Align to locate: Registering photogrammetric point clouds to BIM for robust indoor localization

Junjie Chen [a], Shuai Li [b, *], Weisheng Lu [a]

[a] Department of Real Estate and Construction, The University of Hong Kong, Hong Kong, China
[b] Department of Civil and Environmental Engineering, The University of Tennessee, Knoxville, TN, USA

Abstract

Indoor localization is critical for many smart applications in built environments such as service robot navigation and facility management. Building information models (BIM) provide new streams of spatial and appearance information regarding building interiors that can be exploited for robust indoor localization. However, previous localization methods using BIM were unable to achieve high precision and accuracy, limiting their practical applications. To address this challenge, a new approach, "align-to-locate (A2L)", was proposed in this study to leverage BIM as a reference to rectify and finetune coarse camera poses estimated photogrammetry. The camera pose rectification is achieved by a new registration algorithm that aims to align a photogrammetric point cloud with a BIM-referenced point cloud. The experiments demonstrated the effectiveness of the proposed A2L approach, which outperformed the state of the art with the localization error of 1.07 m and the orientation deviation of 3.7°. It was also found that query point clouds generated from photographs taken along the lateral or longitude directions are more conducive for registration. Increasing the number of data collection locations and images from each location could lead to higher accuracy, but may compromise the computational speed. This study contributes to the challenging indoor localization problem by proposing the "align-to-locate" concept and evaluating its feasibility for more robust camera pose estimation through point cloud-to-BIM registration. The developed A2L approach can be integrated as a post-processing module in existing vision-based localization methods to finetune their estimated camera poses.

Keywords: Smart building; Location-based services; Indoor localization; Building information model (BIM); Point cloud; Registration.

[*] Corresponding author.
E-mail address: sli48@utk.edu (S. Li).

1. Introduction

Many industrial or daily activities in built environments relies on robust indoor positioning services. An example is pedestrian navigation in large commercial buildings, where visitors need to quickly access their destinations with the help of wayfinding smartphone programs [1, 2]. For facility maintenance, precise position information is a prerequisite for augmented reality devices to retrieve corresponding contents to assist decision-making. Robots are increasingly being used for various scenarios such as construction progress inspection [3], cleaning and sterilizing [4], and comfort monitoring [5] in buildings. To enable such advanced applications, localization is an indispensable module for the robots to understand their positions in the environment. Because of the occluded and bounded nature of built environments, indoor localization is challenging. Traditional techniques based on Wi-Fi, Bluetooth, and radio frequency identification (RFID) are not only subject to severe deviations, but also requires large investment on installing and maintaining external signal emission infrastructure [6].

Compared with traditional techniques, vision-based approaches [7-10] stand out for its cost-effectiveness, and being infrastructure-independent [6]. Given one or multiple photos of a scene, such approaches can recover its or their corresponding camera poses when the photos were taken. In the most common settings, however, these approaches require a pre-mapping of the environment of interest so as to estimate the camera pose in a global reference system. The pre-mapping operation is tedious and expensive to implement, hindering the wide adoption of such techniques. In recent years, the wide adoption of building information model (BIM) [11] makes it possible to fully deliver the strengths of visual localization without the need of the labor-intensive pre-mapping. BIM serves as a readily available source of a wide spectrum of geospatial building information [12-14], including not only visual appearance (i.e., images), but also geometry and spatial layout (i.e., position) of indoor environments.

Therefore, instead of collecting real-life photos in the field, latest research [3, 15-18] sought to exploit the information in BIM to enable visual indoor localization. To overcome the challenge of a cross-domain gap between BIM and real photographs, Ha et al. [15] proposed an image retrieval approach based on deep transfer learning features for the task of indoor localization. Chen et al. [17] demonstrated the feasibility of generative adversarial networks (GAN) in bridging the cross-domain gap, and proposed a photogrammetric approach to estimating six degrees of freedom (6DoF) camera pose based on information retrieved from a style-transfer BIM. Asadi et al. [3] inferred indoor positions of inspection robots by aligning perspective vanishing points of video frames and BIM-rendered views. Inspired by [19], Acharya et al. [16, 20] and Zhao et al. [21] performed a series of works to regress 6DoF camera pose via convolutional neural networks (CNN) trained on BIM-rendered images. Despite the progress, the precision of existing BIM-enabled visual localization is still not adequate. To accomplish demanding tasks such as service robot

navigation in the built environment, a new solution with more robust localization performance is necessary.

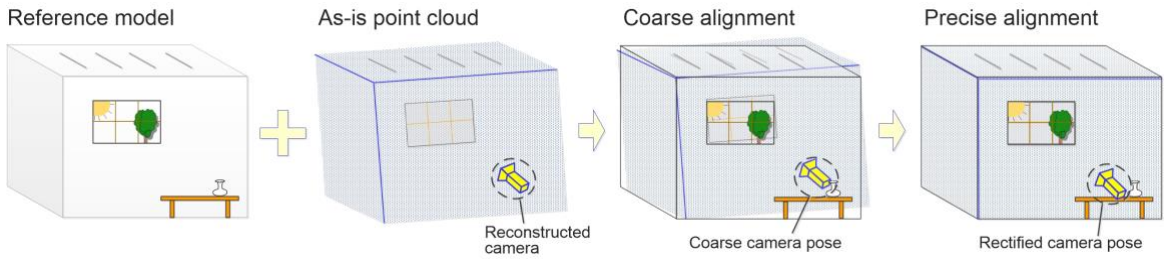

Fig. 1. The conceptual diagram of the "align-to-locate" approach for robust indoor localization.

Existing BIM-enabled solutions mainly focused on matching visual features extracted from one or serval BIM-rendered views with features obtained from corresponding camera poses. However, other information or features that could have been extracted from BIM to rectify camera pose estimation have not been fully exploited. One example is the three-dimensional (3D) geometry of an indoor space formed by its surrounding walls, columns, and (or) floor and ceiling. As shown by Fig. 1, it is straightforward to separate a reference geometric model from BIM for any indoor spaces in a building. As for the as-is status of the space in real life, a photogrammetric point cloud (PC) can be easily generated from image sequences or videos of a subject's surrounding based on the structure-from-motion (SfM) technique. By aligning the as-is PC with the reference model, the initial camera poses estimated by any previous vision-based approach [3, 15-17] can be rectified, and thus the subject's position can be precisely located. Although this "align to locate" concept seems promising, few studies have explored how a photogrammetric PC, representing only a part of the entire environment with much data noise, can be registered to BIM for camera pose rectification and robust indoor localization.

To fill the knowledge gap, this study aims to investigate a new mechanism for registering photogrammetric PCs to BIM for camera pose estimation, and analyze the influences of various data collection strategies on localization performance. This study contributes to the body of knowledge for indoor localization by proposing a novel "algin-to-locate (A2L)" approach to precisely estimating 6DoF camera poses based on a collection of photographs. The feasibility of the A2L approach was experimentally tested and evaluated, which achieved a 1.07 m localization error and a 3.7° orientation deviation. The proposed approach can be integrated with existing visual localization methods as a post-processing module to finetune the estimated camera poses to a precision level applicable in demanding tasks such as service robot navigation and AR-assisted inspection.

2. Related work

2.1. Vision-based indoor localization

The potential of machine vision in indoor localization has long been acknowledged for its cost-effectiveness and independence from external infrastructure. The classical simultaneous localization and mapping (SLAM) based on a single camera [7] and visual odometry (VO) algorithm [8] were proposed to estimate robots' ego-motion and their positions in unknown scenes by continuously triangulating feature correspondences among sequential camera frames. The incremental nature of such algorithms has decided that they can only yield a subject's position relative to a local coordinate system [6, 16]. To locate the subject in a global reference frame, research efforts have been made in visual indoor localization. One line of such efforts considered the indoor localization task as a content-based image retrieval problem [15, 22, 23], in which a database of geo-registered photographs of the built environment has been collected in prior, and a camera pose of a newly image is determined by retrieving its most similar counterpart from the database. Another stream of works first reconstruct a 3D PC model of the environment by applying SfM. With the PC model as a reference, 6DoF camera pose corresponding to the query image can then be estimated either by stereo triangulation [9, 24] or training a regression model based on CNN [19].

A limitation of the above approaches is their requirement for pre-mapping the built environment, either to obtain geo-registered photographs or point clouds. To avoid the tedious pre-mapping operations, latest research sought to directly extract such reference information about the environment from a building information model. While replacing real-life photographs with synthetic ones rendered by BIM seems a straightforward solution, it has been proved very difficult due to a perception gap between the two domains [15, 17]. To address the issue, Ha et al. [15] investigated the feature maps extracted by various layers in VGG, a well-known CNN architecture. They found that the deep features from pooling layer 4 performs best in bridging the cross-domain gap, and can enable accurate retrieval of BIM-rendered images for indoor localization. In [16, 20, 21], the authors used edge maps of BIM-rendered images, instead of the original BIM views, as training data to develop their camera pose regression model. When similar edge maps of input real photographs were used for inference, a localization error of 1.6~2.0 m and an orientation deviation of 7°~ 11° were obtained. Different from previous studies, Chen et al. [17] attempted to address the perception gap by converting textureless BIM views to ones with photorealistic texture by style transfer technique based on GAN. Their experiments demonstrate effectiveness of the style-transfer BIM in facilitating the exploitation of the rich information in BIM by traditional image features such as scale-invariant feature transform (SIFT) and edge histogram descriptor (EHD), and achieved a localization error of 1.38 m.

Although great progress has been made in enabling visual localization with BIM, the performance

is still not sufficient for tasks having high requirements on localization precision. Such tasks include robot navigation in occluded indoor environments [4, 5] and AR-based facility maintenance [25]. To achieve higher precision, other information contained in BIM should be better exploited, and one aspect that can potentially contribute is the 3D geometry of an indoor space. By registering a photogrammetric point cloud into BIM, the coarse camera pose estimated by previous methods can be further rectified.

2.2. Point cloud to BIM registration

Point cloud registration is a general problem encountered in many applications such as autonomous driving, panorama stitching and robotics, and has been investigated for decades in the computer science community. One classical solution for PC registration is the iterative closest point (ICP) algorithm [26], which iteratively searches for an optimal rigid transformation that can minimize the overall distance among closest points between two clouds. However, ICP performs best only if the query PC is sufficiently close to the reference PC, or referred to as the problem of fine registration in [27]. For the more challenging problem of global registration, research efforts have been made, including a series of variants developed from ICP, e.g., Sparse ICP [28] and Go-ICP [29], and methods based on matching the salient features in PC, e.g., the '4-point congruent sets' (4-PCS) algorithm [30]. However, there is still no universally applicable robust solution for automated PC registration.

