Automated Scan-vs-BIM Registration Using Columns Segmented by Deep Learning for Construction Progress Monitoring

"This paper is based on the MSc thesis of the first author."

Girma Zewdie Tsige[1], Bashar Saleem Abbas Alsadik [1], Sander Oude Elberink[1] and Maarten Bassier[2]

[1]Earth Observation Science Department, ITC Faculty, University of Twente, 7522 NH Enschede, The Netherlands
[2]Dept. of Civil Engineering, TC Construction- Geomatics KULeuven- Faculty of Engineering Technology Ghent, Belgium

(grmtsige@gmail.com ,b.s.a.alsadik@utwente.nl , s.j.oudeelberink@utwente.nl, maarten.bassier@kuleuven.be)

Keywords: Building Information Modelling, Building, Coarse Registration, Deep Learning, Point Cloud, Progress Monitoring.

Abstract

In construction automation applications, coarse registration between 3D Building Information Modelling (BIM) and the as-built point cloud is vital for the monitoring of construction progress. This can be achieved by extracting highly distinct geometric features in both datasets to speed up the correspondence search. However, the existing geometric feature-based coarse registration methods have limitations in the Architecture, Engineering, Construction & Facility Management (AEC/FM) context because building designs often contain a considerable self-similarity, symmetry, and lack of texture.

In this work, we propose an automatic coarse registration method that is motivated by the Random Sample Consensus (RANSAC) algorithm to estimate the transformation parameters that best align the as-built point cloud in the coordinate frame of the BIM model by matching the corresponding columns. The method is based on the extraction of columns from the as-built point cloud and the as-planned BIM model. For the point cloud data, fully automated column extraction techniques are used by applying deep learning, whereas the BIM model columns are extracted from the available semantic information. Experiments are carried out on real-life datasets from the building construction site to validate the proposed method. The results show that our proposed column-based registration method achieved an RMSE of 2 centimeters , and the cloud-to-cloud mean distance of 1.6cm ± 1.8cm after fine registration. The accuracy of the co-registration result shows that our proposed approach contributes to automating the registration between the as-built point cloud and the as-planned BIM model for construction progress monitoring.

1. Introduction

Monitoring work progress in building construction is crucial to enhance the performance of construction management: progress measurement (Bosché et al., 2015), material tracking (Scott & Assadi, 1999), and quality control (Arditi et al., 2015). Currently, with the development of Building Information Modelling (BIM) and reality capture technologies, such as laser scanning and Photogrammetry in the Architecture, Engineering, Construction &Facility Management (AEC/FM) industry, automating the construction progress monitoring through Scan-vs-BIM is becoming popular (Kim et al., 2020; Lin et al., 2015; Turkan et al., 2012). This requires acquiring the as-built 3D construction site status and comparing it with the design 3D BIM model to retrieve useful as-built information (F. Bosche et al., 2008; Frederic Bosche et al., 2009; Jacob-Loyola et al., 2021; Changwan Kim et al., 2013). This can be substantially simplified by 3D registration of the as-built point cloud with the as-planed BIM model so that they have a common coordinate frame (Bueno et al., 2018; J Chen & Cho, 2018). The 3D registration in the AEC/FM industry is currently implemented manually and the automated methods are still immature. The scope of this work is to automatically align the as-built point cloud from the building construction site to the as-planned BIM model for construction progress monitoring applications.

3D registration of the as-built point cloud with the as-designed BIM model can be carried out by two main consecutive steps: coarse registration to roughly align the datasets, followed by an automated fine registration process to optimally register them. The fine registration of the 3D dataset is an extensively studied problem with well-known computational solutions; those are primarily variants of the fundamental approach of minimizing the Euclidean distance between neighbouring points. The most popular approach is the Iterative Closest Point (ICP) algorithm (Besl & McKay, 1992) and its variants (Y. Chen & Medioni, 1992; Segal et al., 2009). On the other hand, Coarse registration of two 3D datasets can be performed when corresponding geometric features, such as points, lines, and planes, are identified and matched in the model and data points (Aiger et al., 2008; Bueno et al., 2018). Our work proposes a geometric feature-based coarse registration method since buildings are made up of evenly distributed prominent structural features, making it easier to solve the registration problem. Furthermore, our suggested feature extraction technique is robust to frequent problems in the AEC/FM context, such as a lack of texture, large self-similarities, and the presence of significant noise e.g., clutter from the construction site.

This work aims at extracting columns from an as-built point cloud from a construction site and matching them to corresponding columns in the as-designed BIM model for the co-registration of both datasets. To tackle the profound self-similarity problem in the extracted columns, we consider certain sets of columns with a distinct pattern. Columns were chosen as the main distinctive features for coarse registration because they are the most dominant building structural components usually constructed at early stages in the building construction process (Truong-Hong & Lindenbergh, 2022), and their centre of mass can be easily calculated from geometric data.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-1455-2025

The method employs deep learning to automatically extract columns from the as-built point cloud via semantic segmentation, and we then leverage the existing semantic information in the as-designed BIM to retrieve the corresponding columns. The approach is specifically developed to deal with the presence of clutters from the construction site, as well as self-similarity and design symmetries in building components. Our proposed approach enables us to automate the registration of the as-built point cloud and the as-planned BIM model, allowing us to track construction progress over time.

The rest of this work is organized as follows. The background and related work are provided in Section 2. In Section 3, the methodology is explained. Section 4 presents the test design and experiment outcomes. The conclusions are reported in Section 5.

