

EXEMPLAR-BASED BUILDING ELEMENT RETRIEVAL FROM POINT CLOUDS

Jingdao Chen¹ and Yong K. Cho^{2*}

¹ Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, U.S.A

² School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, U.S.A

* Corresponding author

ABSTRACT Laser scanning of built infrastructure is an integral process in construction management for accurate progress estimation, deviation detection, and asset tracking. Raw data from laser-scanned point clouds can be converted into portable, semantically-rich formats in a process known as Scan-to-BIM (Building Information Model). Manual annotation for placing BIM objects in an as-built model is time-consuming and error-prone. Whereas, automated methods for Scan-to-BIM rely on registration of pre-built Computer-Aided Design (CAD) models with the point cloud to detect objects of interest. This registration process is also computationally-intensive and depends on the availability of matching CAD models. Thus, this paper proposes an exemplar-based search method to efficiently retrieve duplicate elements from a point cloud. The point cloud is first processed with a machine learning-based feature extractor that can derive feature vectors at each point from geometrical information. Next, the point cloud is grouped into segments using K-means clustering and region-growing algorithms. Then, the user selects an instance of the building component of interest from the point cloud from the output of point cloud segmentation. Duplicate elements are retrieved from the original point cloud by sliding a 3D bounding box along the scene and calculating the feature correlation scores with respect to the exemplar element. A peak finding algorithm is used to group the detections into non-overlapping units. The proposed method is validated with laser-scanned point clouds of the façade of a university building. Experimental results showed that the proposed method achieved an overall precision of 96% and recall rate of 82%.

1. Introduction

The 3D geometry of built infrastructure is commonly acquired in the form of 3D point clouds. These point clouds can be acquired using technologies such as photogrammetry (Dai and Lu 2010), RGBD sensing (Roca *et al.* 2013), or laser scanning (Fekete *et al.* 2010). In building construction and management engineering, it is often desirable to reconstruct an as-built building model from 3D point clouds of the site. This entails converting the raw, unorganized point cloud data into portable, semantically-rich Building Information Model (BIM) formats that are easily accessible to engineers and site managers. The obtained building model can be used for multiple applications such as construction progress monitoring (Rebolj *et al.* 2017), asset management (Turkan *et al.* 2014a), deviation detection (Chen and Cho 2018), safety analysis (Park *et al.* 2016), and restoration of historical buildings (Armesto-gonzález *et al.* 2010).

One important subtask in site modeling is building element retrieval, which is identifying the number and locations of specific building elements which are present in the as-built environment that match a query element. Such elements include structural components (Son and Kim 2017), Mechanical, Electrical, and Plumbing (MEP) elements (Bosché *et al.* 2013), and temporary structures (Turkan *et al.* 2014b). Manual annotation of these objects based on point cloud data is labor-intensive, tedious and time-consuming due to the repetitive nature of this task. Thus, several automated methods have been proposed to expediate the task of 3D object

extraction from building point clouds (Tang *et al.* 2010).

Conventional automated methods for identifying 3D objects in a point cloud include simple matching of parametric shapes such as planes and cylinders to the point cloud. However, this method fails to retrieve objects with more complex geometry. Other methods rely on registration and matching of 3D Computer Aided Design (CAD) models to the point cloud. Such methods work well when the point cloud data is clean and is a good match with the 3D models. However, they are not easily generalizable to building sites that contain complexities such as occlusion, clutter, and variations in point cloud density, surface roughness, and curvature (Dimitrov and Golparvar-Fard 2015). These factors and other scanning artefacts may cause the scanned point cloud to differ from the reference 3D model.

To overcome these limitations, this research proposes a semi-automated exemplar-based building element retrieval method for cases where the 3D CAD models are not available or do not match the point cloud data. In this case, the mismatch between point cloud geometry and CAD geometry is avoided by allowing end users to select the query element directly from the point cloud itself. Through the developed user interface, an exemplar building element is first manually selected from the point cloud data. This exemplar serves as the query object from which similar object instances can be automatically retrieved from the point cloud scene. Candidate matches are scored based on the similarity of point features to that of the exemplar. Finally, a peak finding algorithm is used to group together

neighboring detections and eliminate false positives.

2. Literature Review

The task of object retrieval can be defined as the process of automatically finding all instances of a query object from a larger dataset (Arandjelovi and Zisserman 2012). In the context of building construction and management, it can refer to the task of finding instances of a building element of interest from laser-scanned point clouds of built infrastructure. There have been multiple methods proposed in the literature to perform object retrieval from point clouds, namely geometry-based, model-based and feature-based methods.

