspondence between the indoor and outdoor models [24,25]. In earlier works, Langyue Wang and Guoho Sohn [26,27]

emphasized the benefits of having both semantic and geometric information in the full building model. However, they could only extract the semantic objects from 2D floor plans and use lines of the 3D model section planes for feature detection. The method could not provide an accurate 3D localization of the objects, and the alignment could be corrupted when the point cloud was particularly incomplete. A more recent work [16] used RGBD sensors to push forward the research into a 3D semantic object alignment direction. The authors recognized windows in the RGB images and extracted their corresponding 3D locations in the indoor and outdoor point clouds. A brute-force search was applied for all possible matching choices between the two sets of window objects by examining the shape similarity, the building outline fitness and the matched number. The one with a comparatively low cost in these three aspects was output as the correspondence matching result. The method was sensitive to symmetry conditions without considering the relative positions among the objects, and manual assistance was required to select the best candidate.

From the literature discussed above, it is clear that both geometric and semantic information is valuable for indoor–outdoor alignment tasks. On the one hand, semantic instances can provide less noisy and less complex feature expressions for the windows and doors. On the other hand, not only the shape similarities of the objects but also their relative topology and position relationships matter when finding the correct correspondence between the indoor and outdoor data. This article thereby aims to invent a unique descriptor and the corresponding matching method to solve the problem, which considers both semantic and geometric affinities between two 3D object sets.

3. SGD Construction

SGD is the essential element in this paper to describe the features of semantic instances. In this section, we introduce the method of object detection and localization as a prerequisite. With a carefully designed procedure, as shown in Figure 2, the doors and windows can be localized and bounded by 3D boxes, which provide both the semantic information and geometric location for the detected objects. Since the objects' distribution pattern is identical in indoor and outdoor point clouds for the same scene, it can be utilized as a matching reference with a properly defined descriptor, i.e., the SGD proposed in this paper. The important abbreviations and symbols used in the paper are listed in Table 1 for clarity.

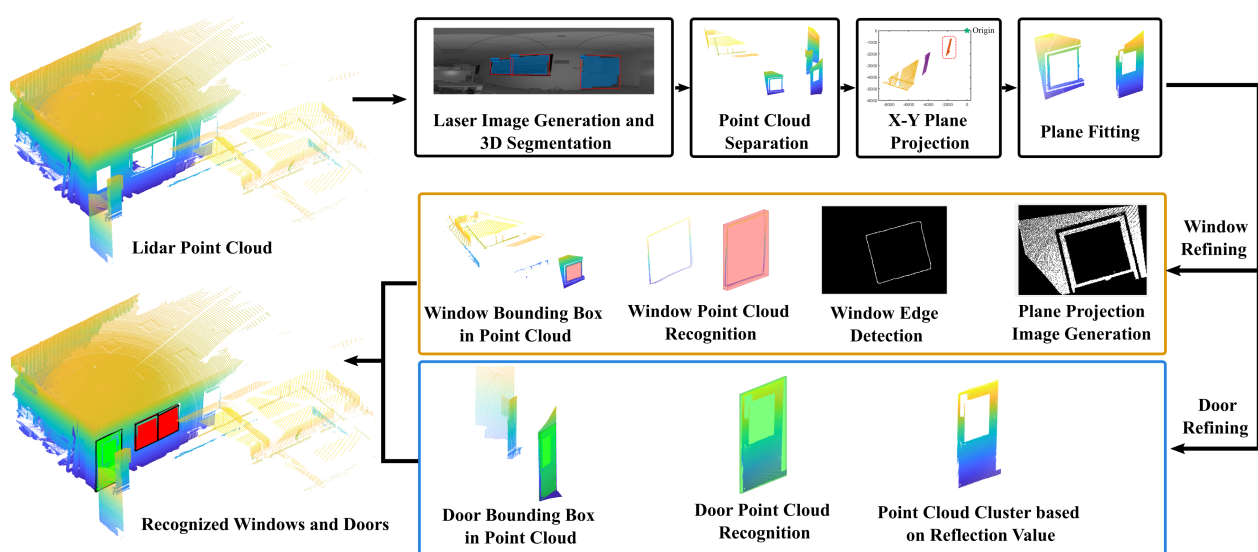


Figure 2. The procedure of windows and doors' bounding-box detection from the point clouds.

Table 1. List of important abbreviations and symbols used in the paper.

Abbreviation	Definition	Abbreviation	Definition
$\{\alpha_i^j, \beta_i^j, \theta_i^j\}$	The included angle between origin vector and the x–y–z axes in O_i coordinates	G^i	The relationship of object O_i with respect to other objects in the point cloud
$\{\alpha_e, \beta_e, \theta_e\}$	The absolute angle errors between angle elements	HA	Hungarian algorithm
$\{rotx_i^j, roty_i^j, rotz_i^j\}$	The Euler angles that represent the rotation between O_i coordinates and O_j coordinates	ICP	Iterative closest point
$\{rotx_e, roty_e, rotz_e\}$	The absolute rotation errors between two local coordinates	IoU	Intersection over union
$C_{SGD_{in}}^{out}$	The adjacency matrix between indoor and outdoor models	N	The number of recognized objects
$C_{SGDU_k}^l$	The distribution matrix between $SGDU^k$ and $SGDU^l$	O_i	The i th recognized object
C_{sp}^q D	The matching element in C_{SGDU} The matrix definition of SGD	S^i SGD	The semantic category of object O_i Semantic–geometric descriptor
d_i^j	The Euler distance between the origins of local coordinates of O_i and O_j	SGDU	Semantic–geometric descriptor unit
d_e	The absolute distance error between two distance elements	SVD	Singular value decomposition

3.1. Window and Door Detection in Point Clouds

Three-dimensional object detection from LiDAR point clouds has been widely studied [28,29]. However, window and door detection is not a trivial task for two reasons. First, they usually present as hollow structures instead of solid bodies in point clouds. The nature of sparse information makes it hard to be directly identified. Second, as they are used for map alignments, the final merging error is largely affected by the localization accuracy of the semantic objects. General 3D object detection and localization techniques cannot provide 3D bounding boxes with sufficiently high precision. To solve the first problem, we used the projected 2D image of the point cloud for 3D segmentation with deep learning neural networks [30]. The details are found in our previous work in [17]. There are inevitable outliers on the 2D segmentation edges, which induce extra points in the background. We used an X–Y projection and point-clustering method to group the points, which is similar to the method in [28]. The points belonging to the target objects could be extracted as the group in the front. Since doors and windows are flat objects, a plane fitting was further used to remove possible noises induced by the edge effect [31]. Outliers were removed if their distance to the fitted plane was beyond the threshold. After the procedure above, groups of points with the rough contour of the doors and windows could be extracted.

To achieve a higher bounding-box accuracy, we designed a refining step. Due to their different geometric properties, the refining methods for the windows and doors were separately designed. For the windows, we first projected the point cloud on the fitted plane and generated a 2D image. The image was further processed by dilating and eroding it to eliminate small holes. The window edge was taken as the innermost closed edge. The 3D points with a projection inside the detected edge contour were taken as the window points, and the bounding box could be generated accordingly. As for the doors, we utilized the fact that a door usually has a different texture from the surrounding wall. Therefore, their reflection measurements are distinct. We clustered the points by their reflection value, and the group with the largest number of points was taken as the

door. The surrounding points belonging to the wall were removed. The final results of the door and window localization are shown in Figure 2, as an example. With the refinement procedure, the localization accuracy is evidently improved.

It is worth mentioning that the reflectivity measurement may not be available if the data resource is not LiDAR. In this case, some steps in the procedure above may fail. The readers need to find an alternative way to obtain an accurate 3D box for the objects, for example, using the depth image to replace the reflectivity image. Alternatively, the RGB channels for an RGB-D point cloud can provide a similar function as the reflectivity measurement in the LiDAR data. No matter how the 3D localization of the windows and doors is obtained, the following SGD generation and the corresponding matching algorithms can be applied.

3.2. SGD Design

Given a point cloud F , the objects' semantic labels and spatial positions are known after the detection procedure above (Section 3.1). The local coordinates of each object are built based on their bounding boxes. Figure 3 describes the principle of the local coordinates' definition. The origin locates at the center of the bounding box; its x-axis points to the outside of the building, the z-axis points upward and the y-axis is built by the right-hand rule.