2. RELATED WORK

As discussed above, coarse registration of the as-built point cloud with the as-designed BIM model can be performed when corresponding distinct geometric features are identified and matched in the model and data points. Kim et al. (2011) proposed an automated coarse registration method using principal component analysis (PCA) for the alignment of a 3D CAD model with the as-built point cloud for construction progress monitoring. They first converted the 3D CAD model into a point cloud representation. Their method resampled both point cloud and CAD models to have a uniform point resolution without extracting corresponding geometric features. This method works in a simple construction site used by Kim et al. (2011). Bosché (2012) proposed a plane-based semi-automated coarse registration approach to extract and match planes from both as-planned and as-built 3D models. He first converted the 3D BIM model into mesh for extracting the plane patches based on the normal vector information. His method requires a user to pick at least three non-parallel planes from both datasets and compute the 3D rigid transformation using a least-squares alignment approach. Bueno et al. (2018) proposed a 4-plane congruent set (4-PlCS) for the registration of as-built scanned data with the as-planned BIM mesh/model. The 4-PlCS developed by Bueno et al. (2018) starts with extracting all the planar patches from both the 3D BIM model and point cloud. For each '4-plane base' of the BIM model point cloud, the matching congruent '4-plane bases' are searched as a candidate set within point cloud plane patches using geometric descriptors, such as parallelism, orthogonality, and distance. Then, the 3D rigid transformation matrix is computed for each pair of the 4-PlCS, and the optimal one is selected based on the maximum number of scores from the rest of the congruent sets. Zhao et al. (2022) propose an improved plane detection approach based on region growing, which enhances the accuracy and efficiency of aligning various forms of spatial data, including point clouds, meshes, and CAD models. This unified framework aims to streamline the registration process across different data representations by leveraging the geometric properties of planar surfaces.

Processing of the construction site point clouds to extract the distinct geometric features for coarse registration remains mainly a manual affair with its challenges due to the variation in point density, clutters, self-similarity, and occlusions from the construction site (Anil et al., 2011). These distinct geometric features can be identified and extracted from point clouds using geometric feature descriptors, such as fast point feature histograms (Rusu et al., 2009) that specify the local geometry of point clouds, which are then utilized to construct the transformation matrix. For instance, Mahmood and Han (2019) used a fast point feature histogram as a feature descriptor and a random sample consensus (RANSAC) (Fischler & Bolles, 1981) to reject erroneous correspondences in aligning multiple scans in 3D point clouds. Adan & Huber (2011) used the Hough transform algorithm to extract the geometric features in the point cloud which assumes primitive parametric models can represent the objects in a point cloud scene. On the other hand, Machine learning techniques have also been used for semantic segmentation of point clouds to detect the components of the building, where computer models learn how to accomplish a task through supervised feature learning (Swetha Koppula et al., 2011). To learn a parametric model and classify objects based on the feature vector, supervised machine learning classifiers such as random forest or support vector machines can be employed (Weinmann et al., 2015). However, the above-mentioned methods are semi-automated or require intensive human interaction to some extent.

Currently, fully automated feature extraction methods using deep learning are becoming state of the art as it eliminates the requirement to extract and select feature descriptors and offers a more robust non-parametric classification model (Jingdao Chen et al., 2019). These methods can learn the pointwise features directly from the unstructured point cloud to capture various local and contextual characteristics of the point (Qi et al., 2017). Before putting the entire data into the deep learning model, the 3D point cloud is subsampled for efficiency (Girardeau-Montaut, 2015). Then, the point clouds from the construction site are semantically segmented to assign the label to one of several predefined classes, such as columns, walls, and floors that form the building’s main structure (Perez-Perez et al., 2021a).

3. METHODOLOGY

In this research, a registration algorithm is proposed that aligns the as-built point cloud with the as-planned BIM model for construction progress monitoring. The procedure takes as input a set of clustered corresponding columns from both datasets and outputs the registered dataset. So, the registration process consists of the following steps: (i) As-built point cloud column detection using deep learning, (ii) As-planed BIM model column detection, and (ii) Co-registration of both datasets. A detailed discussion of the research methodology is given below.

3.1 Point cloud column detection using deep learning

We used the KPConv (Kernel Point Fully Convolutional Network) point-based semantic segmentation model for extracting columns (Thomas et al., 2019). KPConv is a convolutional point-based semantic segmentation network that is motivated by an image-based convolution neural network (CNN), but instead of kernel pixels in the image convolution, the model utilizes a set of kernel points to describe the location where each kernel weight is applied. Once the class of the points is established, the observations belonging to the column object are filtered out and clustered. These clustered columns are the input of the presented registration algorithm to align the as-built point cloud (source datapoint) in the reference frame of the as-designed BIM model (target datapoint).

The datasets for training the deep learning model are collected from various sources: the construction site point cloud acquired by Terrestrial laser scanner (TLS), the publicly available published Raamac Lab dataset (Perez-Perez et al., 2021b) the S3DIS datasets (Armeni et al., 2017) and synthetic point cloud sampled from the BIM model exported from the Revit software

(BIM model structural projects) by adding 6 mm standard deviation Gaussian noise to the sampled synthetic point clouds to simulate the real behaviour of the laser scanner. The training data was labeled into five (5) interest classes: Column, Wall, Beam, Slab, and Clutter. Prior to inputting into the selected deep learning network for training and validating, the collected training data was preprocessed using various techniques, such as outlier removal, down sampling, additional attribute extraction, and labelling training data. The processed training dataset was split into a training, validation, and test using the 70:20:10 ratio from the original dataset. The model was trained using the training dataset and evaluated using the validation dataset. A description of the classes and data used for training, validation, and testing are shown in Table 1.