In the architecture, engineering, construction, and operation (AECO) sector, the registration of PC to BIM (PC2BIM) becomes an active research field with the proliferation of BIM. Essentially, the PC2BIM registration problem can be transformed to a PC2PC problem after quantizing the BIM meshes into points [27]. Leveraging the domain-specific characteristics (symmetry and regularities) in architecture, numerous research efforts have been made to register as-built or as-is point cloud to BIM for various applications. One such application that attracts most attentions is construction progress control, which enables the detection of construction deviation by aligning an as-built PC with an as-designed BIM. For the purpose of deviation measurement, Chen and Cho [31] proposed a method to register a laser-scanned PC with the corresponding BIM by aligning the detected columns from the two models. Kim et al. [32] proposed an algorithm pipeline, which involves pre-processing, global registration based on principal component analysis (PCA) and local registration based on ICP, to allows intuitive construction progress monitoring with the aligned PC and BIM. Bueno et al. [27] took the uniqueness of construction buildings into account, and developed the '4-Plane congruent Set' (4-PlCS) algorithm for the global registration of laser scanning data with BIM, which can be used for construction quality and progress control.

Other research endeavors aimed to facilitate AR-assisted facility maintenance [33, 34] and semantic enrichment of digital models by PC to BIM registration [35, 36]. Kopsida and Brilakis

[34] presented a semiautomated markerless solution to alignment as-is context captured by RGB-D cameras with BIM for AR-based inspection. To achieve similar AR applications, Mahmood et al. [33] developed an automated registration approach based on geometric features, which was validated with PC scanned by Microsoft HoloLens. Xue et al. [35, 36] conducted a series of researches to register as-is point cloud with as-designed drawings or element models for semantic enrichment of digital twin city. Despite the extensive research input, much remains unclear how the PC2BIM registration can be used for robust indoor camera pose estimation.

2.3. Knowledge gap

The literature review revealed three aspects of knowledge gaps. First, existing BIM-enabled visual indoor localization methods are not well-established, presenting much room for precision improvement. Such improvement will enable demanding tasks that require high localization performance such as navigating a service robot in the built environment.

Second, prior PC2BIM registration studies mainly focused on scenarios such as construction deviation checking [27, 31, 32] that are implemented offline with dense PCs of the entire space collected by laser scanners over a certain period. These methods are not readily extendable to indoor localization because of 1) the shorter processing time required, 2) the sparse point cloud generated, and 3) the partial space represented by the point cloud. Existing methods fall short of registering such partial PCs to BIM models, nor have they investigated how to use the registration to rectify a coarse camera pose to improve localization precision.

Third, dense PCs are usually generated by laser scanning [27, 31] or RGB-D cameras [34] in prior studies. However, for a photogrammetric point cloud, its quality (e.g., data noise and point density) may be compromised as the SfM reconstruction results can be impacted by the way raw photographs are taken, which subsequently affects the precision of registration and localization performance. Little research has been done to investigate effects of different data collection schemes on the camera pose estimation precision.

3. Methods

3.1. Preparing a referenced database for registration

In order to implement the proposed approach, a referenced database needs to be constructed from the original BIM model. The referenced database will serve as the target of registration in later steps. As shown in Fig. 2, the preparation of the database involves the following steps. First, the entire BIM is divided into many model units. This division is necessary because of the partial nature of the as-is point cloud reconstructed by SfM. Without it, the partial point cloud will be directly aligned with the entire BIM model, potentially impairing the registration performance due to the interference of building elements that are not captured in the partial cloud. Each individual

room with closed space is divided as a separate model unit. As for other open areas with relatively large floor space, e.g., corridors, they are also divided to obtain separate parts with relatively regular shapes. Second, the mesh model of each unit is downsampled into a point cloud for the convenience of registration. This "mesh-to-point" operation is a widely adopted practice in existing studies [27, 32, 33]. Finally, the boundary coordinates and the range of elevation are extracted for each model unit as its corresponding metadata. The metadata can ensure that corresponding reference point cloud will be quickly indexed and retrieved with initial camera pose.

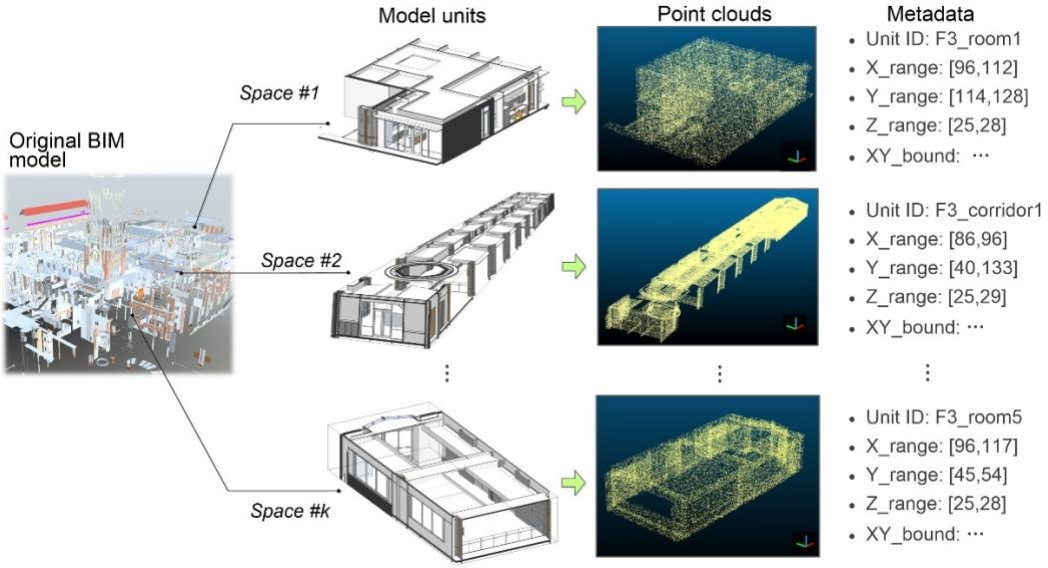

Fig. 2. Preparing a database of reference point clouds from original BIM.

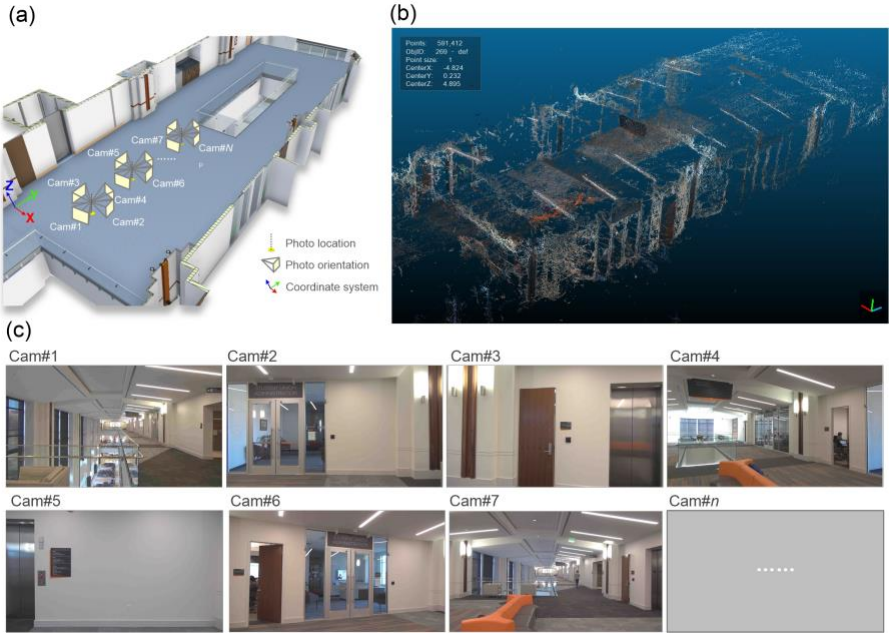

Fig. 3. (a) An example photo-taken strategy which collect data along a longitude direction; (b) The generated as-is query point cloud; (c) The collected Photos from locations marked in (a).

3.2. Generating as-is query point cloud

A point cloud of the as-is built environment is generated with the SfM technique. The point cloud, referred to as a "query" point cloud, will be used to match and align with the reference model. The query point cloud might have undesired noise and outliers, which can potentially impair the registration in later steps. Therefore, the sparse outlier removal (SOR) algorithm introduced in [37] is applied to denoise the raw point cloud. In addition, various strategies can be used to collect photos for generating the query point cloud, and Fig. 3 shows one example of such strategies. Different strategies can result in point clouds of different quality, which will then lead to different registration performance, and ultimately affect camera pose estimation accuracy. In later part of this study, a sensitivity analysis will be performed to find the best data collection practice. For each collected photo, a corresponding initial camera pose can be coarsely estimated with previous vision-based approaches such as [15], [16], and [17]. The initial camera pose will be used for coarse registration in next step.

3.3. Coarse registration based on initial camera pose information

A photogrammetric point cloud based SfM rationale is one with undetermined scale and has a coordinate system inconsistent with the global system used by the reference model, as demonstrated by Fig. 4 (a). However, it preserves the spatial relativity between the point cloud and the photo capture locations. With the initial camera pose estimated by previous approaches, it is viable to coarsely align the query point cloud with the reference counterpart, as depicted by the process from Fig. 4 (a) to (b).