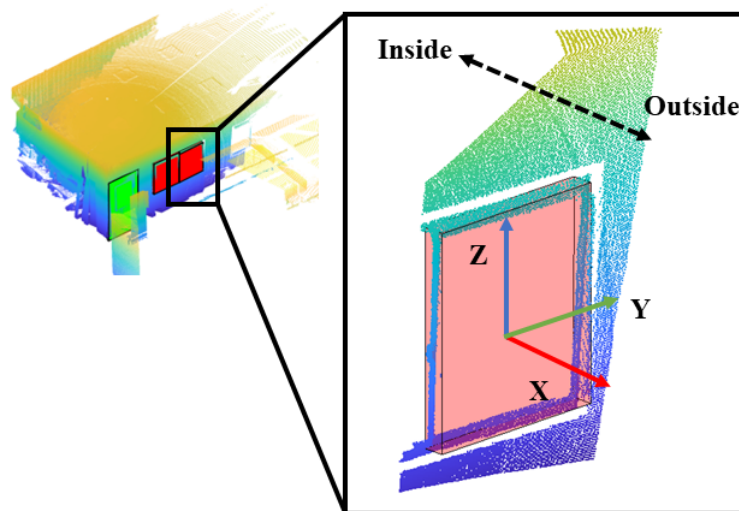


Figure 3. The principle of generating local coordinates on the bounding box.

The SGD of F describes the number and categories of the objects, as well as their relative positions and poses with respect to each other. The SGD is defined as a matrix $D \in R^{N \times (8N-7)}$, as illustrated in Figure 4, where N is the number of recognized objects in F . Each row element $\{SGDU^i = (S^i, G^i) \in R^{8N-7} | i = 1, \dots, N\}$ in D is the semantic-geometric descriptor unit (SGDU) of the i th object O_i . The first element S^i in $SGDU^i$ describes the semantic category of O_i . The other element $\{G^i = ({}^1G^i, \dots, {}^jG^i) | j = 1, \dots, n, \dots, N; j \neq i\}$ in $SGDU^i$ depicts the relationship of O_i with respect to the rest of the objects. Each component ${}^jG^i$ in G^i contains eight elements, denoted as ${}^jG^i = \{(S^j, d_i^j, \alpha_i^j, \beta_i^j, \theta_i^j, \text{rot}x_i^j, \text{rot}y_i^j, \text{rot}z_i^j) | j \neq i\}$. Specifically, S^j means the semantic label of O_j , d_i^j represents the Euler distance between the origins of the local coordinates of O_j and O_i . Taking a vector from the origin of O_i to the origin of O_j , the included angles between this vector and the x-y-z axes in the O_i coordinates are calculated as $\{\alpha_i^j, \beta_i^j, \theta_i^j\}$, respectively. $\{\text{rot}x_i^j, \text{rot}y_i^j, \text{rot}z_i^j\}$ are the Euler angles that represent the relative rotation between the O_i and O_j coordinates. Figure 5 illustrates the example of the components in ${}^2G^1$.

O_1	S^1	■	${}^2G^1$	...	${}^{n-1}G^1$	${}^nG^1$	${}^{n+1}G^1$	...	${}^{N-1}G^1$	${}^NG^1$
O_2	S^2	${}^1G^2$	■	...	${}^{n-1}G^2$	${}^nG^2$	${}^{n+1}G^2$	...	${}^{N-1}G^2$	${}^NG^2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
O_n	S^n	${}^1G^n$	${}^2G^n$	...	${}^{n-1}G^n$	■	${}^{n+1}G^n$	...	${}^{N-1}G^n$	${}^NG^n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
O_{N-1}	S^{N-1}	${}^1G^{N-1}$	${}^2G^{N-1}$	...	${}^{n-1}G^{N-1}$	${}^nG^{N-1}$	${}^{n+1}G^{N-1}$	...	■	${}^NG^{N-1}$
O_N	S^N	${}^1G^N$	${}^2G^N$	...	${}^{n-1}G^N$	${}^nG^N$	${}^{n+1}G^N$	...	${}^{N-1}G^N$	■

■ Empty Space

Figure 4. The SGD of a frame F : each row element in the SGD is the SGDU of object O_i . The first element in an SGDU describes the semantic category of O_i , and the other elements in SGDU depict the spatial relationship of O_i with respect to the other objects.

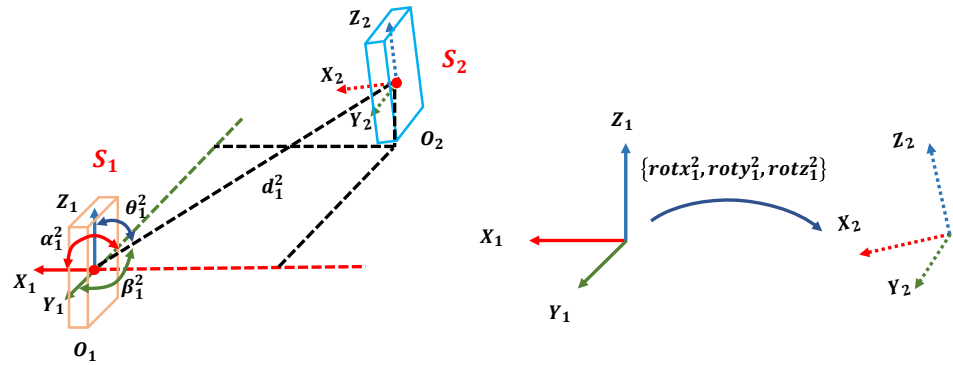


Figure 5. The description of each element in ${}^2G^1 = \{(S^2, d_1^2, \alpha_1^2, \beta_1^2, \theta_1^2, \text{rotx}_1^2, \text{roty}_1^2, \text{rotz}_1^2)\}$. S^2 is the semantic labels of object O_2 , d_1^2 is the Euler distance between O_1 and O_2 . $\{\alpha_1^2, \beta_1^2, \theta_1^2\}$ are the included angles between the vector O_1O_2 and the x-y-z axes in O_1 . $\{\text{rotx}_1^2, \text{roty}_1^2, \text{rotz}_1^2\}$ are the Euler angles that transform local coordinate 1 to local coordinate 2.

4. Semantic Object Matching with SGDs

The uniqueness of the SGD, compared to the commonly seen 2D and 3D descriptors, such as SIFT [32], SURF [33] and ORB [34], is that the values of the elements in an SGDU are dependent on the geometric relationship between the current object and the others. Therefore, the SGDU changes with the sequence of identified objects. As a result, we need to define the distance between two SGDUs in a creative way. The matching between the SGDs of the indoor and outdoor frames can be solved accordingly.

4.1. SGDU Distance Definition

Each SGDU represents the semantic label of one object and its spatial relationship with the other identified objects. The distance between two SGDUs describes whether they belong to the same semantic category and whether they are placed in similar relative positions with respect to the rest of the objects in the scene. Given $\{SGDU^k = (S^k, G^k)\}$ in frame 1 and $\{SGDU^l = (S^l, G^l)\}$ in frame 2, their distance $C_{SGDU_k^l}$ can be calculated in two steps. First, the semantic labels S^k and S^l are compared. If they have different semantic labels, the distance $C_{SGDU_k^l}$ is assigned as the infinity. Otherwise, we need to design an SGDU distance involving the similarity evaluation between $SGDU^k$ and $SGDU^l$ regarding their distribution pattern with respect to other objects. We define a distribution matrix $C_{SGDU_k^l}$.

$$C_{SGDU_k^l} = \begin{bmatrix} C_{g1}^1 & C_{g1}^2 & \cdots & C_{g1}^q & \cdots & C_{g1}^{N_l} \\ C_{g2}^1 & C_{g2}^2 & \cdots & C_{g2}^q & \cdots & C_{g2}^{N_l} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ C_{gp}^1 & C_{gp}^2 & \cdots & C_{gp}^q & \cdots & C_{gp}^{N_l} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ C_{gN_k}^1 & C_{gN_k}^2 & \cdots & C_{gN_k}^q & \cdots & C_{gN_k}^{N_l} \end{bmatrix} \quad (1)$$

where N_k and N_l denote the dimensions of G^k and G^l separately. The element C_{gp}^q in $C_{SGDU_k^l}$ represents the matching value of the relationship between the k th object in frame 1 with the p th object in $SGDU^k$ and the relationship between the l th object in frame 2 with the q th object in $SGDU^l$. The matching cost C_{gp}^q is set to be infinity if S^p in frame 1 does not equal S^q in frame 2. Otherwise, the geometric errors, $d_e = |d_k^p - d_l^q|$, $\alpha_e = |\alpha_k^p - \alpha_l^q|$, $\beta_e = |\beta_k^p - \beta_l^q|$, $\theta_e = |\theta_k^p - \theta_l^q|$, $rotx_e = |rotx_k^p - rotx_l^q|$, $roty_e = |roty_k^p - roty_l^q|$ and $rotz_e = |rotz_k^p - rotz_l^q|$, are examined. Each error is assigned a threshold as d_t , α_t , β_t , θ_t , $rotx_t$, $roty_t$ and $rotz_t$, respectively. If any error is larger than its threshold, C_{gp}^q is set to infinity. Otherwise, it can be calculated by the following equation:

$$C_{gp}^q = \lambda_1 * d_e + \lambda_2 * \alpha_e + \lambda_3 * \beta_e + \lambda_4 * \theta_e + \lambda_5 * rotx_e + \lambda_6 * roty_e + \lambda_7 * rotz_e \quad (2)$$

where $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ and $\lambda_7\}$ are the weight factors. There is no guarantee that the object sequences in $SGDU^k$ and $SGDU^l$ are identical, and the object numbers can be different. Therefore, we need to find the best sequence correspondence among all the possibilities when calculating the distance between $SGDU^k$ in frame 1 and $SGDU^l$ in frame 2. We delete all columns or rows with all elements being invalid, and the matching matrix is updated to $C_{SGDU_k^l}'$. This is to further decrease the matching candidates and save the computational cost for the following steps.