The KPConv model was trained from scratch on our prepared training dataset. During the training phase, the model hyperparameters, such as the learning rate, the number of epochs, the optimizer, and the regularization parameters, were adjusted to achieve the best possible results. We used workstation computers with high processing and memory capacity for training, validation, and inference.

Table 1: Description of the number of points per class used for training and testing the deep learning architecture

Data type	# Column points	# Wall points	# Beam points	#Slab points	# Clutter points	Total
Training	6.5M	14.4M	7.7M	40.1M	0.4M	69.1M
Validation	1.9M	4.1M	2.2M	11.5M	0.2M	19.9M
Testing	0.9M	2.1M	1.1M	5.7M	0.1M	9.9M

The trained model was evaluated on the test dataset acquired using a terrestrial Laser Scanner (TLS) from the building construction site which is located at the University of Twente (UT) campus in Enschede and has achieved 73% accuracy in overall classes using the mean intersection over union(mIoU) and column segmentation accuracy of 69%. The test result of the trained model is summarized in Table 2 in terms of the confusion matrix of intersection over union (IoU). The slab category achieved the highest accuracy, followed by the beam and wall class.

As seen in Table 2, the confusion matrix shows that there is some confusion between classes in the classification result. It can be observed that about 18% of wall class points were misclassified as column classes. The confusion between column and wall classes could mainly be caused due to the class similarity as both classes are vertical elements and possess similar normal vector components.

Table 2: Confusion Matrix of IoU metrics for semantic segmentation results from the KPConv model

	Clutter	Column	Beam	Slab	Wall
Clutter	0.31	0.10	0.01	0.08	0.48
Column	0.00	0.69	0.05	0.00	0.00
Beam	0.00	0.04	0.89	0.01	0.00
Slab	0.00	0.00	0.00	0.98	0.00
Wall	0.00	0.18	0.01	0.03	0.76

The main reason behind the very low prediction accuracy of the clutter class might be due to the limited quantity of training data for the clutter class (as seen in Table 1). The construction site point cloud was segmented using the trained model, and the semantic segmentation results are visualized in Figure 1.

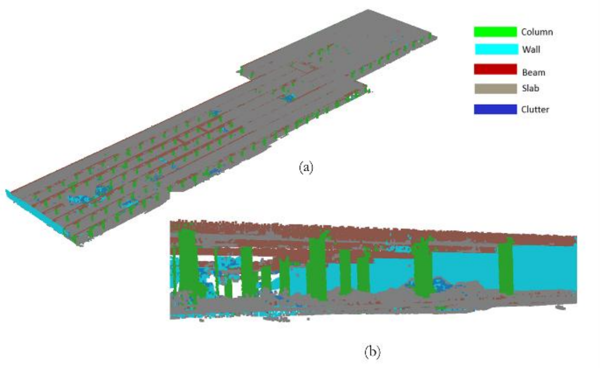

Figure 1: Results obtained from semantic segmentation of the building construction site TLS point cloud ground floor: a) Outside view, b) Inside view

Then, only the points belonging to the columns are filtered out among the other classes as illustrated in Figure 2 (left) for later use to estimate the transformation parameters. Then, the misclassified points that are a certain distance further from the column cluster points are removed using the outlier removal techniques. These misclassified points may contribute to the deviation in the computed geometric centroid of the columns and slabs in the subsequent stage, affecting the registration accuracy of the established approach as discussed in section 3.4.

3.2 BIM model column detection

The BIM model in IFC format was imported into the Revit software, and columns were retrieved from the available semantic information in the BIM. The columns were filtered from the structural section of the BIM model and exported as a mesh in STL format. Then, we sampled uniform resolution point clouds from the BIM column mesh. Figure 2 (right) illustrates the exported ground floor columns of the BIM model from the Revit Software in a point cloud format.

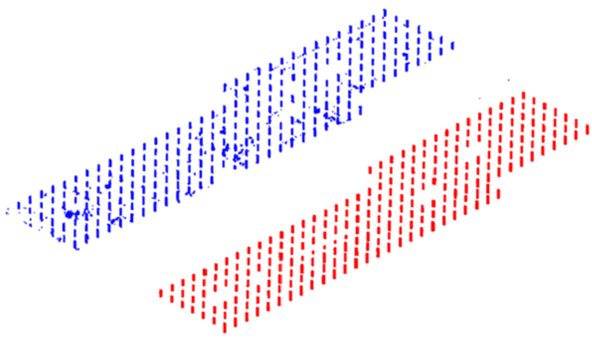

Figure 2: The detected ground floor columns (point cloud format): from the as-built point cloud using KPConv semantic segmentation model (left), from the as-planned BIM model

3.3 Column centroid computation result

The extracted columns from both datasets were subjected to the clustering algorithm to get the column instances of each dataset. We obtained 252 and 256 column clusters for the BIM, and TLS

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-1455-2025 | © Author(s) 2025. CC BY 4.0 License.

datasets, respectively. Once the column clusters were obtained, we computed the 3D geometric centroid of each cluster of the detected columns to use as input for our developed registration algorithm. We used the ground floor columns for both datasets. The computed centroid points for each dataset are shown in Figure 3.