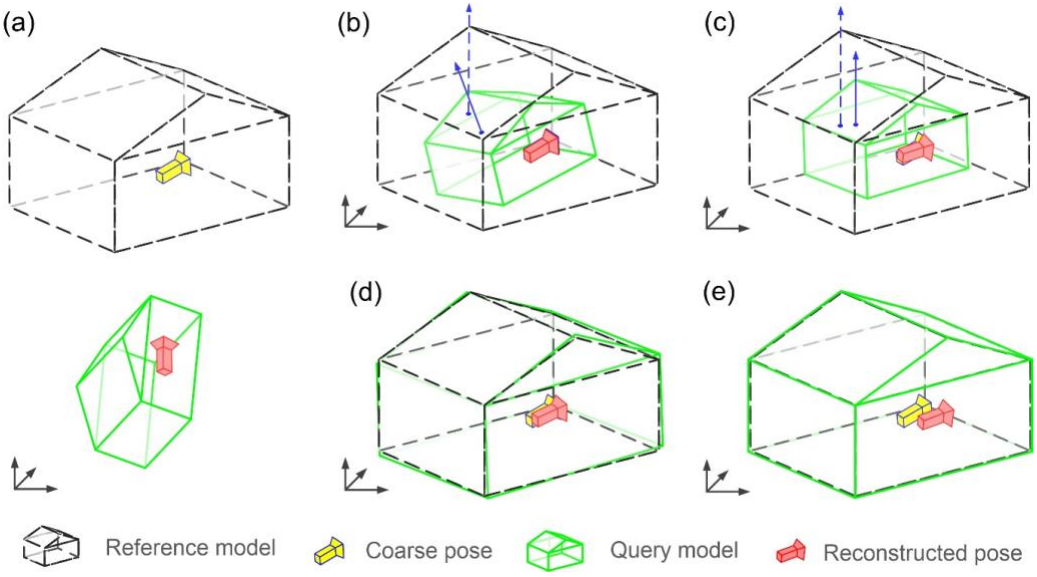

Fig. 4. (a) The inconsistent scale and coordinate system between reference and query point clouds; (b) Results of coarse registration; (c) Results of orientation alignment; (d) Results of scale

normalization; (e) Results of finetuning alignment.

Suppose there are totally N_P photos that have been used to generate the query point cloud, from which we can randomly select N_C subsamples for coarse registration. Let $\{\mathbf{C}_Q^i\}(i=1,2,...,N_C)$ and $\{\mathbf{C}_{coar}^i\}(i=1,2,...,N_C)$ denote the transformation matrices of camera poses reconstructed in the query point cloud and estimated by coarse localization approaches, respectively. The transformation matrices are in a homogeneous form as illustrated by Eq. (1):

$$\mathbf{C}_{coar}^i = \begin{bmatrix} rc_{11}^i & rc_{12}^i & rc_{13}^i & 0 \\ rc_{21}^i & rc_{22}^i & rc_{23}^i & 0 \\ rc_{31}^i & rc_{32}^i & rc_{33}^i & 0 \\ tc_x^i & tc_y^i & tc_z^i & 1 \end{bmatrix} \tag{1}$$

Where $\begin{bmatrix} rc_{11}^i & rc_{12}^i & rc_{13}^i \\ rc_{21}^i & rc_{22}^i & rc_{23}^i \\ rc_{31}^i & rc_{32}^i & rc_{33}^i \end{bmatrix}$ and $\begin{bmatrix} tc_x^i & tc_y^i & tc_z^i \end{bmatrix}$ are respectively rotation matrix and translation vector.

Suppose the database of reference point clouds is represented as $\{PC_R^k\}(k=1,2,...,N_{RPC})$, where N_{RPC} is the total number of reference point clouds in the database. Then, the point cloud which has covered $\begin{bmatrix} tc_x^i & tc_y^i & tc_z^i \end{bmatrix}$ within its boundaries will be selected as the target registration reference $PC_R^{k_0}$. Note that the selection results of different photos might not coincide with each other; such case can be resolved with a majority vote mechanism—selecting the reference point cloud with the most $\begin{bmatrix} tc_x^i & tc_y^i & tc_z^i \end{bmatrix}$ $(i=1,2,...,N_C)$ falling inside.

With the registration target ready, the initial transformation matrix $\mathbf{T}_{init}$ for coarse registration can be determined as follows:

$$\begin{cases} \mathbf{T}_{init}^i = (\mathbf{C}_Q^i)^{-1}\mathbf{C}_{coar}^i, \ (i=1,2,...,N_C) \\ \text{s.t.} \ \min\limits_{\mathbf{T}_{init}^i}(rmse(PC_{init}^i, PC_R^{k_0})) \end{cases} \tag{2}$$

Where $rmse(PC_1,PC_2)$ is the root mean square error (RMSE) between two point clouds; PC_{init}^i is

the resulting point cloud after applying the initial transformation matrix $\mathbf{T}_{\text{init}}^{i}$ to the original query point cloud PC_{Q}. Let i_0 denote the final selected camera pose for coarse alignment, then the adopted initial transformation matrix is $\mathbf{T}_{\text{init}}^{i_0}$, and the query point cloud after transformation is $PC_{\text{init}}^{i_0}$.

3.4. Precise registration

The coarsely aligned point cloud PC_{Q} is further processed for precise alignment with reference point cloud $PC_{\text{R}}^{k_0}$. The procedure includes three steps, i.e., orientation alignment, scale normalization, and alignment finetune.

3.4.1. Orientation alignment based on principal component analysis

The first step of precise registration is to align the point cloud pairs along the elevation direction (i.e., the *Z* axis), as depicted by the process from Fig. 4 (b) to (c). The rationale of using *Z* axis as the direction for alignment is twofold. First, in indoor localization scenarios, the collected query point cloud tends to incomplete, representing only a part of the reference space. Because of the characteristics, the building elements along z axis (i.e., ceiling and floor) have the highest chance to be captured in the point cloud. Second, compared with other axis, architecture design follows a certain regularity along the z axis, with a relatively stable floor height among different stories. This can be made used of to normalize the point cloud scale in later section.

Principal component analysis (PCA) is a widely used dimension reduction technique, which can find the most representative components with high degree of variance from the original features. It does so by producing linear combinations of the original variables to generate the components, and ordering them by their eigenvalues. The area of architecture follows the general Manhattan-world assumption for built environment, which states that there exist three dominant axes orthogonal to each other in manmade structure. PCA is an ideal technique to find such dominant axes (or components) from a cloud of points representing their spatial layout. Let $\mathbf{v}_{\text{init}}$ and $\mathbf{v}_{\text{R}}$ denote principal components along the elevation direction for $PC_{\text{init}}^{i_0}$ and $PC_{\text{R}}^{k_0}$, respectively. Then we have:

$$\mathbf{Rz}_{\text{init}} = rotmat([0 \quad 0 \quad 1], \mathbf{v}_{\text{init}}) = \begin{bmatrix} rz_{11}^{\text{init}} & rz_{12}^{\text{init}} & rz_{13}^{\text{init}} \\ rz_{21}^{\text{init}} & rz_{22}^{\text{init}} & rz_{23}^{\text{init}} \\ rz_{31}^{\text{init}} & rz_{32}^{\text{init}} & rz_{33}^{\text{init}} \end{bmatrix} \tag{3}$$

$$\mathbf{Rz}_{\mathrm{R}} = rotmat([0 \quad 0 \quad 1], \mathbf{v}_{\mathrm{R}}) = \begin{bmatrix} rz_{11}^{\mathrm{R}} & rz_{12}^{\mathrm{R}} & rz_{13}^{\mathrm{R}} \\ rz_{21}^{\mathrm{R}} & rz_{22}^{\mathrm{R}} & rz_{23}^{\mathrm{R}} \\ rz_{31}^{\mathrm{R}} & rz_{32}^{\mathrm{R}} & rz_{33}^{\mathrm{R}} \end{bmatrix} \tag{4}$$

Where *rotmat*(**a**,**b**) is a function calculates the rotation matrix from **a** to **b**; hence, $\mathbf{Rz}_{\mathrm{init}}$ and $\mathbf{Rz}_{\mathrm{R}}$ represent rotation matrices from the unit vector along *Z* axis to $\mathbf{v}_{\mathrm{init}}$ and $\mathbf{v}_{\mathrm{R}}$, respectively. Then the corresponding homogeneous transformation matrices can be obtained by incorporating coarsely estimated camera location $\begin{bmatrix} tc_x^{i_0} & tc_y^{i_0} & tc_z^{i_0} \end{bmatrix}$:

$$\mathbf{Tz}_{\mathrm{init}} = \begin{bmatrix} rz_{11}^{\mathrm{init}} & rz_{12}^{\mathrm{init}} & rz_{13}^{\mathrm{init}} & 0 \\ rz_{21}^{\mathrm{init}} & rz_{22}^{\mathrm{init}} & rz_{23}^{\mathrm{init}} & 0 \\ rz_{31}^{\mathrm{init}} & rz_{32}^{\mathrm{init}} & rz_{33}^{\mathrm{init}} & 0 \\ tc_x^{i_0} & tc_y^{i_0} & tc_z^{i_0} & 1 \end{bmatrix} \tag{5}$$

$$\mathbf{Tz}_{\mathrm{R}} = \begin{bmatrix} rz_{11}^{\mathrm{R}} & rz_{12}^{\mathrm{R}} & rz_{13}^{\mathrm{R}} & 0 \\ rz_{21}^{\mathrm{R}} & rz_{22}^{\mathrm{R}} & rz_{23}^{\mathrm{R}} & 0 \\ rz_{31}^{\mathrm{R}} & rz_{32}^{\mathrm{R}} & rz_{33}^{\mathrm{R}} & 0 \\ tc_x^{i_0} & tc_y^{i_0} & tc_z^{i_0} & 1 \end{bmatrix} \tag{6}$$

With $\mathbf{Tz}_{\mathrm{init}}$ and $\mathbf{Tz}_{\mathrm{R}}$, the transformation matrix for orientation alignment can be obtained according to Eq. (7).

$$\mathbf{T}_{PCA} = (\mathbf{Tz}_{\mathrm{init}})^{-1}\mathbf{Tz}_{\mathrm{R}} \tag{7}$$

Applying $\mathbf{T}_{PCA}$ to $PC_{\mathrm{init}}^{i_0}$, we can obtain a *Z* direction aligned query point cloud denoted by PC_{PCA}.

3.4.2. Scale normalization

After aligning the pair of point clouds along *Z* axis, the scale of the query point cloud is normalized to the same level as its reference counterpart, as depicted by the process from Fig. 4 (c) to (d). The scale normalization is conducted to equalize story height of the two point clouds. To obtain story height, searching for the highest and lowest points along the *Z* axis and subtracting the two sounds like a straightforward method, but is not viable due to the existence of noise. Inspired by [36, 38], a histogram-fit approach is proposed to determine story height of a point cloud, as shown in Fig. 5. The distribution histogram of the *Z* component of all points is generated. In most common settings, the distribution will concentrate on the ceiling and floor regions, corresponding to the two

most important elements for height calculation. Next, the histogram is fitted by a polynomial curve with degree d (e.g., $d = 8$), which should not be too small so as to find sufficient peaks. After fitting, the peaks (i.e., local maxima) of the curve are detected and sorted in a descending order. The z values of the top two peaks correspond to the elevation of the ceiling and floor, respectively, and the story height can be obtained by subtracting them.