The Hungarian algorithm (HA) [35], which can solve the maximum weighted bipartite matching problem, was adopted and improved for this purpose. Given a bipartite graph that is made of two sets U and V with the same dimension and the corresponding square adjacency matrix E , the HA can find the maximum-weight matching and output the best correspondence assignment between the elements in U and V . This conventional HA requires the dimensions of U and V to be identical, which is not satisfied in the case of the indoor–outdoor object-matching problem. Therefore, we combined the HA with a brute-force searching algorithm to solve the dimension inconsistency problem. Denoting the dimensions of U and V as d_U and d_V , we regarded the smaller number of d_U and d_V as the potential matching number η . Then, we applied the HA on all square submatrices of E with a dimension of η . The best matching result should be the one with the minimum matching weight. The details of the process can be found in Algorithm 1 and its worse-case computational complexity is $O(\eta^4)$. Since the object numbers in both frames are limited, the computation cost is acceptable.

Applying the improved HA on $C_{SGDU_k^l}'$, we can find the matching cost and matched object numbers between $SGDU^k$ and $SGDU^l$, as presented in Algorithm 2. The distance between two SGDUs is their matching cost.

4.2. SGD Matching

The SGD of a frame represents the categories of the recognized objects and their spatial positions. Therefore, the SGDs of the indoor and outdoor frames contain the semantic–geometric information of the recognized windows and doors. Given the SGD $D_{in} \in R^{m_{in}}$ of the indoor frame and the SGD $D_{out} \in R^{m_{out}}$ of the outdoor frame, the SGD matching consists in finding the corresponding SGDUs in the two frames using their distances.

Algorithm 1: The improved Hungarian algorithm

Data: Set U , set V , weight matrix E
Result: Matching cost $Cost$
 $(Row, Col) =$ the number of rows and columns of E ;
 $\eta = \min(Row, Col)$;
if $Row > \eta$ **then**
 E' is the set that contains all possibilities that choose η rows in E .
else
 E' is the set that contains all possibilities that choose η columns in E .
end
 $costList = []$;
for $i = 1 : \text{length}(E')$ **do**
 Calculate the matching cost c of $E'(i)$ with conventional Hungarian algorithm;
 $costList(i) = c$;
end
 $Cost = \min(costList)$;
return $Cost$;

Algorithm 2: Calculating the match cost between two SGDUs

Data: $SGDU^k, SGDU^l$
Result: Match cost value C , number of common adjacent objects Num
 $SGDU^k = (S^k, G^k)$;
 $SGDU^l = (S^l, G^l)$;
if $S^k \neq S^l$ **then**
 $C = INF$;
 return $(C, 0)$;
else
 Calculate matching cost matrix $C_{SGDU_k^l}$ according to Equation 1;
 Update $C_{SGDU_k^l}$ to $C'_{SGDU_k^l}$ by removing columns and rows with all elements being invalid;
 if $C'_{SGDU_k^l}$ is empty **then**
 $C = INF$;
 return $(C, 0)$;
 else
 $(Row, Col) =$ the number of rows and columns of $(C'_{SGDU_k^l})$;
 $Num = \min(Row, Col)$;
 $C =$ the matching cost calculated based on Algorithm 1, by taking $C'_{SGDU_k^l}$ as the weight matrix E ;
 return (C, Num) ;
 end
end

Inherently, it is again an unbalanced assignment problem. Therefore, we can repeat the procedure in Section 4.1 for an optimal solution. The adjacency matrix is defined as

$$C_{SGD_{in}^{out}} = \begin{bmatrix} (C_{SGDU}^{1-1}, Num_{1-1}) & (C_{SGDU}^{1-2}, Num_{1-2}) & \cdots & (C_{SGDU}^{1-m_{out}}, Num_{1-m_{out}}) \\ (C_{SGDU}^{2-1}, Num_{2-1}) & (C_{SGDU}^{2-2}, Num_{2-2}) & \cdots & (C_{SGDU}^{2-m_{out}}, Num_{2-m_{out}}) \\ \cdots & \cdots & \ddots & \cdots \\ (C_{SGDU}^{m_{in}-1}, Num_{m_{in}-1}) & (C_{SGDU}^{m_{in}-2}, Num_{m_{in}-2}) & \cdots & (C_{SGDU}^{m_{in}-m_{out}}, Num_{m_{in}-m_{out}}) \end{bmatrix} \quad (3)$$

The element in $C_{SGD_{in}}^{out}$ is calculated according to Algorithm 2, and there are two parts for each element. The first part is the SGDU match cost of the two objects, and the second part is the number of their common neighbor objects. If the distance is infinity, the corresponding element is invalid. We delete all columns or rows with all elements being invalid to save further calculation costs, and the matching matrix is updated to $C_{SGD_{in}}^{out}$. At last, $C_{SGD_{in}}^{out}$ is provided as inputs to the same HA as in Algorithm 1, and the best-matching pairs of the objects in the indoor and outdoor frames can be found.

The transformation between the two frames is finally calculated by the SVD method on the corner points of the matched window/door bounding boxes. This is a commonly used method for transformation calculation between two matched sets of points and thus is not elaborated here. The procedure of SGD matching and transformation calculation between indoor and outdoor models is presented in Algorithm 3 in detail.

Algorithm 3: SGD matching and transformation calculation between indoor and outdoor models

Data: $D_{in} \in R^{m_{in}}, D_{out} \in R^{m_{out}}$

Result: The transformation $\mathcal{T}$ from outdoor model to indoor model

for $1 \leq i \leq m_k$ **do**

for $1 \leq j \leq m_l$ **do**

 Calculate the matching cost $C_{SGDU}^{i,j}$ and matching num $Num_{i,j}$ between $SGDU^i$ of D_{in} and $SGDU^j$ of D_{out} according to algorithm 2;

$C_{SGD_{in}}^{out}(i,j) = (C_{SGDU}^{i,j}, Num_{i,j})$;

end

end

(m'_k, m'_l) = the number of rows and columns for $C_{SGD_{in}}^{out}$;

for $1 \leq i \leq m'_k$ **do**

 SGD_row = the i th row in $C_{SGD_{in}}^{out}$;

if All elements in SGD_row are invalid **then**

 Removing i th row of $C_{SGD_{in}}^{out}$;

else

 continue;

end

end

for $1 \leq j \leq m'_l$ **do**

 SGD_column = the j th column in $C_{SGD_{in}}^{out}$;

if All elements in SGD_column are invalid **then**

 Removing j th column of $C_{SGD_{in}}^{out}$;

else

 continue;

end

end

Update $C_{SGD_{in}}^{out}$ to $C_{SGD_{in}}^{out}$;

C_{SGD}^v = the matching cost calculated based on Algorithm 1, by taking $C_{SGD_{in}}^{out}$ as the weight matrix E ;

The corresponding matching relationship for C_{SGD}^v represents the final matching result;

Pts_{in} = the bounding box's corner points of corresponding matching objects in indoor model;

Pts_{out} = the bounding box's corner points of corresponding matching objects in outdoor model;

$\mathcal{T} = SVD(Pts_{out}, Pts_{in})$;

Return $\mathcal{T}$

5. Experimental Results and Discussion

5.1. Experimental Dataset Description

In this study, a custom-made laser scanning system was adopted to record the indoor and outdoor point cloud data, as shown in Figure 6. The primary sensor of the scanning system was a laser scanner (Hokuyo UST-30LX), which was installed on a rotating platform and could measure the object's distance and reflection value. The scanning system covered a wide field of view ($270^\circ \times 180^\circ$) with a resolution of $0.25^\circ \times 0.1^\circ$. The system could be controlled remotely, which avoided the outliers caused by the operator. With the above configuration, the scanning system could capture the panoramic point cloud data of the environment. Additionally, the projected 2D image for each scanning was generated for the window/door detection, and one example is illustrated in Figure 7.