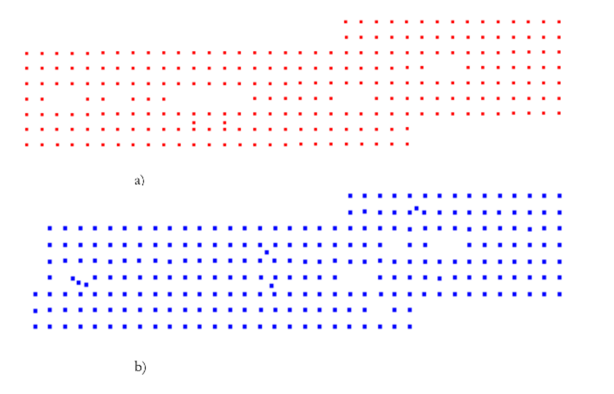

Figure 3: The centroids of the detected columns: a) BIM model, b) TLS Point cloud

3.4 Proposed registration method

We propose an automated coarse registration approach which is Motivated by the RANSAC algorithm (see Figure 4). The motivation behind this approach is to apply a trial-and-error co-registration approach that is robust to the outliers in the detected columns of the as-built point cloud from the semantic segmentation. The outliers should be detected and removed from the process of estimating transformation parameters. The key idea is to find the best partition of points in the inlier set and outlier and estimate the transformation model from the inlier set. The method assumes that both the model and point cloud columns have previously been detected. The rigid 3D transformation can be made along three orthogonal axes and is characterized by six unknown 3D similarity transformation parameters, i.e., three rotations, and three translation parameters, which are to be estimated. The 3D rotation matrix is composed using Euler angles (KUIPERS, 1999).

The building construction context has some unique benefits that can be taken advantage of during the initial estimation of transformation parameters, but it also has some unique limits that must be addressed (Bosché, 2012). The advantage is that the as-built point clouds are reconstructed with the vertical axis (Z) orthogonal to the ground this axis often correlates to the design 3D (BIM) model's vertical (Z) axis. So, based on this argument, our developed method assumes that the rotation mainly occurs along the vertical direction, and the rotation along the horizontal (X, Y) axis is assumed to be negligible given that the dual-axis compensator is enabled on the scanner. On the other hand, construction site as-built data are usually taken in noisy areas with a significant amount of clutters that are not part of the structure under target. These components cause occlusion that creates gaps in the as-built data acquired from the scene of interest. Furthermore, the TLS point cloud columns segmented from the deep learning semantic segmentation model, as illustrated in Figure 2, contain some outliers (misclassified points). Those points possibly contribute to the deviation in the computed geometric centroid of the column clusters and slabs, which on the other hand, affects the registration accuracy of the developed method in the later stage. Consequently, our proposed registration approach in Figure 4 estimates the transformation parameters in two steps: first estimate 2D transformation parameters, then followed by the 3D registration.

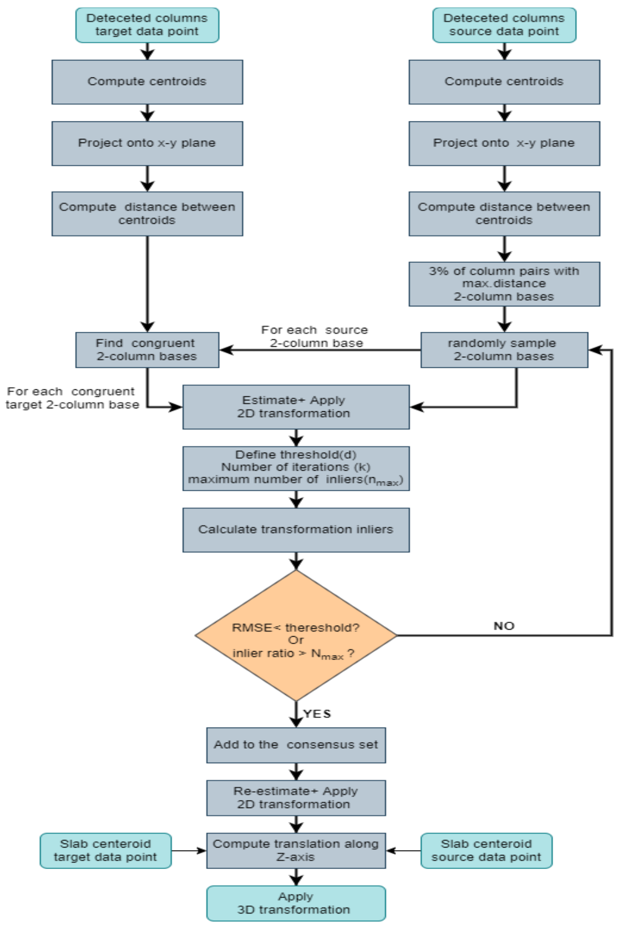

Figure 4: Workflow of the proposed coarse registration algorithm

A. Problem Setting

Given the detected columns of the BIM model (target data points) and as-built data (source data points), our approach starts with computing the 3D centroids of the detected columns for both data points and then projects it to the XY plane. Then, the method leverages the canter points as a basis for the correspondence search for estimating the transformation parameters. The method first determines the 2D rigid body transformation parameters in the XY Euclidian space, then is followed by estimating 3D rigid body transformation parameters. If (x,y) and (u,v) represent the cartesian coordinates of the projected source data point and target data point column centroids in XY plane respectively, the 2D rigid body transform is represented by:

$$\begin{bmatrix} u \\ v \end{bmatrix} = [t_x \quad t_y]^T + \mathrm{R}\begin{bmatrix} x \\ y \end{bmatrix} \qquad (1)$$

Where R is the rotation part of the transform

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

and t_x and t_y are the translations along the X and Y axes respectively.