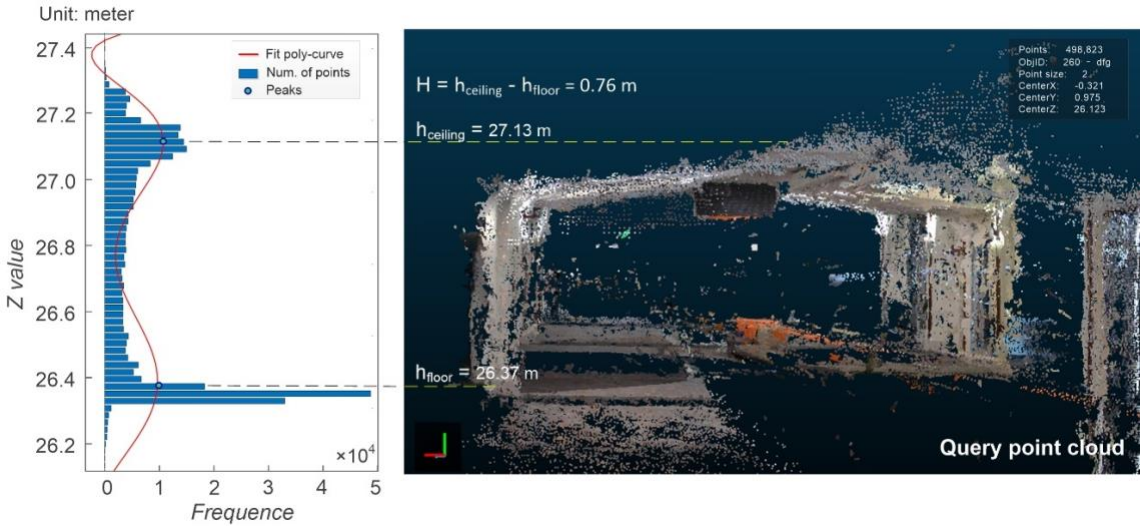

Fig. 5. The proposed histogram-fit approach to determining story height.

Let h_{R} and h_{PCA} respectively denote the story height of $PC_{\mathrm{R}}^{k_0}$ and PC_{PCA} derived from the aforementioned approach. Then, the scaling factor θ_{s} and corresponding transformation matrix $\mathbf{T}_{\mathrm{scale}}$ can be calculated as follows:

$$\theta_{\mathrm{s}} = \frac{h_{\mathrm{R}}}{h_{\mathrm{PCA}}} \tag{8}$$

$$\mathbf{T}_{\mathrm{scale}} = \begin{bmatrix} \theta_{\mathrm{s}} & 0 & 0 & 0 \\ 0 & \theta_{\mathrm{s}} & 0 & 0 \\ 0 & 0 & \theta_{\mathrm{s}} & 0 \\ (1-\theta_{\mathrm{s}})tc_x^{i_0} & (1-\theta_{\mathrm{s}})tc_y^{i_0} & (1-\theta_{\mathrm{s}})tc_z^{i_0} & 1 \end{bmatrix} \tag{9}$$

Applying $\mathbf{T}_{\mathrm{scale}}$ to PC_{PCA}, a new point cloud denoted by PC_{scale} will be obtained, which has the same scale with $PC_{\mathrm{R}}^{k_0}$, as shown in Fig. 4 (d).

3.4.3. Finetune the alignment by ICP

After the above steps, we shall obtain a point cloud (i.e., PC_{scale}) with quite decent alignment with the reference model, i.e., one at roughly the same location and with the same Z direction and identical scale. However, because of the coarse nature of the estimated initial camera pose, the PC_{scale} might still have deviation from $PC_{\text{R}}^{k_0}$ in terms of translation and orientation along X and Y axis.

Thus, iterative closest point is used to finetune the alignment, as demonstrated by the process from Fig. 4 (d) to (e). The ICP technique is an optimization algorithm that aims to minimize the error metric between two clouds of points by iteratively trying out different transformations. Suppose $\mathbf{T}_{\text{ICP}}^{j}$ is an arbitrary transformation matrix, then the process of ICP is mathematically described as follows:

$$\begin{cases} PC_{\text{ICP}}^{j} = Trans(PC_{\text{scale}}, \mathbf{T}_{\text{ICP}}^{j}) \\ \text{s.t.} \quad \min\limits_{\mathbf{T}_{\text{ICP}}^{j}} (rmse(PC_{\text{ICP}}^{j}, PC_{\text{R}}^{k_0})) \end{cases} \tag{10}$$

Where $Trans(PC, \mathbf{T})$ represents the resulting point cloud after applying transformation matrix $\mathbf{T}$ to PC. The meaning of $rmse(PC_1, PC_2)$ is the same as mentioned in section 3.3. In practice, it is computational inefficient to find the global minimum of $rmse(PC_{\text{ICP}}^{j}, PC_{\text{R}}^{k_0})$. Therefore, the iteration is terminated when certain criteria are met, e.g., maximum number of iterations or tolerance of RMSE. Suppose the optimal transformation matrix given by ICP is $\mathbf{T}_{\text{ICP}}^{j_0}$, then the final precisely aligned query point cloud can be obtained and denoted by $PC_{\text{ICP}}^{j_0}$.

3.5. Rectify camera pose with the point cloud transformation matrix

With a series of transformation matrices to register the query PC to the reference BIM, the initial camera poses can be rectified for robust indoor localization. The precise camera pose of i (i = 1,2,…,N_{C}) photo is calculated according to the following equation:

$$\mathbf{C}_{\text{prec}}^{i} = \mathbf{C}_{\text{Q}}^{i} \mathbf{T}_{\text{init}}^{i_0} \mathbf{T}_{\text{PCA}} \mathbf{T}_{\text{scale}} \mathbf{T}_{\text{ICP}}^{j_0}, \quad (i = 1, 2, ..., N_{\text{C}}) \tag{11}$$

Where the camera pose $\mathbf{C}_{\text{prec}}^{i}$ is presented by a form of homogeneous transformation matrix, including both description of orientation and location of the camera. Suppose $\mathbf{C}_{\text{prec}}^{i}$ is represented as follows:

$$\mathbf{C}_{\text{prec}}^{i} = \begin{bmatrix} rp_{11}^{i} & rp_{12}^{i} & rp_{13}^{i} & 0 \\ rp_{21}^{i} & rp_{22}^{i} & rp_{23}^{i} & 0 \\ rp_{31}^{i} & rp_{32}^{i} & rp_{33}^{i} & 0 \\ tp_{x}^{i} & tp_{y}^{i} & tp_{z}^{i} & 1 \end{bmatrix} \tag{12}$$

Then the estimated camera position is $\begin{bmatrix} tp_{x}^{i} & tp_{y}^{i} & tp_{z}^{i} \end{bmatrix}$. The camera posture/orientation can be characterized by a vector along the camera line of sight, which is computed as follows:

$$\mathbf{v}_{\text{prec}} = [0 \quad 0 \quad 1] \times \begin{bmatrix} rp_{11}^{i} & rp_{12}^{i} & rp_{13}^{i} \\ rp_{21}^{i} & rp_{22}^{i} & rp_{23}^{i} \\ rp_{31}^{i} & rp_{32}^{i} & rp_{33}^{i} \end{bmatrix} \tag{13}$$

Therefore, the camera direction vector $\mathbf{v}_{\text{prec}} = \begin{bmatrix} rp_{31}^{i} & rp_{32}^{i} & rp_{33}^{i} \end{bmatrix}$.

4. Experimental study

In order to validate the efficacy of the proposed approach, experimental studies were implemented in a campus building at the University of Tennessee, Knoxville (UTK). The BIM model of the building is a .rvt file with level of development (LOD) 350. The initial camera pose was estimated with the approach proposed by [17]. Both the coarse and precise registration algorithms were instantiated in MatLab. The used computing hardware is an OptiPlex 7080 computer with Intel(R) Core (TM) i7-10700 CPU and NVIDIA GeoForce RTX 2070 SUPER GPU.

4.1. The constructed reference database

Our experiment zone was set up at the third floor of the UTK campus building. Fig. 6 (a) shows the floor plan of the experiment zone, wherein we selected 11 spaces to construct the reference database. The “Section Box” function of Autodesk Revit was used to segment a separate model unit for each space, which was then exported as an individual .fbx file. Fig. 6 (b) shows snapshots of the 11 separated BIM model units. The model units of FBX format were imported to Blender for further processing, e.g., removing redundant elements. Finally, the mesh models were loaded into CloudCompare for “Mesh-to-Point” conversion, and metadata (e.g., XYZ range and boundaries) extraction.

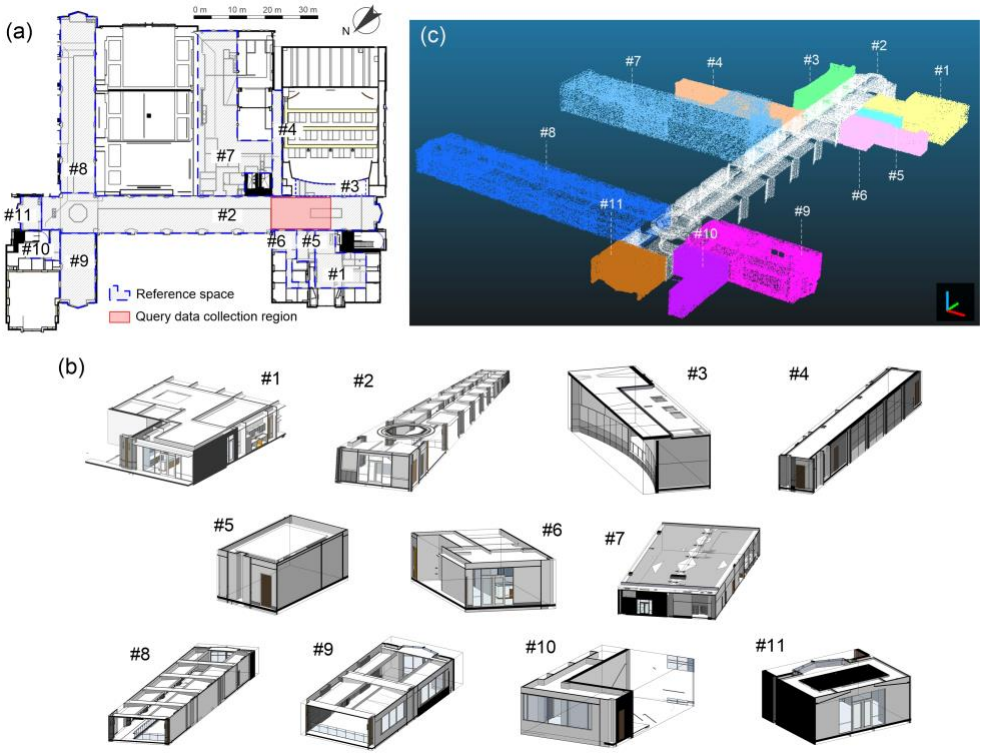

Fig. 6. (a) Floor plan and spaces used for constructing the reference database; (b) Separated models units of the referenced space; (c) The generated point clouds in the reference database.