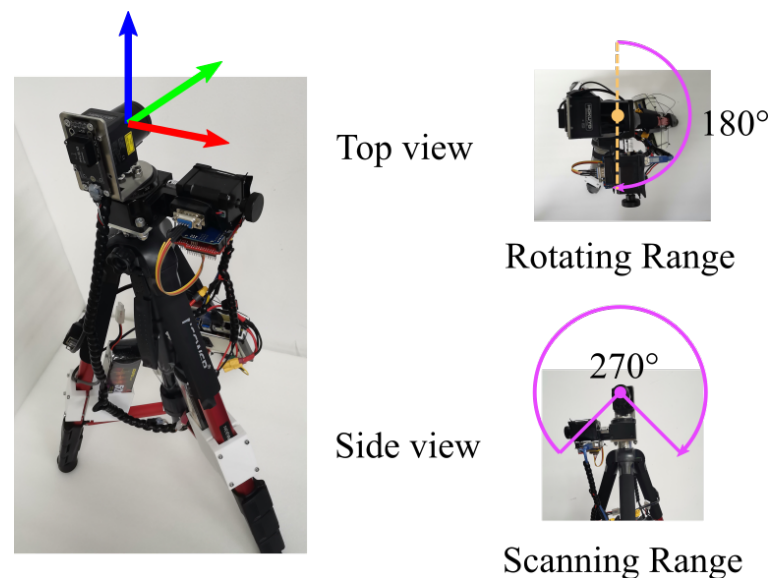


Figure 6. The custom-made laser scanning system. The primary sensor is the Hokuyo UST-30LX laser scanner, which covers a wide field of view $270^\circ \times 180^\circ$.

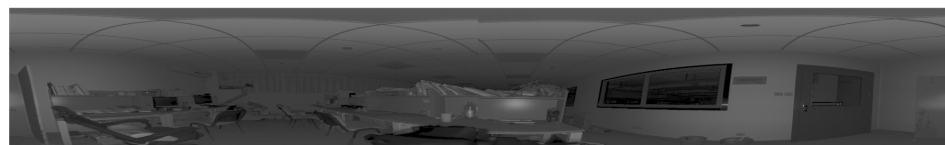


Figure 7. Example of a generated 2D image for one laser scan.

We collected data in two different scenarios, as shown in Figure 8. Scenario 1 included an L-shape corridor as the outdoor model and two separate rooms as the indoor model. Scenario 2 included a U-shape corridor with four separate rooms. Scene 3 and scene 4 contained two indoor frames for one large room. The outdoor frames corresponding to the indoor frames are shown as magenta points. Several outdoor point clouds were registered as a unified outdoor model for each scenario using an improved ICP method [36]. Eventually, there were two and five indoor–outdoor frame alignment tasks for scenario 1 and scenario 2, respectively. The collected dataset has been published.

5.2. Window and Door Detection Results

The indoor and outdoor frames in Figure 8 were projected as LiDAR images. After applying the 2D segmentation method from Section 3.1, the windows and doors were identified in the images, as shown in Figure 9. Though different in sizes and positions, most windows and doors were successfully separated from the background. Occasionally, some windows/doors may fail to be recognized due to occlusions in the indoor view. This only

decreased the usable number of semantic objects but did not lead to assignment failures, as shown in the following content.

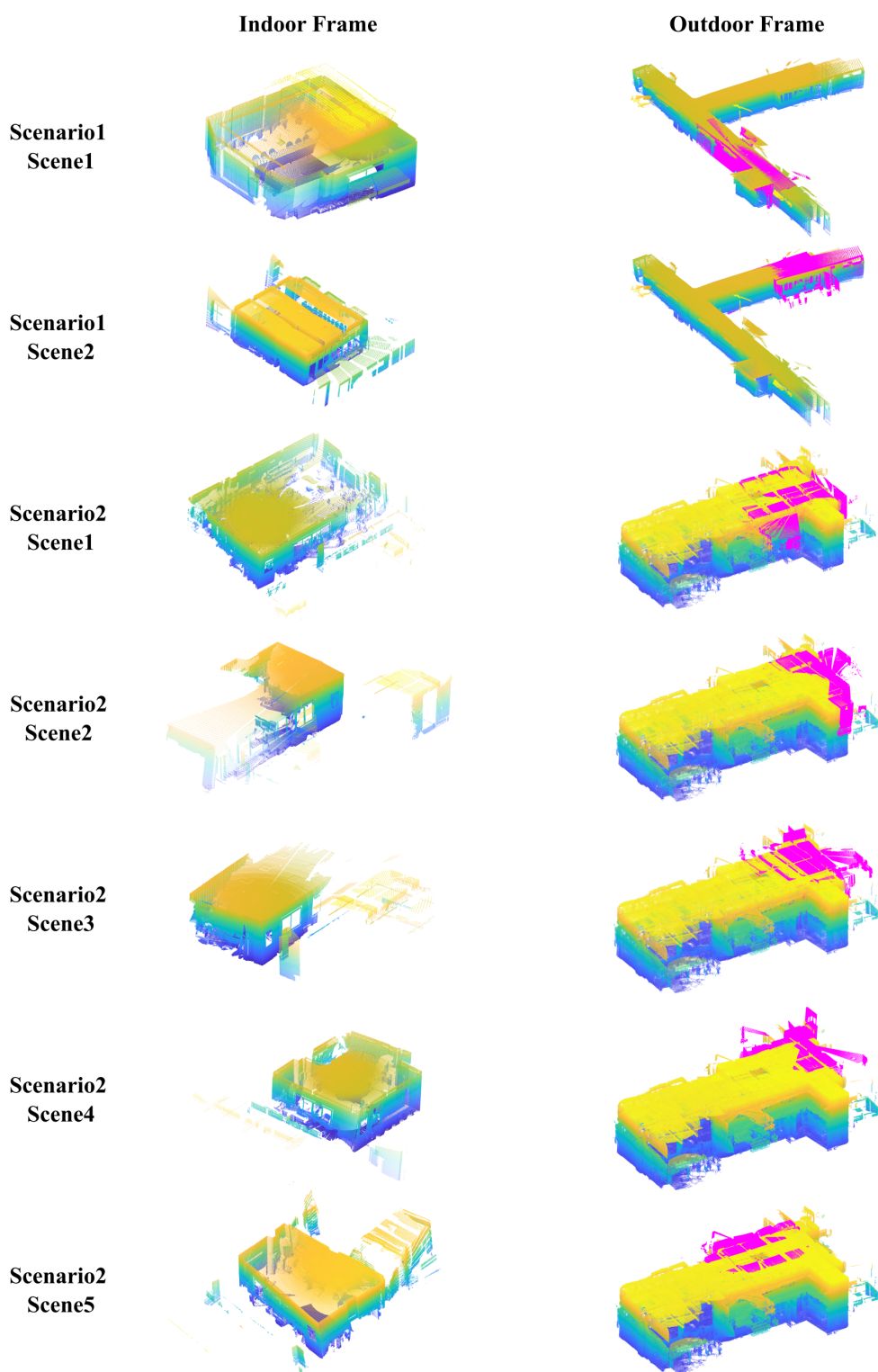


Figure 8. The collected dataset in two different scenarios. The point clouds of the indoor frames are presented in the left column, and the corresponding outdoor frames, colored in magenta, and their locations in the outdoor model are presented in the right column.