At least a pair of matching column centroids from each data point is required to solve the 2D rigid body transformation equation above.

B. Implementation of the proposed algorithm

The proposed modified RANSAC algorithm randomly picks a pair of column centroids (2-column base) from the source data point and searches for all possible corresponding congruent pairs of column centroids from the target data points to compute the 2D rigid body similarity transformation. The congruent bases from the target data points are determined based on the distance between the randomly selected source column centroid pair, i.e., the distance between the target and source column bases should be equal within a certain distance tolerance δ. This trial-and-error approach to get the optimal transformation continues until the best transformation is found.

Given the symmetry and repetitiveness found in the built environment, it is likely that many congruent sets do not actually lead to the correct transformations. So, to tackle this problem, first, we consider only the top 3% of source column bases with the maximum 2D distance between column centroids. This significantly reduces the number of corresponding congruent target data point candidates required for estimating the optimal transformation parameters. Note that the method considers the entire target data point column centroids for the selection of all possible matching candidate sets based on the congruent distance. Then, the likelihood of the transformation derived from each pair of column bases to be optimal must be assessed by evaluating whether the rest of the dataset supports it. All the transformed source data point columns within a certain threshold distance (e.g., 30cm) from the corresponding closest target column centroids are selected as supports (inliers). The optimal transformation is selected based on the maximum number of inliers.

Once the 2D transformation parameters are obtained using our proposed modified RNSAC algorithm, then the remaining unknown translation vector along the Z-direction t_z is estimated using the centre of mass of the slabs detected from both datasets. We extracted the points belonging to the ground and the first-floor slabs of the source data points from the semantic segmentation result followed by postprocessing to remove the outliers. On the other hand, the corresponding slabs of the as-planned BIM model are extracted from the existing semantic information in the BIM model. The height between the ground and first floor for both datasets is determined using the centre of mass of slabs. Then, the translation t_z along Z-direction is calculated by subtracting the floor height of the source data point from the target data point.

Finally, the as-built point cloud is transformed based on the estimated 3D transformation parameters. The acquired result can be considered as an initial alignment (coarse registration). Then, we apply fine registration using ICP as a refinement registration on top of our developed coarse registration to get a more accurate result.

4. EXPERIMENTS

We tested our proposed method using the as-built point cloud and the as-planned BIM model of the building construction site which is located at the University of Twente (UT) campus in Enschede. It is a one-story building with a length of about 220m and a width of about 50m. The as-built point cloud is acquired using the terrestrial laser scanner (TLS). The column detection procedure is already discussed in section 3.1 and section 3.2. We tested the proposed registration method on our presented dataset for the following different scenarios:

- Incompleteness in the source data points: usually, the as-built point cloud from the construction site might contain only a part of the building, and only a limited number of columns could be obtained.
- Presence of symmetry and self-similarities in the source data points due to significant symmetry and self-similarity (e.g. the uniform column pattern) in our test dataset, it was expected that several transformations would have very high inliers, although only one of them is the correct one. This often poses a challenge in the process of estimating the correct 3D transformation parameters.

Three experimental setups were done on the given case study dataset to test the robustness of our proposed method for the above scenarios. These are: i) using the entire source data points, ii) using the left outlined section of source data points (Figure 5a), and iii) using the top outlined part of the source data points (Figure 5b). We explained each case in detail below. Our detected reference/target data points illustrated in Figure 3a are used entirely in all the test cases.

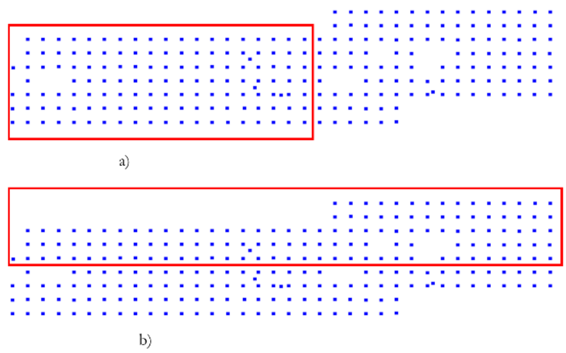

Figure 5: Experimental dataset configuration for symmetry test using a different section of the source data points as outlined by the rectangular red line: a) left section, b) top section

For the transformation parameters computation, we used “Intel(R) Core (TM) i5-4200M CPU @ 2.50GHz” with RAM storage of 16GB laptop. Note that the run times presented in Table 3 for various scenarios exclude the segmentation time which is done separately using workstations as discussed in section 3.1. In all scenarios, our proposed method considers only 3% of the pair of source data points with the maximum distance between them for the registration with the entire target data points, as discussed in section 3.4. For all test scenarios, we set the following parameter for our proposed algorithm:

- Maximum main iteration= 100
- Inlier ratio= 90%
- threshold distance= 30cm

The proposed coarse-registration approach was evaluated based on two criteria: -`

1. **Registration speed**: Computational time required to perform the registration. And the number of iterations needed until the optimal transformation parameters selecting process reaches the termination criterion
2. **Registration quality**: The quality of our proposed registration method is assessed based on the number

of inliers and the root mean square error (RMSE) both after coarse and fine registration.