Fig. 6 (c) shows the obtained reference point clouds, where points corresponding to different spaces have been highlighted by different colors. Metadata corresponding to all the 11 reference point clouds is listed in Table S1 in the Supplementary Material.

4.2. Query data collection schemes

As highlighted by the red rectangle in Fig. 6 (a), the query data was collected on a platform at the west end of space #2, covering an area of 112.8 m^2. We designated 30 data collection points on the platform, locations of which are presented in Fig. 7 (a). At each location, a video of its surrounding environment was recorded with a digital video (DV) camera (SONY HDR-CX760V). The DV camera was attached to a tripod to maintain its stability, and was designated to spin 360° around the central vertical axis of the tripod during the recording. Each video lasts for around 2~3 minutes, from which static image frames can be extracted for the production of photogrammetric PCs. There are many off-the-shelf commercial solutions (e.g., Agisoft Metashape, Pix4D) or open-source packages (e.g, WebODM) for photogrammetry applications. As a preliminary study aiming to testify the effectiveness of the proposed A2L approach, we select one of the most mature products in the market, Agisoft Metashape, for point cloud reconstruction from a bunch of images. For

practical deployment in future applications, Web application programming interface (API) of commercial or open-source photogrammetry software [39, 40] can be integrated as a service implemented on the cloud. For performance evaluation, the camera pose corresponding to a selected photograph from each data collection point was measured to serve as the ground-truth value. The camera orientations of the selected photographs are indicated by the arrow directions in Fig. 7 (a). In addition, the coarse camera poses of the selected photographs were estimated with the approach proposed by [17].

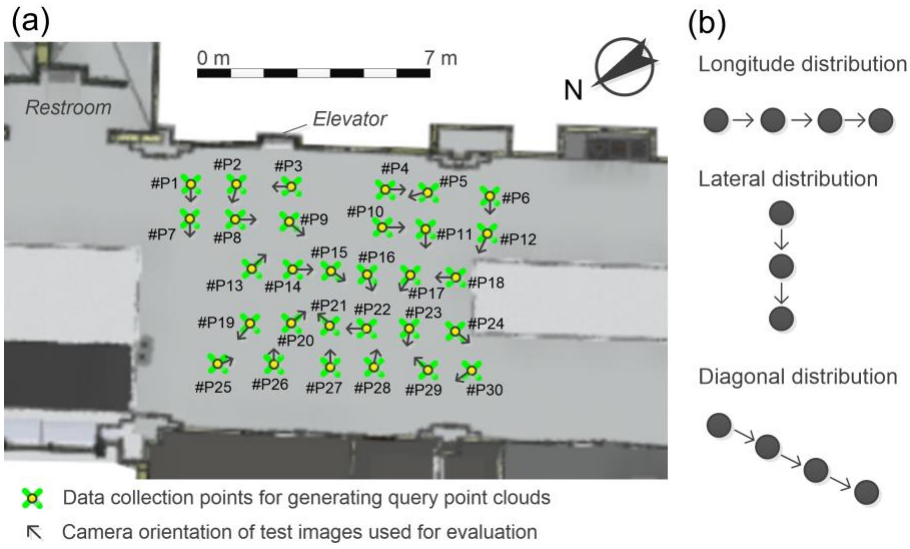

Fig. 7. (a) Distribution of the 30 designated data collection points; (b) Schematic diagram of the point distribution types when using different strategies.

Different strategies can be used to combine the images taken from different data points for generating the query PC. Three aspects of factors are considered, which are the number of locations (NoL), number of images per location (NoI), and distribution of locations (DoL). The NoL (e.g., NoL = 3, 4, 5, 6) reflects the quantity of data points from which the corresponding photographs are used to generate the point cloud, while NoI (e.g., NoI = 5, 10, 15, 20) is the number of used photographs from each selected data point. As shown in Fig. 7 (b), DoL indicates how the selected locations distribute, which includes three main types, i.e., longitude, lateral, and diagonal distribution. To determine the best strategy, different combinations of the three factors will be used to generate query PCs, and their registration and final localization performance will be investigated and compared. Table 1 lists all the combinations investigated in this study. For example, the "#1#2#3" means that photographs from data collection points #P1, #P2, and #P3, as indicated in Fig. 7, are used to generate corresponding PCs. Note that for each combination, different numbers of photographs can be used, i.e., NoI = 5, 10, 15, 20. The total number of locations is the 30 data collection points presented in Fig. 7, which, however, will not be fully made use of in certain

strategies due to insufficient number of points meeting the required distribution. For example, when the "DoL=Longitude" and "NoL=4" are used, the number of points in each row will not be divided evenly by four, leaving some points excluded, e.g., the #P5 and #P6 in the first row.

Table 1. Details of the investigated data collection strategies.

DoL	NoL	NoI			
		5	10	15	20
Longitude	3	#1~#3, #4~#6, #7~#9, #10~#12, #13~#15, #16~#18, #19~#21, #22~#24, #P25~#27, #28~#30			
	4	#1~#4, #9~#12, #14~#17, #20~#23, #25~#28			
	5	#1~#5, #8~#12, #13~#17, #20~#24, #25~#29			
	6	#1~#6, #7~#12, #13~#18, #19~#24, #25~#30			
Diagonal	3	#1#8#14, #2#9#15, #4#11#18, #7#19#26, #10#17#24, #13#20#27, #16#23#29			
	4	#1#7#13#20, #2#8#14#21, #3#9#15#22, #10#17#24#30, #12#18#23#28			
	5	#1#7#13#20#27, #2#9#15#16#23, #3#10#17#24#30, #8#14#21#22#28			
Lateral	3	#2#8#13, #3#9#14, #4#10#16, #5#11#17, #6#12#18, #15#21#27			
	4	#2#8#13#19, #3#9#14#20, #4#10#16#22, #5#11#17#23, #6#12#18#24			
	5	#2#8#13#19#25, #3#9#14#20#26, #4#10#16#22#28, #5#11#17#23#29, #6#12#18#24#30			

* Note: 1. The "DoL", "NoL" and "NoI" stands for distribution of locations, number of locations, and number of images per location, respectively;

2. The "#xx#xx#xx" stands for the combination of data collection points as depicted in Fig. 7 (a).

4.3. Performance evaluation

Four metrics were used to comprehensively evaluate the performance of the proposed approach, including localization error, orientation error, computation time, and pose recovery rate. The localization error is reflected by the Euclidean distance (m) between the predictive and the observed camera locations, and the orientation error, on the other hand, is measured by the angle deviation (°) between the predictive and the observed camera line of sight. The computation time includes both the time used to generate the query PC and the time of registration. When generating a point cloud, camera pose of some photos relative to the cloud may not be reconstructed due to unsuccessful alignment. In such case, the subsequent registration will not be able to recover their camera pose in the global reference system. To measure performance in this aspect, the pose recovery rate (PRR) was proposed and defined as the proportion of successfully recovered camera poses accounting for the total number of investigated poses.

A prerequisite for robust localization by PC2BIM registration is the correct selection of the reference PC. Among all the investigated test data, 26 out of 30 initial coarse camera poses estimated by [17] were correctly located within the range of reference space #2 (see Fig. 6 for the layout of the reference spaces). After majority voting, a correct reference model (i.e., space #2) has been selected for all the query point clouds generated from the strategies listed in Table 1. By trying out all the listed strategies (see Section 4.4), the combination of "NoL = 5", "NoI = 15", and "DoL = Lateral" is observed to perform best in trading of precision against time performance.