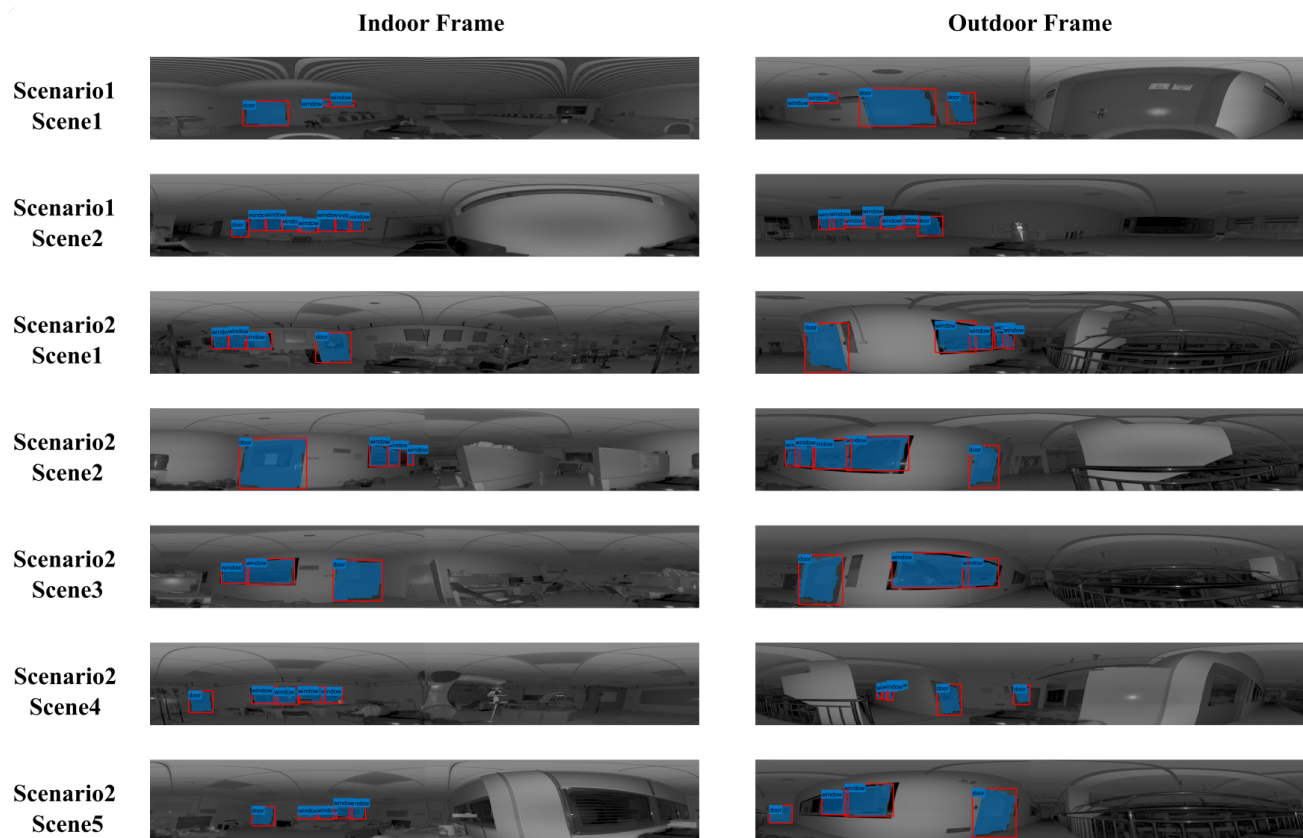


Figure 9. The projected LiDAR images for two scenarios and the windows and doors in the image are recognized using the 2D segmentation method.

Figure 10 illustrates the 3D bounding box results of the recognized windows and doors. We manually specified the bounding boxes as the ground truth. The accuracy of the window and door detection was evaluated separately by calculating the mean absolute error (MAE) of the corresponding bounding box corners, and the relative error was calculated by dividing the error value by the diagonal lengths of the corresponding windows and doors. Additionally, the intersection over union (IoU) of windows and doors for each scene was calculated, respectively. Two-dimensional IoUs measured the accuracy of window/door segmentation in the LiDAR images, and 3D IoUs measured the localization accuracy of the 3D bounding boxes.

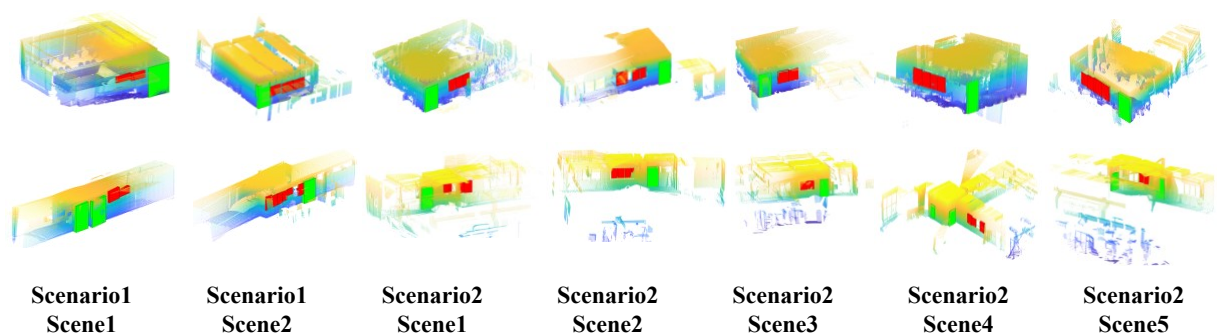


Figure 10. The extracted 3D bounding boxes of the windows and doors for each scene. The green boxes represent the door objects and the red boxes represent the window objects.

All evaluation results are shown in Table 2. Overall, the average MAEs of the windows detection were smaller than the MAEs of the doors detection, which were 0.0810 ± 0.0242 m and

0.1147 ± 0.0192 m, respectively. The relative errors were $5.6657 \pm 1.5406\%$ and $7.9993 \pm 1.1879\%$, respectively. The scene size was as large as twenty times that of the objects' size, and the detection error was less than 1% of the scene's dimension.

Table 2. The accuracy of the windows and doors detection.

	Scenario 1 Indoor	Scenario 1 Outdoor	Scenario 2 Indoor	Scenario 2 Outdoor	Scenario 3 Indoor	Scenario 3 Outdoor	Scenario 4 Indoor	Scenario 4 Outdoor	Scenario 5 Indoor	Scenario 5 Outdoor	Mean $\pm$ std				
MAE of windows (m)	0.0930	0.1454	0.0877	0.0711	0.0527	0.0921	0.0593	0.0501	0.0689	0.0790	0.0759	0.0940	0.0985	0.0666	0.0810 $\pm$ 0.0242
MAE of doors (m)	0.1285	0.1251	0.1329	0.1375	0.1322	0.1187	0.1056	0.0878	0.0987	0.1322	0.0826	0.1213	0.1168	0.0855	0.1147 $\pm$ 0.0192
Relative error of windows (%)	6.1200	9.6100	6.1300	4.9800	3.9100	6.6200	4.3100	3.7700	4.4500	5.3100	5.6400	6.3800	7.1800	4.9100	5.6657 $\pm$ 1.5406
Relative error of doors (%)	8.4700	8.0900	8.4200	9.6400	9.6500	8.6900	7.8200	6.4200	6.3800	8.9300	6.1900	8.0500	8.7300	6.5100	7.9993 $\pm$ 1.1879
2D IoU of windows	0.7681	0.8650	0.8970	0.8039	0.8908	0.8421	0.9283	0.8886	0.9621	0.9525	0.9052	0.8515	0.9063	0.9108	0.8837 $\pm$ 0.0538
2D IoU of doors	0.8989	0.8765	0.9190	0.9216	0.9494	0.9810	0.9170	0.8894	0.8920	0.7618	0.8975	0.8618	0.9449	0.9373	0.9034 $\pm$ 0.0516
3D IoU of windows	0.6316	0.5415	0.4017	0.4535	0.4948	0.5841	0.3758	0.6174	0.4002	0.4629	0.5072	0.6462	0.5680	0.5651	0.5179 $\pm$ 0.0896
3D IoU of doors	0.5731	0.7792	0.7023	0.5829	0.6796	0.9245	0.5827	0.7861	0.3527	0.7851	0.7320	0.6737	0.3766	0.9417	0.6766 $\pm$ 0.1739

The 2D IoU was generally higher than the 3D IoU. The average 2D IoUs of the windows and doors were 0.8837 ± 0.0538 and 0.9034 ± 0.0516 , and they were 0.5179 ± 0.0896 and 0.6766 ± 0.1739 for the 3D IoUs. It is a commonly existing problem for indoor–outdoor alignment. The surfaces inside the wall are hard to fully capture by sensors. Such incomplete information is the inherent reason for a less precise position estimation perpendicular to the wall. In fact, the relative corner errors in the Y and Z directions were 7.94% and 2.79%, respectively, but in the X direction, they were as large as 35.07%. Luckily, the dimensions of the windows and doors in the X direction (the thicknesses) were small, thus the absolute location error was acceptable.