- **The number of inliers:** the number of correctly placed point cloud columns that support the optimal transformation (within a threshold distance).
- **RMSE:** the root mean square error of the distances between the column Centroid inliers of the point cloud and the BIM model is computed as seen in equation 2, where N is the number of inliers and d_i is the distance between inliers after transformation.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i)^2} \qquad (2)$$

- **Cloud-to-cloud distance:** After registering the point cloud columns to the reference BIM model columns, we compute the distance between the corresponding points using the Cloud-to-Cloud distance computation tool in CloudCompare. Note that we use the corresponding inliers (detected column point cloud) for the cloud-to-cloud distance computation (but not the column centroids).

Case I: Registration using the entire source data points

We inputted entire source and target data points into our developed registration algorithm and the results are summarized in Table 3. The total number of the detected as-built point cloud column centroids is 256, out of which about 13 (5%) centroids are outliers (non-columns) as discussed in section 3.3. Note that we estimated the outlier ratio by visual inspection since the inlier column centroids are often aligned linearly in X and Y directions.

Case II: Registration using a left section of source data points (see Figure 5**a)**

The selected section of source data points contains 139 column centroids. The outlier ratio in the selected dataset is about 3%. The selected source data points and the entire target data points were loaded into the registration algorithm, and the results are summarized in Table 3. In this case, when compared to the case for the entire point cloud column registration result, the average iteration and computational time needed to get the best transformation is higher. This might have happened because of many congruent pairs of column centroids in the target data points due to the repetitive column pattern for the selected 3% candidate bases of column centroids in the source data points.

Case III: Coarse registration using the top part of source data points (see Figure 5b).

The selected source data point contains 134 column centroids with an outlier ratio of less than 3%. During the algorithm's execution with all the given parameters, the selected source data points were not correctly aligned due to the symmetry and self-similarity (a repetitive pattern of columns) in the dataset. For this limitation, we suggest that the user should use a more strict parameter setting for the inlier ratio at the cost of computational time. Since, in our case, the source data point contains about a 3% outlier ratio, we achieved the correct transformation by increasing the inlier ratio from 90% to 95%, keeping the other parameters fixed. The results are summarized in Table 3.

Table 3: Summary of the transformation results for the three different cases of source data points

Test cases	# Inliers	# Iteration	Registration run time (sec)	Modified RANSAC RMSE (m)	ICP RMSE (m)
Case I	238	57	52	0.06	0.02
Case II	126	128	85	0.04	0.02
Case III	130	57	63	0.04	0.02

As observed from Table 3, our developed registration approach has correctly aligned source data points to the reference target data points in all scenarios with an average RMSE of 5cm in 2D (XY plane). The performance was also evaluated in terms of computational speed, and the algorithm requires, on average, 50 iterations within 55 sec to achieve the optimal result for both case I and case III. This high speed in computation is achieved by considering congruent distance descriptors and only the top 3% of pairs of source data points for correspondence search in target data points. This would take many hundred iterations if the basic RANSAC algorithm were applied by considering entire source data points for correspondence search.

As discussed in section B, we computed the vertical translation using the center of mass of the slabs for 3D rigid body transformation. Figure 6(a) illustrates the overlaying of the point cloud columns to the corresponding BIM model columns after 3D coarse registration using our proposed coarse registration. For the fine registration equivalent, we applied ICP on top of the coarse registration for a more accurate result and achieved an average RMSE result of 2cm in 2D (XY plane) as summarized in Table 3. Figure 6(b) illustrates the overlaying of the point cloud columns to the corresponding BIM model columns after transformation using ICP.

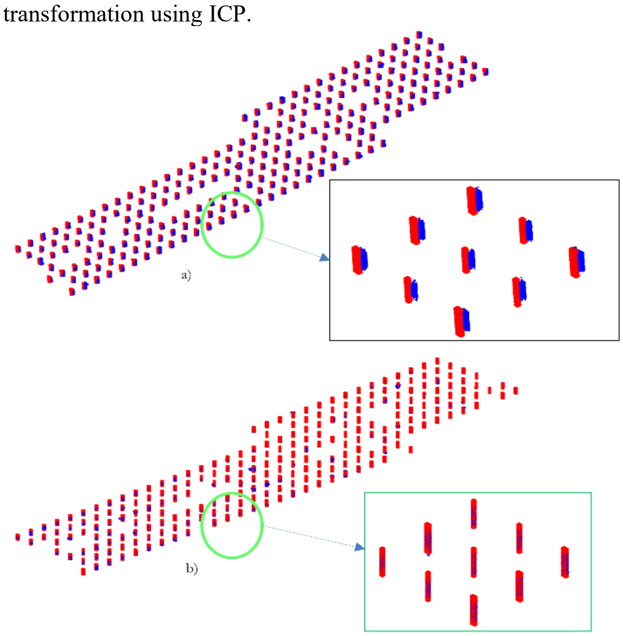

Figure 6: Overlay of the columns of source and target datasets; a) after coarse registration using our proposed method (a), after fine registration using ICP (b).

We have also computed the cloud-to-cloud distance between the reference BIM model and the transformed point cloud first-floor slab, as illustrated in Figure 7. We have achieved the average mean distance of 1.6cm ± 1.8cm after registration using ICP. The red points are the outliers in the detected slabs that cause the discrepancy in the cloud-to-cloud distance. This might be caused by the outliers in the detected slabs due to clutters and also the semantic segmentation model detection accuracy.