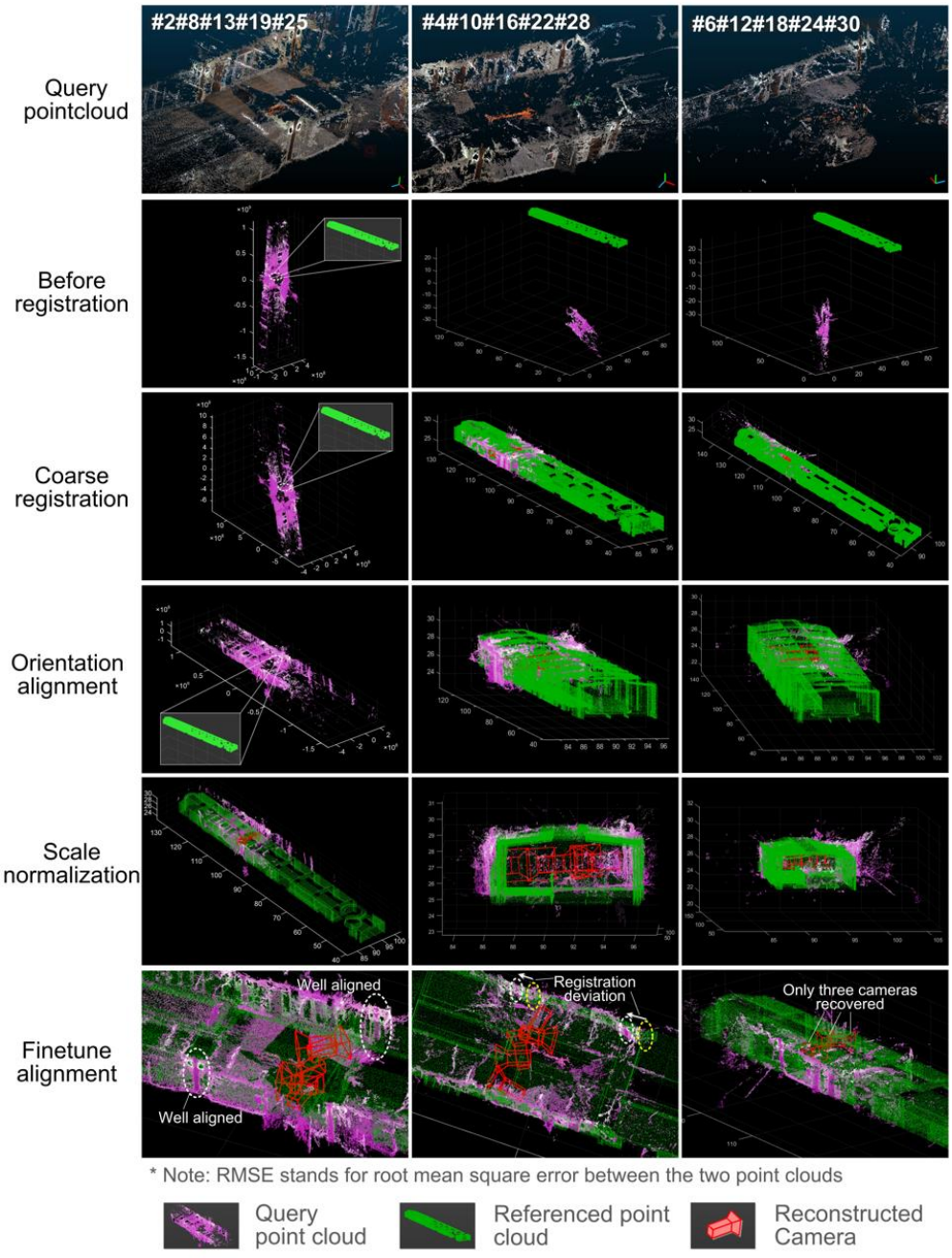

Fig. 8. Results of each registration step for query point clouds generated based on "NoL = 5", "NoI

= 15", and "DoL = Lateral" strategy, where the query and reference point clouds are highlighted by magenta and green, respectively.

Fig. 8 shows a step-by-step breakdown of the registration process for three example PCs generated under the "NoL = 5", "NoI = 15", and "DoL = Lateral" strategy. Despite the incompleteness and uncertainties in terms of scale, location and orientation, the query PCs have been successfully aligned with their reference counterpart after registration. To be more specific, the coarse registration puts the query PC into the right place; then, the orientation alignment rectified its direction so as to be in line with the reference PC; the scale normalization makes scale of the point cloud pairs consistent with each other; and finally, the transformation of the query point cloud is finetuned by ICP for precise and robust alignment. The success of the registration lays the foundation for subsequent camera pose estimation. Fig. 9 shows the camera poses estimated by our approach, where in Fig. 9 (a) the deviation with the ground-truth locations is visualized with dash lines, and in Fig. 9 (b) the localization and orientation errors for all the investigated camera poses are presented. It is observed that errors of the estimated camera poses for batch #3 are higher than those of others, which is mainly because of its relatively poor registration performance. As depicted in the last row of Fig. 8, observable deviation can be found for batch #3 (2nd column, consisted of point #4#10#16#22#28) as compared to the well aligned PC for batch #1 (1st column, consisted of point #2#8#13#19#25), which holds the highest localization precision among the five batches. For batch #5, two camera poses have not been successfully recovered by SfM, as also presented in the 3rd column of Fig. 8.

Fig. 9 indicates that 23 out of the 25 camera poses have been successfully recovered, with an average localization and orientation error of 1.07 m and 3.7°, respectively. As listed in Table 2, performance of the proposed A2L approach was compared with that of three BIM-enabled visual localization methods [16, 17, 20] proposed in recent years. BIM-PoseNet [16] was a deep neural network trained on synthetic images rendered by BIM and their corresponding rendering camera poses, which was later improved by [20] via exploiting the spatio-temporal BIM-rendered view sequences. In [17], a style transfer generative network was employed to further improve the localization precision, which, however, resulted in relatively large camera orientation errors. It was observed that the A2L approach significantly improved the precision of vision-based indoor localization enabled by BIM. However, as the proposed approach requires generating query point clouds from photographs, it takes more time for computation compared with other approaches.

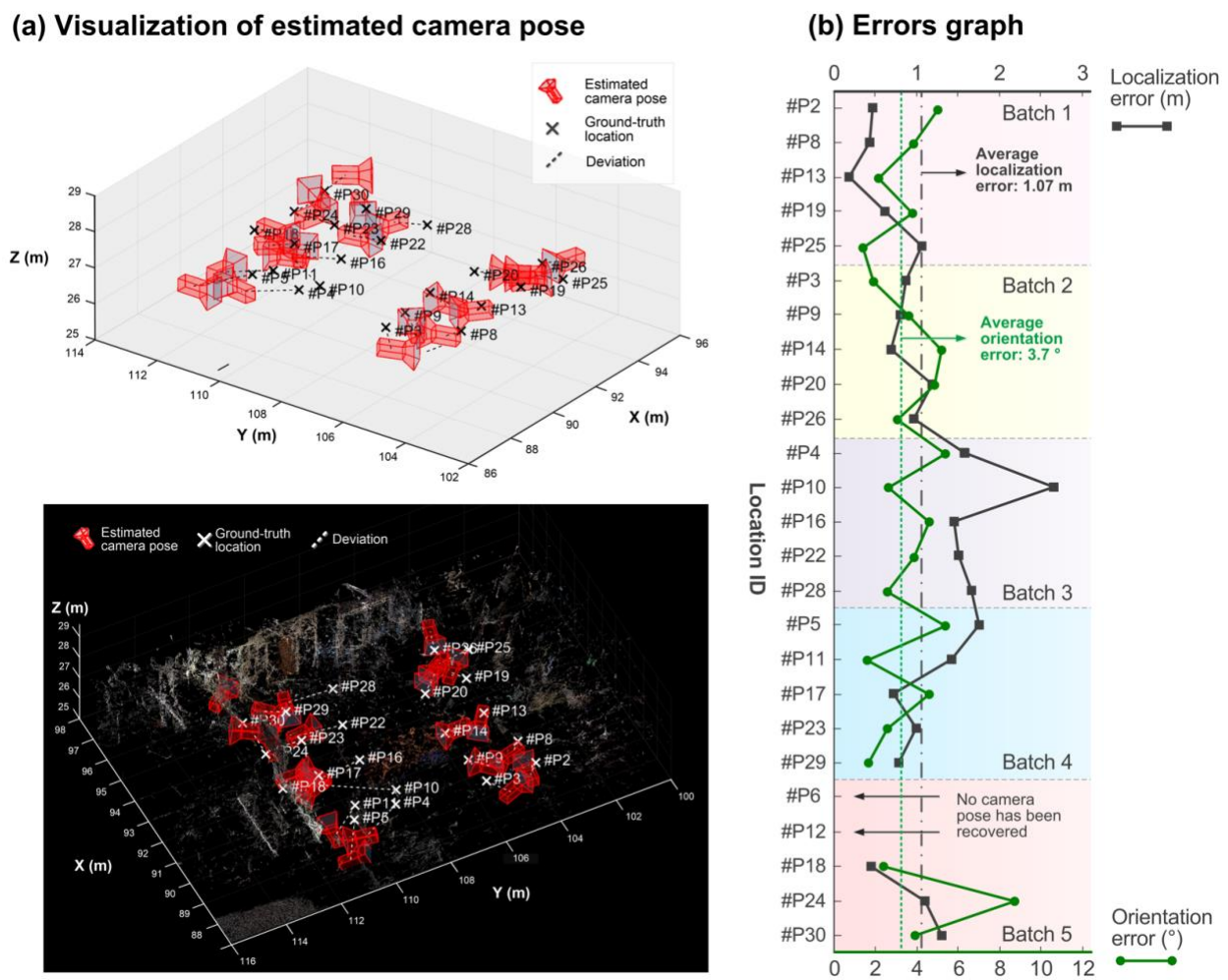

Fig. 9. Camera poses estimated by the proposed "align to locate" approach.

Table 2. Comparison with previous visual localization approaches based on BIM.

Approach	Localization error (m)	Orientation error (°)
BIM-PoseNet [16]	2.00	7.73
Recurrent BIM-PoseNet [20]	1.60	9.29
Chen et al. [17]	1.38	10.1
A2L (Our approach)	1.07	3.7

4.4. Sensitivity analysis

Sensitivity analysis is performed to determine how different data collection strategies will affect the camera pose estimation performance. The sensitivity analysis is based on the combinations of data collection points listed in Table 1. The average localization error, orientation error, computation time, and the pose recovery rate of all the investigated locations in a strategy are used to represent its corresponding performance. The first, middle and last column of Fig. 10 show

results of the four performance metrics for the lateral, longitude, and diagonal distribution, respectively. In each graph, the horizontal axis is the number of images (NoI) per location, and the number of locations (NoL) used in different strategies is depicted by different scattered curves.

Fig. 10 demonstrates a clear trend of improving pose estimation performance along with the increase of NoI. Both the localization and orientation errors, the two primary metrics to describe pose estimation precision, decrease as the NoI grows, although the decreasing level varies with the change of distribution directions (e.g., lateral, longitude, or diagonal). For the success rate of pose recovery, a larger NoI results in a higher PRR. The observed pattern can be explained by the basic rationale of point cloud generation based on SfM. A low NoI usually means less likelihood of overlap among the photographs, undermining the quality of the generated point cloud for effective registration or even making it difficult to reconstruct the corresponding camera poses (as indicated by the low PRR in Fig. 10 (g)~(i) when NoI=5, or the extreme cases in Fig. 10 (c) and (f)). With the growth of NoI, the improving query point clouds lead to better registration performance, and consequently higher precision is obtained. However, the positive effects of increasing NoI becomes marginal when it exceeds 15. In addition, a higher NoI also means more images to process, making the required computation time longer.

As for the number of locations for data collection, a higher NoL should presumably contribute to higher pose estimation precision. This has been well reflected in the metrics of localization error and PRR. In Fig. 10 (a) and (b), for example, if we neglect the condition of "NoI = 5" when the PRR is too low to allow objective evaluation, the scatter curves for higher NoL tend to distribute in lower position along the vertical axis, indicating smaller localization error. Fig. 10 (g) and (h) demonstrate an opposite pattern, with scatter curves representing greater NoL distributing at higher positions which indicate better chances of successful pose recovery. Comparatively, the effects of NoL on orientation errors are relatively difficult to identify, as the metrics for different NoL values all distribute closely at a low level (see Fig. 10 (d) and (e)).