5.3. Indoor and Outdoor Model Alignment Results

Figure 11 presents the identified corresponding objects of indoor and outdoor models for each scene on the left. Accordingly, the transformation between the two models was calculated, and they were aligned without visually observable errors, as shown in the middle column. Even though the numbers of windows and doors were not equal in the indoor and outdoor models, and their distribution patterns were distinct, the presented algorithm could still identify correct matches. For example, there were four windows and one door in the indoor model of scenario 2, scene 4, but only three windows and two doors, located on perpendicular walls, were found in the outdoor model. Figure 12 presents its alignment procedure. The dimension of its adjacent matrix C_{SGD}^{out} was 5×5 . After eliminating the rows and columns with all elements being invalid, the dimension of the updated adjacent matrix became 4×4 , and the best matches could be found with Algorithm 1 accordingly. Eventually, the algorithm successfully found the corresponding three windows and one door. The eliminated window in the indoor model was window 2, its theoretically matching window in the outdoor was not recognized in the window detection phase. As for the eliminated door in the outdoor model, it was located on the perpendicular wall to the door in the indoor model, thus could not be captured in the indoor model. Compared to the geometric-feature-based matching methods, which usually need to identify the corresponding relationships among hundreds, even thousands, of geometric features, we built the alignment only based on five semantic objects. Thus, the computational efficiency was highly improved. Taking the line-based alignment algorithm presented in [7] as an example, the number of extracted line segments was $10^2 \sim 10^3$, and the computational magnitude for feature matching was $10^4 \sim 10^6$ with the classical ICP method. In comparison, the computational magnitude of the proposed SGD-based

matching method hardly reached the 10^3 level, much smaller than that of the geometric-feature-based matching methods.

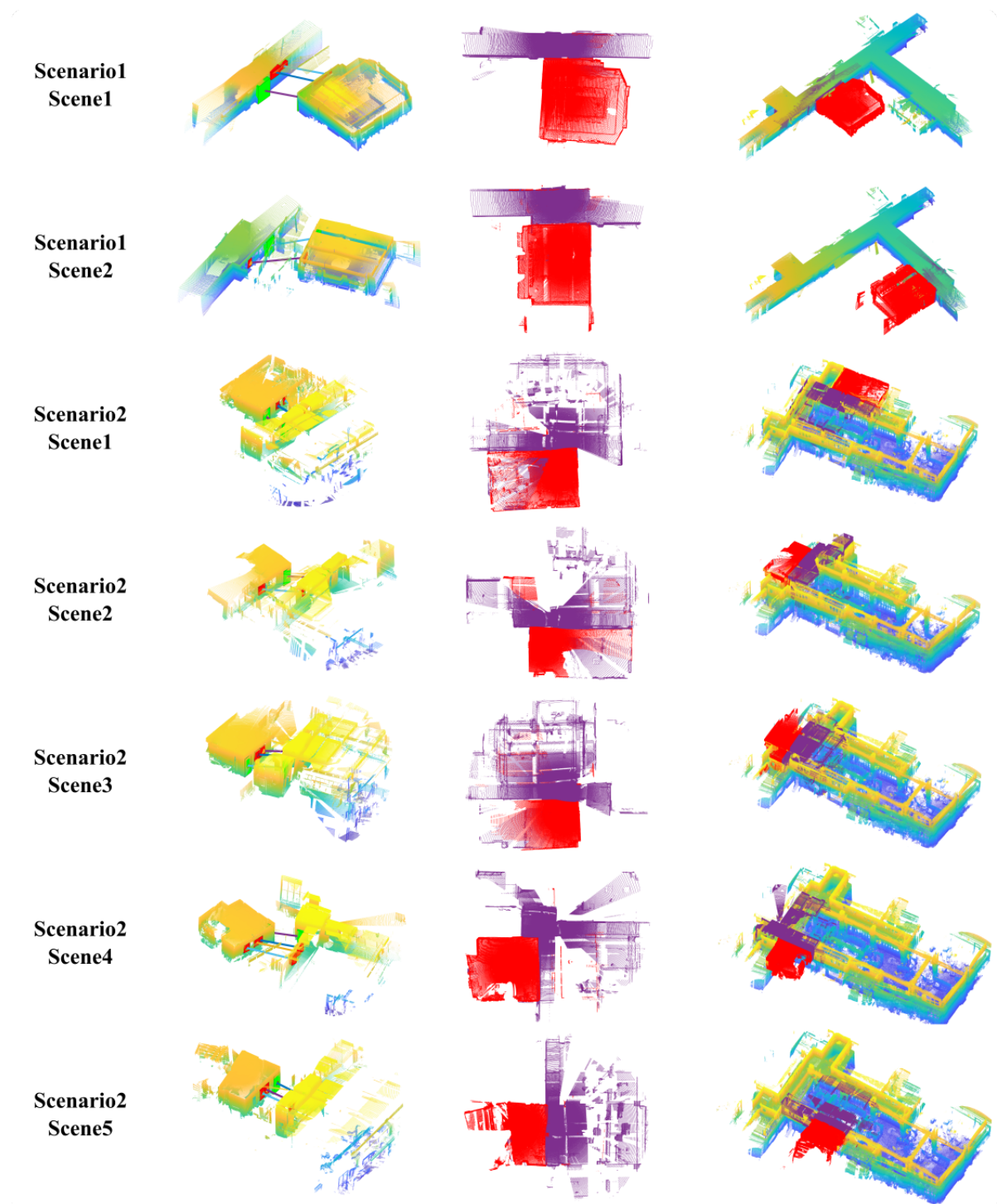


Figure 11. The alignment effect using semantic object matching results. The first column illustrates the found correspondences between indoor and outdoor models. The second column depicts the point cloud after alignment on the top view, where the indoor models are colored in red, and the outdoor models are colored in purple. The third column shows the position of single-scene alignments with respect to the full outdoor models.

For each scene, the outdoor model was fixed, and the indoor model was registered to the outdoor model with the calculated transformation matrix. The accuracy was evaluated

by the MAE of bounding box centers and corners between the identified windows/doors among indoor and outdoor models (Table 3). The center error mostly reflected the translational error. The corner error included translational and rotational errors, as well as the inevitable 3D box's dimension discrepancy between the two models. The average center error was only 0.0579 m, which achieved a centimeter level. The alignment accuracy was considerably higher compared to previous related research [16].

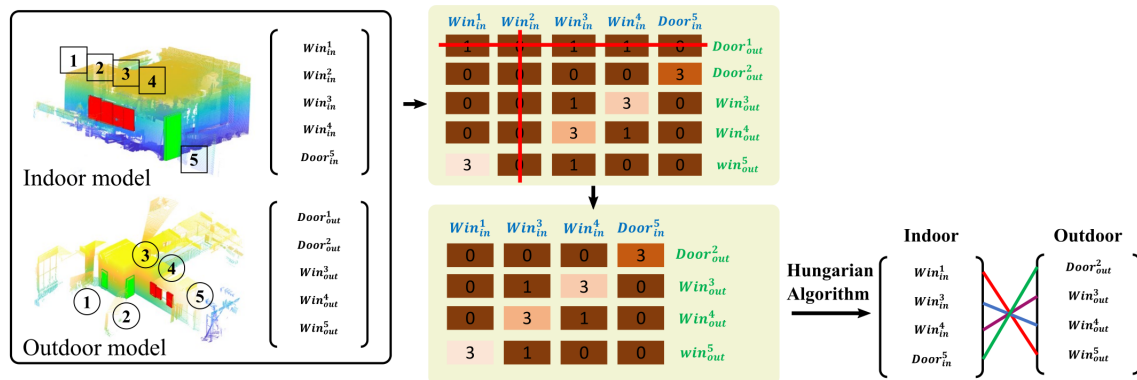


Figure 12. The alignment procedure for scenario 2, scene 4. There are four windows and one door in the indoor model and three windows and two doors in the outdoor model. The color of each element in the adjacent matrix represents the matching distance, the lighter color means a small matching distance and the darkest color means an infinite matching distance. The number in the element is the number of adjacent objects after SGDU matching. The rows and columns with all elements being invalid were eliminated (in the red line). Eventually, three windows and one door were aligned correctly.

Table 3. The accuracy of indoor and outdoor model alignment.

	Scenario 1 Scene 1	Scenario 1 Scene 2	Scenario 2 Scene 1	Scenario 2 Scene 2	Scenario 2 Scene 3	Scenario 2 Scene 4	Scenario 2 Scene 5	Mean $\pm$ std
MAE of object centers (m)	0.0638	0.0630	0.1119	0.0614	0.0284	0.0688	0.0081	0.0579 $\pm$ 0.0328
MAE of object corners (m)	0.1322	0.0821	0.1474	0.1145	0.1109	0.1221	0.1151	0.1177 $\pm$ 0.0202

As can be seen in Figures 13 and 14, the point clouds of the outdoor models were usually very complex and noisy. The indoor data, especially in scenario 2, were captured in a cluttered lab environment. Some partially blocked windows failed to be detected in the indoor view. The overlap data between the indoor and outdoor models were highly limited. Under such challenging conditions, the method proposed in this paper managed to align all rooms with the outdoor model accurately and achieved desirable reconstruction results with complicated building structures.