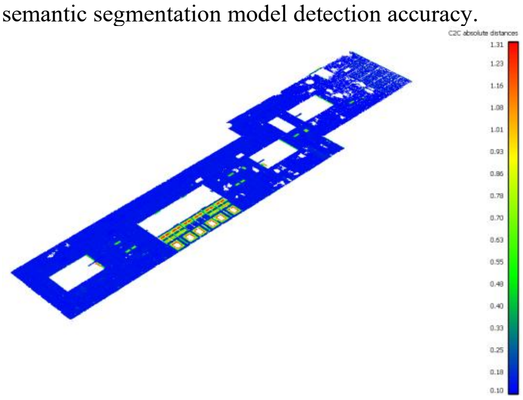

Figure 7: Cloud-to-cloud distance between the reference BIM model mesh to the corresponding as-built point cloud slabs (in m) after registration using ICP. Colours are relative to the minimum and maximum distance values.

We have applied the 3D transformation obtained using our proposed registration on the entire TLS point cloud and illustrated the result in Figure 8.

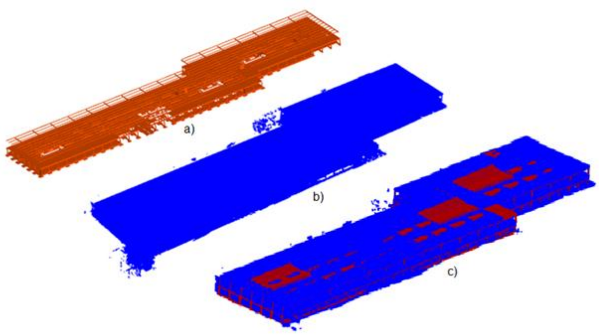

Figure 8: Overview of the result of the presented automated Scan-vs-BIM co-registration procedure: as-planned BIM model mesh (a), as-built point cloud (b), and the co-registration result of both datasets (c).

5. Conclusion and future work

This research proposed a column-based automatic coarse registration method for the alignment of the as-built point cloud in the reference frame of the 3D as-planned BIM model for construction progress monitoring. The developed method estimates the 3D rigid transformation parameters by automatically matching the corresponding columns in the as-built point cloud and as-planned BIM model. The point cloud columns are extracted by using the point-based KPConv deep learning model, whereas the corresponding columns from the BIM model are retrieved from the BIM's structural family using Revit software. Both the detected point cloud and BIM model columns were clustered, and the centroids were computed and loaded into the proposed coarse registration approach. The corresponding column centroids are then automatically matched to estimate the 2D rigid transformation parameter. Then, we computed the vertical translation using the center of mass of slabs for the 3D coarse registration.

We applied ICP as the fine registration equivalent on top of the coarse registration to achieve accurate registration for construction progress monitoring. Our experimentation result shows that the proposed alignment approach has reliably detected the best portion of the inlier columns that supports the estimated optimal transformation parameters. The results show that our proposed column-based registration method achieved an RMSE of 2 centimeters within the computational time of 55 seconds, and the cloud-to-cloud mean distance of 1.6cm ± 1.8cm in 3D after fine registration using ICP. As a result, we conclude that our proposed method contributes to automating the registration between the as-built point cloud and the as-planned BIM model to monitor the construction progress (by comparing the TLS and UAV as-built point cloud with the as-planned BIM model).

However, our proposed method is tested on the as-built point clouds where the columns have not been covered by walls. However, the column detection using semantic segmentation fails if the columns are embedded in the walls. Therefore, in such a case, other surface feature extraction and matching approaches can provide a better automated coarse registration result for progress monitoring.

6. References

Adan, A., & Huber, D. (2011). 3D reconstruction of interior wall surfaces under occlusion and clutter. *Proceedings - 2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2011*, 275–281. https://doi.org/10.1109/3DIMPVT.2011.42

Aiger, D., Mitra, N. J., & Cohen-Or, D. (2008). 4-Points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics*, *27*(3). https://doi.org/10.1145/1360612.1360684

Anil, E. B., Akinci, B., & Huber, D. (2011). Representation requirements of as-is building information models generated from laser scanned point cloud data. *Proceedings of the 28th International Symposium on Automation and Robotics in Construction, ISARC 2011*, *June*, 355–360. https://doi.org/10.22260/isarc2011/0063

Arditi, D., Arditi, D., & Gunaydin, H. M. (2015). Total quality management in the construction process Total quality management in the construction process. *International Journal of Project Management*, *7863*(AUGUST 1997), 235–243.

Armeni, I., Sax, S., Zamir, A. R., & Savarese, S. (2017). *Joint 2D-3D-Semantic Data for Indoor Scene Understanding*. http://arxiv.org/abs/1702.01105

Besl, P. J., & McKay, N. D. (1992). Method for registration of 3-D shapes. In P. S. Schenker (Ed.), *Sensor fusion IV: control paradigms and data structures* (pp. 586–606). https://doi.org/10.1117/12.57955

Bosché, F. (2012). Plane-based registration of construction laser scans with 3D/4D building models. *Advanced Engineering Informatics*, *26*(1), 90–102. https://doi.org/10.1016/j.aei.2011.08.009

Bosché, F., Ahmed, M., Turkan, Y., Haas, C. T., & Haas, R. (2015). The value of integrating Scan-to-BIM and Scan-vs-BIM techniques for construction monitoring using laser scanning and BIM: The case of cylindrical MEP components. *Automation in Construction*, *49*, 201–213. https://doi.org/10.1016/j.autcon.2014.05.014

Bosche, F., Haas, C. T., & Murray, P. (2008). Performance of automated project progress tracking with 3D data fusion. *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, *1*, 349–358.