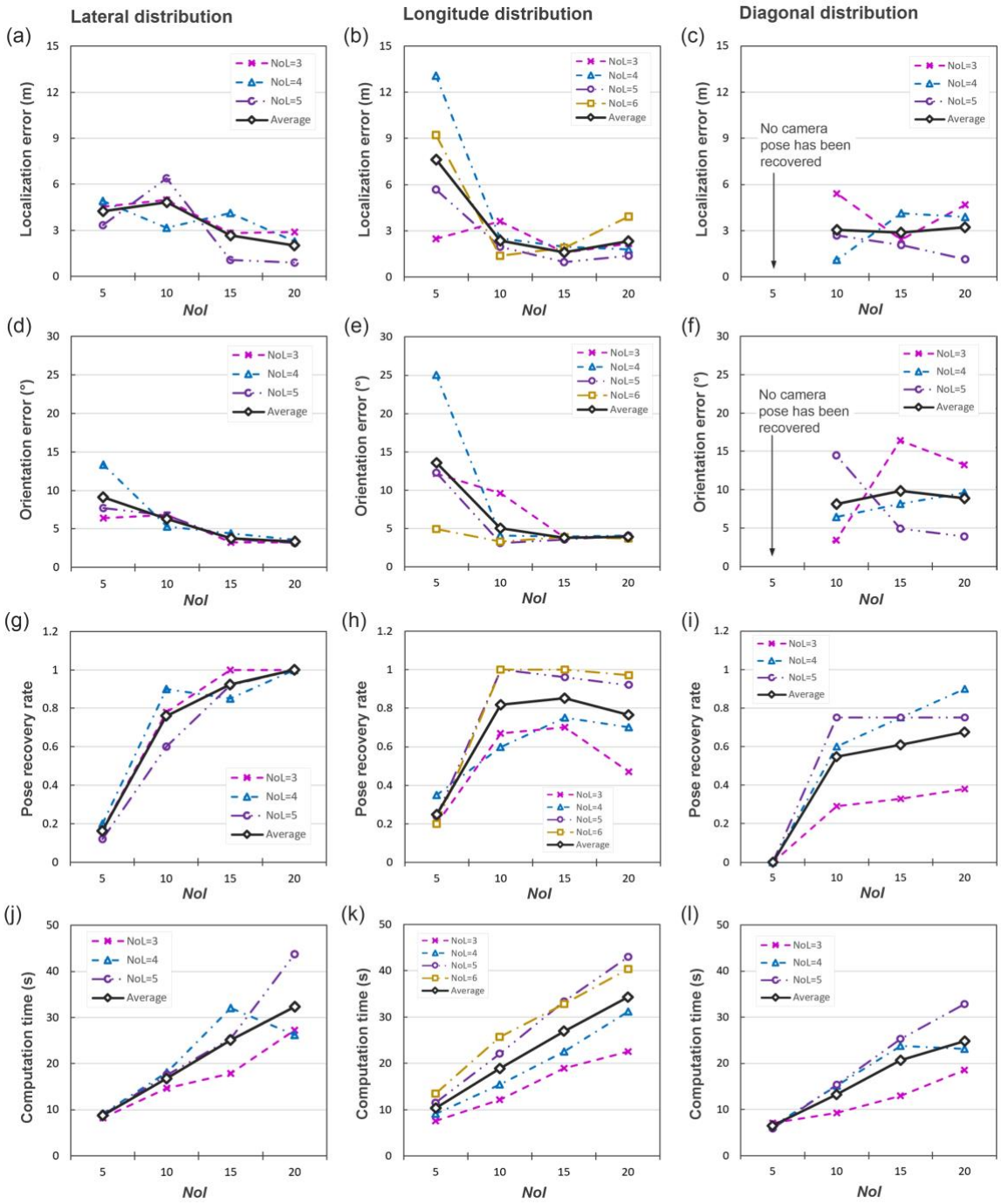

Fig. 10. Results of sensitivity analysis: (a)(d)(g)(j) performance for lateral distribution of locations; (b)(e)(h)(k) performance for longitude distribution of locations; (c)(f)(i)(l) performance for diagonal distribution of locations.

The last notable factor is DoL, the influence of which can be evaluated by horizontally comparing the average performance metrics across each row in Fig. 10. The overall performance, concerning

localization error, orientation error and pose recovery rate, gradually deteriorates as the DoL is changed from "lateral", to "longitude", and to "diagonal" distribution. As shown by Fig. 10 (g), (h) and (i), for example, the average PRR has already reached 80% when NoI equals 10 with the "lateral" distribution applied, while the highest average PRR for the "diagonal" distribution never exceeds 70%, whichever NoI is considered. The same trend can be clearly observed in the orientation error as well from Fig. 10 (d), (e) and (f), and is reaffirmed by a similar pattern revealed by the localization errors in Fig. 10 (a), (b) and (c).

To summarize, when using the proposed A2L approach for robust indoor camera pose estimation, it is recommended to collect data from laterally distributed locations, with around 15 photographs from each location. Although more data collection locations can lead to higher precision, it also requires longer computation time; thus, the NoL should be set in a reasonable range (e.g., NoL = 5) to balance between precision and efficiency.

5. Discussion

5.1. Advantages of the proposed approach

To tackle the challenge of indoor localization, this study proposes an "align-to-locate" approach for robust estimation of camera poses in built environments. The proposed approach outperformed the precision of previous methods, improving BIM-enabled visual localization to 1.07 m for localization error and 3.7° for orientation deviation. The high precision of the approach makes it suitable to various application scenarios such as facility inspection with robots and pedestrian navigation. Sensitivity analysis has been conducted to investigate the effects of different data collection strategies on pose estimation performance, indicating an evident trend of precision improvement with the increasing number of images from per locations.

Other than precision, another strength of the proposed approach lies in its compatibility with existing methods. Rather than replacing them, it leverages camera poses estimated by existing methods as initial parameters for coarse registration with the reference BIM model. In our experiments, the validation was implemented with the initial camera pose provided by [17]. However, other methods such as [16, 20] can also be applicable, as long as their estimated camera poses are corresponding to a selection of the photographs used to generate the query PC. Therefore, our approach serves as a general post-processing module, which can be seamlessly added to existing methods to rectify and finetune the initial camera poses for better reliability and robustness in practical applications.

5.2. Processing time and optimization

The proposed approach took about 25 s to process a batch of photographs when the "1.07 m and 3.7°" performance was achieved under the strategy of "NoL = 5", "NoI = 15", and "DoL = Lateral".

A large portion of the processing time (i.e., ~17 s) was used to generate the PC in an offline manner, which is relatively long. Therefore, optimization of the time performance is explored in this sub-section.

As SfM is based on the processing of the provided image batch (e.g., feature extraction, and correspondence detection), reducing image resolution might be able to shorten the required processing time. Experiments have been implemented with the "NoL = 5, NoI = 15, and DoL = Lateral" strategy to validate the hypothesis. Resolution of the original images is 1920×1080, which was downscaled successively to 1440×810, 960×540, and 480×270 for comparison. It was found that downsizing the original images by 0.25 to a resolution of 1440×810 reduced the required processing time for nearly a half, while can still maintain a decent quality of the generated PC. The computation time can be further reduced by continuing to downsize the images, which, however, would provide too few pixels to allow successful reconstruction, as have been discussed in [41]. Fig. 11 shows the trends with the PC generated from the "#2#8#13#19#25" batch as an example. Considering all five batches with the resolution of 1440×810, the SfM time performance is significantly improved to 7.84 s per batch, while the average localization and orientation errors remain at the original level of around 1.13 m and 4.03°. For batches of 960×540 and 480×270 resolution, because of the extremely low SfM reconstruction quality, no camera pose has been properly recovered.

The above results indicate that reasonably reducing the image size can contribute to the improvement of efficiency without impairing precision of the recovered camera poses. However, the level of downsizing should never exceed a certain range; otherwise, the SfM reconstruction would be jeopardized or even fail.

Fig. 11. Comparison of point clouds generated from images of different resolution (taking the "#2#8#13#19#25" batch as an example). Note that SfM reconstruction from images of 480×270 failed, and thus the corresponding point cloud does not exist.

5.3. Consideration for practical applications

As a proof of concept, the query PC in this study is generated offline by standalone SfM software

(i.e., Metashape), the integration of which into practical applications is an issue to consider. To address the concern, following use cases are proposed:

On the one hand, the query point clouds can still be generated offline by SfM, but on a cloud server. In this case, the Web API provided by commercial software (e.g., Agisoft Metashape [39], Pix4D) or open-source packages (e.g., WebODM [40]) can be seamlessly integrated with robots or any other devices requiring positioning services. As processing a batch of images for SfM can take up to a few seconds (see section 5.2), real-time implementation is not realistic. In most cases, such real-time localization is not necessary as well. Instead, a "stop-and-localize" solution can be used. To be more specific, the robots can take a bunch of indoor photos according to the recommended data collection strategy, and then upload them to the cloud for SfM, registration, and camera pose estimation. The A2L only needs to be implemented at the beginning for providing initial global coordinates, or be executed periodically for drift rectification. Thereafter or for the time windows in-between, tracking algorithms such as visual odometry and dead reckoning can be used to provide continuous information of the device' position.

On the other hand, the point cloud can also be generated continuously "on the go", which can either be done by visual SLAM [7] or newly introducing incremental SfM algorithms [8, 42] that allow real-time implementation. In this "on-the-go" solution, since the query point cloud is incrementally updated as the robots navigate through its surrounding environment, separate computation time for SfM is not required, making the algorithm more efficient. However, even though recent studies [43, 44] have demonstrated the sufficient accuracy of point clouds generated by such incremental approaches, their quality might still be different from those produced by offline tools, which consequently can lead to uncertainty in the registration with BIM. How the online generated photogrammetric point clouds might impact the camera pose estimation would be an interesting research topic worth investigation. As a preliminary study aiming primarily at developing and validating the A2L approach, we leave the topic for future research.