5.4. Evaluation on Public Dataset

To evaluate the robustness and generality of the proposed method, we compare our method with Andrea's method [16] on the same dataset. A diverse set of six datasets with indoor and outdoor models were provided in their paper, and four of them were open access. However, only the dataset "Hall" contained both indoor and outdoor models, so it was chosen as the comparison case. The point cloud in the dataset was reconstructed based on images with the structure-from-motion (SFM) approach and was not suitable for our window/door detection method. For simplicity, we followed the same procedure as in [16] to manually label the bounding boxes of windows/doors in the model. Then, the SGD-based registration was applied, and the result is shown in Figure 15. Eleven corresponding objects were identified after SGD matching, and the indoor–outdoor model registration was calculated accordingly. The MAE of registration by our method was 0.13 m and 32% smaller than that of Andrea's method, which had an MAE of 0.19 m.

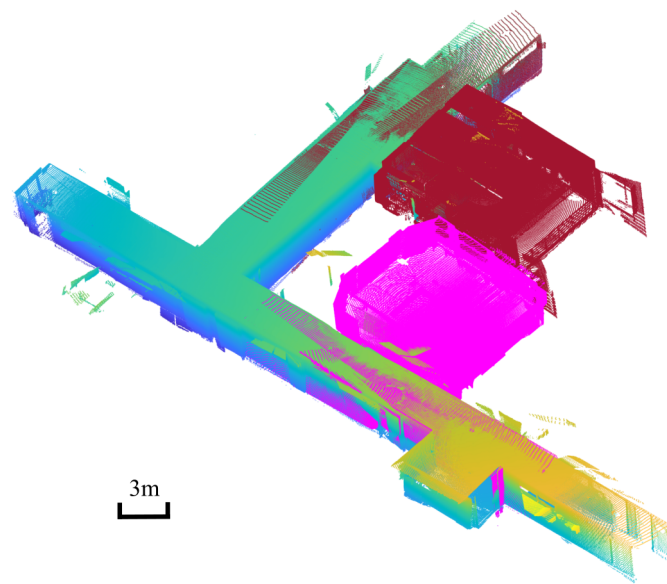


Figure 13. The registration result of scenario 1, which is an L-shape corridor with one room located on each edge. Two indoor models are colored in magenta and claret red separately. Several outdoor point clouds in scenario 1 were registered as a unified outdoor model, and the corresponding outdoor models for two scenes are colored in the same color as their indoor models.

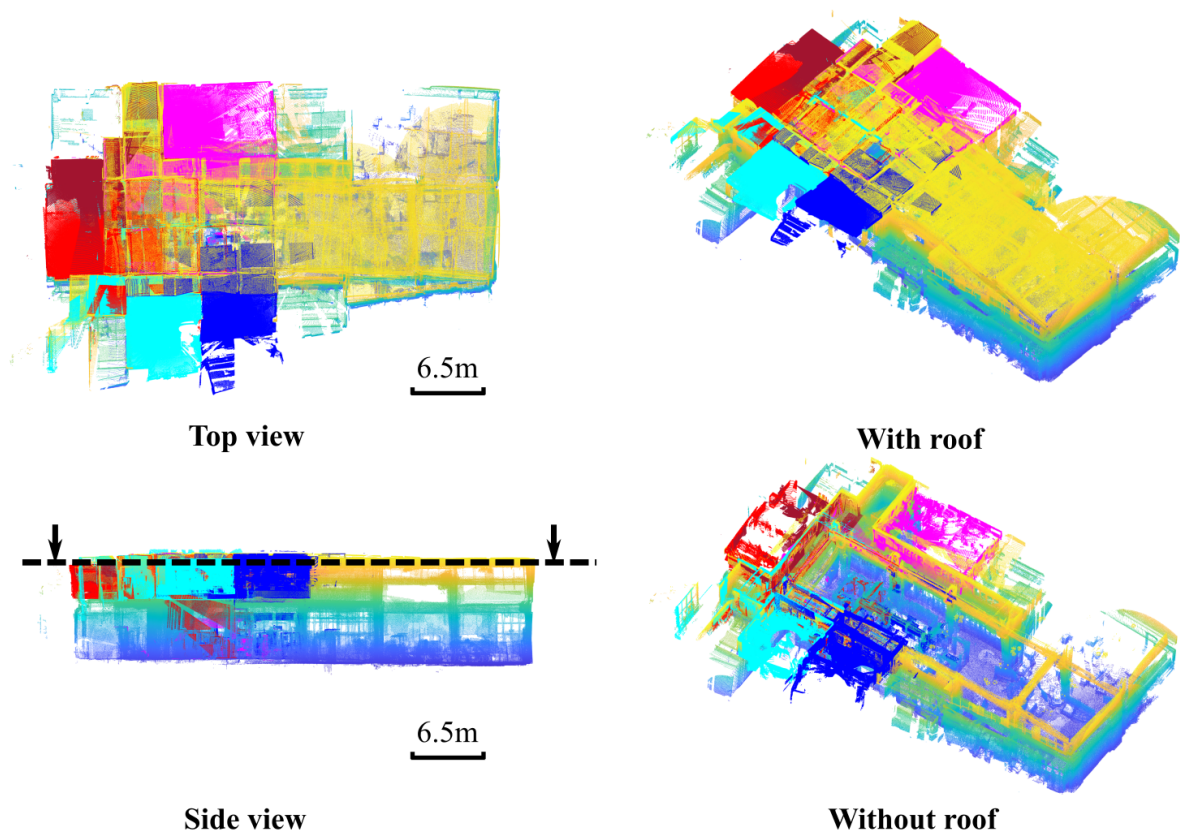


Figure 14. The registration result of scenario 2, which is a complex lab environment. The registered five scenes locate around a U-shape corridor and are colored in magenta, claret red, red, cyan and blue, respectively. By removing the roof part, the complex structure of the indoor model is shown in the bottom right of the figure. Even under such challenging conditions, the proposed method can achieve accurate registration between each indoor and outdoor model.

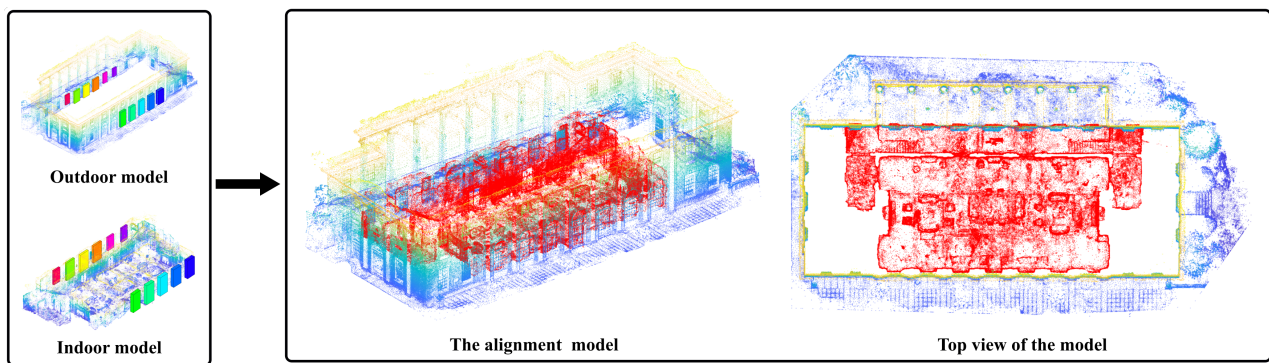


Figure 15. The registration result for the public dataset “Hall”. The windows and doors for indoor and outdoor models were labeled manually, and the registration was conducted with the proposed SGD-based method. The alignment result is illustrated on the right side of the image, and there are no visually observable errors.

The experimental result demonstrated that the proposed method could identify the corresponding windows and doors from indoor and outdoor models correctly and align them accurately. Even if semantic objects were not identical in the two models, the proposed method could still find the correct object correspondence. The accuracy of window/door detection would affect the precision of the SGD for each model significantly. The most likely cause for inaccurate window detection is the occlusion inside the window since the window detection method is based on the assumption that the laser ray can penetrate through the entire window. In addition, the resolution of the laser image is limited, and some very small windows/doors may not be recognized, which could induce a failure in the correspondence matching in extreme cases.