Bosche, Frederic, Haas, C. T., & Akinci, B. (2009). Automated Recognition of 3D CAD Objects in Site Laser Scans for Project 3D Status Visualization and Performance Control. *Journal of Computing in Civil Engineering*, *23*(6), 311–318. https://doi.org/10.1061/(asce)0887-3801(2009)23:6(311)

Bueno, M., Bosché, F., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2018). 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Automation in Construction*, *89*, 120–134. https://doi.org/10.1016/j.autcon.2018.01.014

Chen, J, & Cho, Y. K. (2018). Point-to-point Comparison Method for Automated Scan-vs-BIM Deviation Detection. *{Proceedings of the 2018 17th International Conference on Computing in Civil and Building Engineering, Tampere, Finland*, 5–7. https://www.academia.edu/download/56812721/Jingdao Chen_full_paper2_icccbe2018.pdf

Chen, Jingdao, Kira, Z., & Cho, Y. K. (2019). Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. *Journal of Computing in Civil Engineering*, *33*(4), 04019027. http://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000842

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, *10*(3), 145–155. https://doi.org/10.1016/0262-8856(92)90066-C

Fischler, M. A., & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with. *Communications of the ACM*, *24*(6).

Girardeau-Montaut, D. (2015). CloudCompare: 3D point cloud and mesh processing software. *Webpage: Http://Www.Cloudcompare.Org*.

Jacob-Loyola, N., Muñoz-La Rivera, F., Herrera, R. F., & Atencio, E. (2021). Unmanned aerial vehicles (Uavs) for physical progress monitoring of construction. *Sensors*, *21*(12). https://doi.org/10.3390/s21124227

Kim, Changmin, Lee, J., Cho, M., & Kim, C. (2011). Fully automated registration of 3D CAD model with point cloud from construction site. *Proceedings of the 28th International Symposium on Automation and Robotics in Construction, ISARC 2011*. https://doi.org/10.22260/isarc2011/0169

Kim, Changwan, Kim, C., & Son, H. (2013). Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction*, *31*, 75–82. https://doi.org/10.1016/j.autcon.2012.11.041

Kim, S., Kim, S., & Lee, D. E. (2020). Sustainable application of hybrid point cloud and BIM method for tracking construction progress. *Sustainability (Switzerland)*, *12*(10), 1–16. https://doi.org/10.3390/su12104106

KUIPERS, J. B. (1999). Quaternions and Rotation Sequences. *Geometry, Integrability and Quantization*, 127–143. https://doi.org/10.2307/j.ctvx5wc3k

Lin, J. J., Han, K. K., & Golparvar-Fard, M. (2015). A framework for model-driven acquisition and analytics of visual data using UAVs for automated construction progress monitoring. *Congress on Computing in Civil Engineering, Proceedings*, *2015-January*(January), 156–164. https://doi.org/10.1061/9780784479247.020

Perez-Perez, Y., Golparvar-Fard, M., & El-Rayes, K. (2021a). Scan2BIM-NET: Deep Learning Method for Segmentation of Point Clouds for Scan-to-BIM. *Journal of Construction Engineering and Management*, *147*(9). https://doi.org/10.1061/(asce)co.1943-7862.0002132

Perez-Perez, Y., Golparvar-Fard, M., & El-Rayes, K. (2021b). Segmentation of point clouds via joint semantic and geometric features for 3D modeling of the built environment. *Automation in Construction*, *125*(March 2020), 103584. https://doi.org/10.1016/j.autcon.2021.103584

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*. https://doi.org/10.1109/CVPR.2017.16

Rusu, R. B., Blodow, N., & Beetz, M. (2009). Fast Point Feature Histograms (FPFH) for 3D registration. *2009 IEEE International Conference on Robotics and Automation*, 3212–3217. https://doi.org/10.1109/ROBOT.2009.5152473

Scott, S., & Assadi, S. (1999). A survey of the site records kept by construction supervisors. *Construction Management and Economics*, *17*(3), 375–382. https://doi.org/10.1080/014461999371574

Segal, A. V, Haehnel, D., & Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*, 435.

Swetha Koppula, H., Anand, A., Joachims, T., & Saxena, A. (2011). *Semantic Labeling of 3D Point Clouds for Indoor Scenes*. 1–9. http://pr.cs.cornell.edu/sceneunderstanding

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). *KPConv: Flexible and Deformable Convolution for Point Clouds*. http://arxiv.org/abs/1904.08889

Truong-Hong, L., & Lindenbergh, R. (2022). Extracting structural components of concrete buildings from laser scanning point clouds from construction sites. *Advanced Engineering Informatics*, *51*, 101490. https://doi.org/10.1016/j.aei.2021.101490

Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012). Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction*, *22*, 414–421. https://doi.org/10.1016/j.autcon.2011.10.003

Weinmann, M., Jutzi, B., Hinz, S., & Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, *105*, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Zhao, Y., Zhao, H., Radanovic, M., & Khoshelham, K. (2022). A unified framework for automated registration of point clouds, mesh surfaces and 3D models by using planar surfaces. *Photogrammetric Record*, *37*(180), 366–384. https://doi.org/10.1111/phor.12428