6. Conclusions

Visual indoor localization enabled by BIM is an active research field in recent years, owning to its merits of being infrastructure independent and free from pre-mapping. However, applicability of existing approaches in demanding scenarios is hindered by their relatively low precision. This study proposes an "align-to-locate (A2L)" approach that can rectify the coarse camera poses provided by existing approaches for robust indoor localization. The method achieved camera pose estimation by registering an as-is photogrammetric point cloud to a repository of reference BIM models via a series of operations such as coarse registration, orientation alignment, scale normalization, and alignment finetuning. Effectiveness of the A2L approach was demonstrated by an experimental study implemented at a campus building of the University of Tennessee, Knoxville. It achieved a precision of 1.07 m and 3.7° for localization and orientation error, respectively,

refreshing the state of the art of its kind. A sensitivity analysis was performed to understand the influence of different data collection strategies on localization performance, implying the superiority of the "lateral" strategy than the "diagonal" strategy. While more photographs from more data collection points may potentially lead to higher precision, it requires additional processing time. The A2L approach is compatible with existing methods to finetune their estimated camera poses for advanced applications such as robot navigation.

Future research is suggested to address the following limitations. The most notable one is the efficiency issue. Although for robotic applications, the time for data collection and point cloud reconstruction can be neglected, there is room to further optimize the required computation time for PC2BIM registration (~ 6 s). In this research, an off-the-shelf commercial solution, Agisoft Metashape, was used to produce the query point clouds offline. As different software/algorithms can generate point clouds of various quality, it would be interesting for future research to compare the performance of different SfM and SLAM solutions, and identify the best-performed one. Another limitation is one universally observed in vision-based localization, i.e., the adverse effect of uniform design and self-similarity in built environments. Such effects could impair the performance mainly by providing incorrect initial camera pose in the coarse registration stage. As a countermeasure, extra information (e.g., user input, data collected by other sensors) can be integrated to reduce ambiguities.

Declaration of competing interest

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

Acknowledgement

This research was supported by the U.S. National Science Foundation (NSF) through grant #1850008 and #2038967.

References

[1] ViewAR GmbH, Augmented Reality Indoor Navigation: Overview of Available Tracking Systems and Solutions, 2020. https://www.viewar.com/blog/augmented-reality-indoor-navigation-positioning/. (Accessed August 29 2021).

[2] Q. Zhu, Y. Shi, J. Du, Wayfinding Information Cognitive Load Classification Based on Functional Near-Infrared Spectroscopy, Journal of Computing in Civil Engineering 35(5) (2021) 04021016.

[3] K. Asadi, H. Ramshankar, M. Noghabaei, K. Han, Real-Time Image Localization and Registration with BIM Using Perspective Alignment for Indoor Monitoring of Construction, Journal of Computing in Civil Engineering 33(5) (2019) 04019031.

[4] D. Hu, H. Zhong, S. Li, J. Tan, Q. He, Segmenting areas of potential contamination for adaptive robotic disinfection in built environments, BUILD ENVIRON 184 (2020) 107226.

[5] B. Quintana, K. Vikhorev, A. Adán, Workplace occupant comfort monitoring with a multi-sensory and portable autonomous robot, BUILD ENVIRON 205 (2021) 108194.

[6] S. Winter, M. Tomko, M. Vasardani, K.F. Richter, K. Khoshelham, M. Kalantari, Infrastructure-Independent Indoor Localization and Navigation, ACM COMPUT SURV 52(3) (2019).
[7] Davison, Real-time simultaneous localisation and mapping with a single camera, Proceedings Ninth IEEE International Conference on Computer Vision, 2003, pp. 1403-1410.
[8] D. Nister, O. Naroditsky, J. Bergen, Visual odometry, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., 2004, pp. I-I.
[9] G. Lu, C. Kambhamettu, Image-based indoor localization system based on 3D SfM model, IS&T/SPIE Electronic Imaging, SPIE, 2014, p. 90250H.
[10] A. Rituerto, L. Puig, J.J. Guerrero, Visual slam with an omnidirectional camera, 2010 20th International Conference on Pattern Recognition, IEEE, 2010, pp. 348-351.
[11] C.M. Eastman, The use of computers instead of drawings in building design, AIA journal, 1975, pp. 46-50.
[12] J. Du, D. Zhao, R.R.A. Issa, N. Singh, BIM for Improved Project Communication Networks: Empirical Evidence from Email Logs, Journal of Computing in Civil Engineering 34(5) (2020) 04020027.
[13] R. Sacks, M. Girolami, I. Brilakis, Building Information Modelling, Artificial Intelligence and Construction Tech, Developments in the Built Environment (2020) 100011.
[14] J. Wang, X.Y. Wang, W.C. Shou, H.Y. Chong, J. Guo, Building information modeling-based integration of MEP layout designs and constructability, AUTOMAT CONSTR 61 (2016) 134-146.
[15] I. Ha, H. Kim, S. Park, H. Kim, Image retrieval using BIM and features from pretrained VGG network for indoor localization, BUILD ENVIRON 140 (2018) 23-31.
[16] D. Acharya, K. Khoshelham, S. Winter, BIM-PoseNet: Indoor camera localisation using a 3D indoor model and deep learning from synthetic images, ISPRS Journal of Photogrammetry and Remote Sensing 150 (2019) 245-258.
[17] J. Chen, S. Li, D. Liu, W. Lu, Indoor camera pose estimation via style-transfer 3D models, Computer-Aided Civil and Infrastructure Engineering (2021).
[18] Y. Feng, J. Wang, H. Fan, C. Gao, BIMIL: Automatic Generation of BIM-Based Indoor Localization User Interface for Emergency Response, in: C. Stephanidis, M. Antona, S. Ntoa (Eds.) HCI International 2020 – Late Breaking Posters, Springer International Publishing, Cham, 2020, pp. 184-192.
[19] A. Kendall, M. Grimes, R. Cipolla, PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization, 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2938-2946.
[20] D. Acharya, S. Singha Roy, K. Khoshelham, S. Winter, A Recurrent Deep Network for Estimating the Pose of Real Indoor Images from Synthetic Image Sequences, Sensors 20(19) (2020) 5492.
[21] H. Zhao, D. Acharya, M. Tomko, K. Khoshelham, Indoor LIDAR Relocalization Based on Deep Learning Using a 3D Model, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLIII-B1-2020 (2020) 541-547.
[22] R. Li, Y. Yuan, W. Zhang, Y. Yuan, Unified Vision-Based Methodology for Simultaneous Concrete Defect Detection and Geolocalization, Computer-Aided Civil and Infrastructure Engineering 33(7) (2018) 527-544.
[23] N. Ravi, P. Shankar, A. Frankel, A. Elgammal, L. Iftode, Indoor Localization Using Camera Phones, Seventh IEEE Workshop on Mobile Computing Systems & Applications (WMCSA'06 Supplement), 2006, p. 49.
[24] J.Z. Liang, N. Corso, E. Turner, A. Zakhor, Image-Based Positioning of Mobile Devices in Indoor Environments, in: J. Choi, G. Friedland (Eds.), Multimodal Location Estimation of Videos and Images, Springer International Publishing, Cham, 2015, pp. 85-99.

[25] F. Baek, I. Ha, H. Kim, Augmented reality system for facility management using image-based indoor localization, AUTOMAT CONSTR 99 (2019) 18-26.
[26] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14(2) (1992) 239-256.
[27] M. Bueno, F. Bosché, H. González-Jorge, J. Martínez-Sánchez, P. Arias, 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models, AUTOMAT CONSTR 89 (2018) 120-134.
[28] S. Bouaziz, A. Tagliasacchi, M. Pauly, Sparse Iterative Closest Point, Computer Graphics Forum 32(5) (2013) 113-123.
[29] J. Yang, H. Li, Y. Jia, Go-ICP: Solving 3D Registration Efficiently and Globally Optimally, 2013 IEEE International Conference on Computer Vision, 2013, pp. 1457-1464.
[30] D. Aiger, N.J. Mitra, D. Cohen-Or, 4-points congruent sets for robust pairwise surface registration, ACM Trans. Graph. 27(3) (2008) 1–10.
[31] J. Chen, Y. Cho, Point-to-point Comparison Method for Automated Scan-vs-BIM Deviation Detection, International Conference on Computing in Civil and Building Engineering, Tampere, Finland, 2018.
[32] C. Kim, H. Son, C. Kim, Fully automated registration of 3D data to a 3D CAD model for project progress monitoring, AUTOMAT CONSTR 35 (2013) 587-594.
[33] B. Mahmood, S. Han, D.-E. Lee, BIM-Based Registration and Localization of 3D Point Clouds of Indoor Scenes Using Geometric Features for Augmented Reality, Remote Sensing 12(14) (2020) 2302.
[34] M. Kopsida, I. Brilakis, Markerless BIM Registration for Mobile Augmented Reality Based Inspection, 2016, pp. 1-6.
[35] F. Xue, W. Lu, Z. Chen, C.J. Webster, From LiDAR point cloud towards digital twin city: Clustering city objects based on Gestalt principles, ISPRS Journal of Photogrammetry and Remote Sensing 167 (2020) 418-431.
[36] Y. Wu, J. Shang, F. Xue, RegARD: Symmetry-Based Coarse Registration of Smartphone's Colorful Point Clouds with CAD Drawings for Low-Cost Digital Twin Buildings, Remote Sensing 13(10) (2021) 1882.
[37] R.B. Rusu, Z.C. Marton, N. Blodow, M. Dolha, M. Beetz, Towards 3D Point cloud based object maps for household environments, Robotics and Autonomous Systems 56(11) (2008) 927-941.
[38] E. Turner, A. Zakhor, Watertight As-Built Architectural Floor Plans Generated from Laser Range Data, 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, 2012, pp. 316-323.
[39] Agisoft LLC., Online Processing Services, 2021. https://www.agisoft.com/buy/saas/. (Accessed Nov. 8 2021).
[40] OpenDroneMap, WebODM: Drone Mapping Software, 2021. https://www.opendronemap.org/webodm/. (Accessed Nov. 8 2021).
[41] J. O'Connor, Impact of image quality on SfM Photogrammetry: colour, compression and noise, Kingston University, 2018.
[42] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, P. Sayd, Generic and real-time structure from motion, British Machine Vision Conference 2007 (BMVC 2007), 2007.
[43] M.R.U. Saputra, A. Markham, N. Trigoni, Visual SLAM and structure from motion in dynamic environments: A survey, ACM Computing Surveys (CSUR) 51(2) (2018) 1-36.

[44] H. Strasdat, J.M. Montiel, A.J. Davison, Visual SLAM: why filter?, Image and Vision Computing 30(2) (2012) 65-77.