6. Conclusions

This study tackled the alignment problem of individual indoor and outdoor models. The proposed window and door detection method identified the semantic information and spatial relationship of windows and doors in the model accurately, and the novel SGD model captured their spatial relationships. By matching the SGDs of the indoor and outdoor models, the corresponding windows and doors were properly paired, and the transformation matrix for the indoor–outdoor alignment was obtained accordingly. Since SGD depends on high-level semantic objects, the measurement noises and irrelevant local geometric features had little influence on the values of the descriptors. Therefore, the proposed SGD matching method could output reliable alignment results for complex and noisy environment data, such as scenario 2 in Figure 14. With the proposed semantic–geometric descriptor, the average matching accuracy of indoor and outdoor model alignment reached 0.0579 m.

The SGD describes the distribution patterns for semantic objects and helps discover similarities among point cloud maps. Therefore, the method can be extended to other applications by utilizing semantic objects in more general scenes, such as typical urban views. The SGD matching method would then be able to work with sequential laser-scanning matching for unmarked map merging or loop closing for simultaneous localization and mapping in future work.

Author Contributions: Conceptualization, Y.X. and Y.Y.; methodology, Y.Y. and G.F.; software, Y.Y. and G.F.; validation, Y.Y.; writing—original draft preparation, Y.Y.; writing—review and editing, Y.X. and Z.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Shanghai, grant numbers 20ZR1419100 and 19ZR1455500, and Shanghai Shuguang Program of grant number 20SG40.

Data Availability Statement: Publicly available datasets were analyzed in this study. Available online: <https://github.com/YoungRainy/Indoor-Outdoor-Point-Cloud-Dataset.git> (accessed on 12 October 2022).

Acknowledgments: The authors of this paper would like to thank Nanyan Shen and Hongxia Cai for allowing us to scan their labs for our dataset.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Soheilian, B.; Tournaire, O.; Paparoditis, N.; Vallet, B.; Papelard, J.P. Generation of an integrated 3D city model with visual landmarks for autonomous navigation in dense urban areas. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013; IEEE: Gold Coast City, Australia, 2013; pp. 304–309. [\[CrossRef\]](#)
2. Dudhee, V.; Vukovic, V. Building information model visualisation in augmented reality. *Smart Sustain. Built Environ.* **2021**, ahead-of-print. [\[CrossRef\]](#)
3. Tariq, M.A.; Farooq, U.; Aamir, E.; Shafaqat, R. Exploring Adoption of Integrated Building Information Modelling and Virtual Reality. In Proceedings of the 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Swat, Pakistan, 24–25 July 2019; IEEE: Swat, Pakistan, 2019; pp. 1–6. doi: 10.1109/ICECCE47252.2019.8940805. [\[CrossRef\]](#)
4. Wang, C.; Wen, C.; Dai, Y.; Yu, S.; Liu, M. Urban 3D modeling with mobile laser scanning: A review. *Virtual Real. Intell. Hardw.* **2020**, *2*, 175–212. [\[CrossRef\]](#)
5. Cao, Y.; Li, Z. Research on Dynamic Simulation Technology of Urban 3D Art Landscape Based on VR-Platform. *Math. Probl. Eng.* **2022**, *2022*, 3252040. [\[CrossRef\]](#)
6. López, F.J.; Leronés, P.M.; Llamas, J.; Gómez-García-Bermejo, J.; Zalama, E. A review of heritage building information modeling (H-BIM). *Multimodal Technol. Interact.* **2018**, *2*, 21. [\[CrossRef\]](#)
7. Koch, T.; Korner, M.; Fraundorfer, F. Automatic alignment of indoor and outdoor building models using 3D line segments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 10–18.
8. Djahel, R.; Vallet, B.; Monasse, P. Detecting Openings For Indoor/Outdoor Registration. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, XLIII-B2-2022, 177–184. [\[CrossRef\]](#)
9. Assi, R.; Landes, T.; Murtiyoso, A.; Grussenmeyer, P. Assessment of a Keypoints Detector for the Registration of Indoor and Outdoor Heritage Point Clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 133–138. [\[CrossRef\]](#)
10. Pan, Y.; Yang, B.; Liang, F.; Dong, Z. Iterative global similarity points: A robust coarse-to-fine integration solution for pairwise 3d point cloud registration. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 180–189.
11. Li, Z.; Zhang, X.; Tan, J.; Liu, H. Pairwise Coarse Registration of Indoor Point Clouds Using 2D Line Features. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 26. [\[CrossRef\]](#)
12. Djahel, R.; Vallet, B.; Monasse, P. Towards Efficient Indoor/outdoor Registration Using Planar Polygons. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *2*, 51–58. [\[CrossRef\]](#)
13. Previtali, M.; Barazzetti, L.; Brumana, R.; Scaioni, M. Laser scan registration using planar features. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Riva del Garda, Italy, 23–25 June 2014; Volume 45.
14. Favre, K.; Pressigout, M.; Marchand, E.; Morin, L. A plane-based approach for indoor point clouds registration. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 7072–7079.
15. Malihi, S.; Valadan Zoej, M.J.; Hahn, M.; Mokhtarzade, M. Window detection from UAS-derived photogrammetric point cloud employing density-based filtering and perceptual organization. *Remote Sens.* **2018**, *10*, 1320. [\[CrossRef\]](#)
16. Cohen, A.; Schönberger, J.L.; Speciale, P.; Sattler, T.; Frahm, J.M.; Pollefeys, M. Indoor-outdoor 3d reconstruction alignment. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 285–300.
17. Geng, H.; Gao, Z.; Fang, G.; Xie, Y. 3D Object Recognition and Localization with a Dense LiDAR Scanner. *Actuators* **2022**, *11*, 13. [\[CrossRef\]](#)
18. Cai, Y.; Fan, L. An efficient approach to automatic construction of 3D watertight geometry of buildings using point clouds. *Remote Sens.* **2021**, *13*, 1947. [\[CrossRef\]](#)
19. Imanullah, M.; Yuniarno, E.M.; Sumpeno, S. Sift and icp in multi-view based point clouds registration for indoor and outdoor scene reconstruction. In Proceedings of the 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 28–29 August 2019; pp. 288–293.
20. Xiong, B.; Jiang, W.; Li, D.; Qi, M. Voxel Grid-Based Fast Registration of Terrestrial Point Cloud. *Remote Sens.* **2021**, *13*, 1905. [\[CrossRef\]](#)
21. Popișter, F.; Popescu, D.; Păcurar, A.; Păcurar, R. Mathematical Approach in Complex Surfaces Toolpaths. *Mathematics* **2021**, *9*, 1360. [\[CrossRef\]](#)
22. Wen, C.; Sun, X.; Hou, S.; Tan, J.; Dai, Y.; Wang, C.; Li, J. Line structure-based indoor and outdoor integration using backpacked and TLS point cloud data. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1790–1794. [\[CrossRef\]](#)

23. Chen, S.; Nan, L.; Xia, R.; Zhao, J.; Wonka, P. PLADE: A plane-based descriptor for point cloud registration with small overlap. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 2530–2540. [[CrossRef](#)]
24. Li, J.; Huang, S.; Cui, H.; Ma, Y.; Chen, X. Automatic point cloud registration for large outdoor scenes using a priori semantic information. *Remote Sens.* **2021**, *13*, 3474. [[CrossRef](#)]
25. Parkison, S.A.; Gan, L.; Jadidi, M.G.; Eustice, R.M. Semantic Iterative Closest Point through Expectation-Maximization. In Proceedings of the BMVC, Newcastle, UK, 3–6 September 2018; p. 280.
26. Wang, L.; Sohn, G. An integrated framework for reconstructing full 3d building models. In *Advances in 3D Geo-Information Sciences*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 261–274.
27. Wang, L.; Sohn, G. Automatic co-registration of terrestrial laser scanning data and 2D floor plan. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2010**, *38*, 158–164.
28. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 12697–12705.
29. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; pp. 5105–5114.
30. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE international Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
31. Dey, E.K.; Awrangjeb, M.; Stantic, B. Outlier detection and robust plane fitting for building roof extraction from LiDAR data. *Int. J. Remote Sens.* **2020**, *41*, 6325–6354. [[CrossRef](#)]
32. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [[CrossRef](#)]
33. Bay, H.; Tuytelaars, T.; Gool, L.V. Surf: Speeded up robust features. In Proceedings of the European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; pp. 404–417.
34. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
35. Cao, B.; Wang, J.; Fan, J.; Yin, J.; Dong, T. Querying similar process models based on the Hungarian algorithm. *IEEE Trans. Serv. Comput.* **2016**, *10*, 121–135. [[CrossRef](#)]
36. Xie, Y.; Tang, Y.; Zhou, R.; Guo, Y.; Shi, H. Map merging with terrain-adaptive density using mobile 3D laser scanner. *Robot. Auton. Syst.* **2020**, *134*, 103649. [[CrossRef](#)]