2.1 Geometry-based Object Retrieval

The geometrical shape of building elements such as planar (Wang *et al.* 2015; Xiong *et al.* 2013) and cylindrical structures (Kalasapudi *et al.* 2017) can be used as a basis for object retrieval. The object retrieval process uses plane-fitting and shape-fitting algorithms to recover simple geometric shapes from point cloud data. By performing point cloud clustering and fitting 3D bounding boxes around detected clusters, a volumetric representation of building elements can also be recovered (Chen *et al.* 2017). This is an effective method for identifying building elements since most building elements such as walls and doors can be decomposed into planar units (Bueno *et al.* 2018). However, this property can also lead to ambiguity in object retrieval since different building elements can have similar planar shapes. In addition, this method does not work with objects with complex geometry such as light fixtures and curved walls.

2.2 Model-based Object Retrieval

When the as-planned BIM model for a scanned site is available, object retrieval can be performed by registration and matching of 3D models with the scanned point cloud (Bosché 2010). If the designed BIM model is not available, a library or database of 3D Computer-Aided-Design (CAD) models can also be used to perform matching (Chen *et al.* 2018). However, fully-automated registration of CAD models to a point cloud is computationally challenging due to the large size of point clouds acquired from buildings. Thus, semi-automated methods are commonly used where the initial coarse registration is performed manually (Turkan *et al.* 2014a). This method also assumes that the scanned point cloud can be closely-matched to CAD models. In reality, there are often noise, occlusion, and other scanning artefacts that will cause the scanned point cloud to differ from the original 3D model.

2.3 Feature-based Object Retrieval

Another method for 3D object retrieval is to compute 3D feature descriptors from point clouds and match the feature descriptors of the laser scan data to that of the query object. (Chen *et al.* 2019a; b). These features, which incorporate statistics about distance, area, and angle (Wohlkinger and Vincze 2011), can be defined under a machine learning framework that can compute robust features that uniquely describe each type of building element. The feature descriptors

are able to compress large point cloud data into smaller feature vectors that can succinctly express the geometrical structure of objects that is robust to noise and small variations between similar objects. This enables the feature descriptors to handle inter-class variation and generalize to unknown objects (Chen *et al.* 2016). The shortcoming of this method is that a large database of building element models has to be acquired in advance to train machine learning algorithms.

3. Methodology

The proposed methodology for building elements from point clouds is divided into four steps, namely (i) point feature computation (ii) point cloud segmentation (iii) exemplar selection and (iv) candidate element matching. Each step will be described in detail in the following subsections:

3.1 Point Feature Computation

The first step in processing the point cloud data is to convert the raw point cloud, where each point only has XYZ and optionally RGB information, into a more semantically meaningful representation. This is achieved by using a machine learning-based feature extractor that can derive feature vectors at each point from geometrical information. The use of feature vectors is more advantageous than pure geometric information since they are more robust to noise and small variations between similar objects.

The proposed feature extractor is trained on the auxiliary task of segmenting point clouds of buildings for which the ground truth BIM is available. A total of seven BIM models were used as training data (three of which are shown in Figure 1). Each BIM model is first converted into an intermediate triangle mesh representation, then converted into point clouds by randomly sampling points along the surfaces of the mesh model. Since it is derived from BIM with object annotations, this synthesized dataset contains ground truth labels about which points belong to the same or different objects.

A deep learning model is used as the feature extractor and trained on this dataset with the triplet loss function (Schroff and Philbin 2015). For each input point with XYZ coordinates, the feature extractor aggregates surrounding points at three different resolutions (0.2m, 0.4m, and 0.6m) and computes a 50-dimensional feature vector for that point. That is, it takes as input an $N \times 3$ matrix (N points with XYZ coordinates) and outputs an $N \times 50$ matrix (N points with 50 features each). To train the feature extractor, the training routine iteratively samples three random points from the training data, consisting of one positive pair (belonging to the same object) and one negative pair (belonging to different objects). The training routine attempts to optimize the weights of the feature extractor such that the resulting feature vectors of the positive pair are similar whereas the feature vectors of the negative pair are different. This process is iteratively carried out with different combinations of training samples until the loss function converges.

Figure 1: Examples of synthesized point cloud data used as training samples for the point feature extractor. Points are coloured based on object membership.

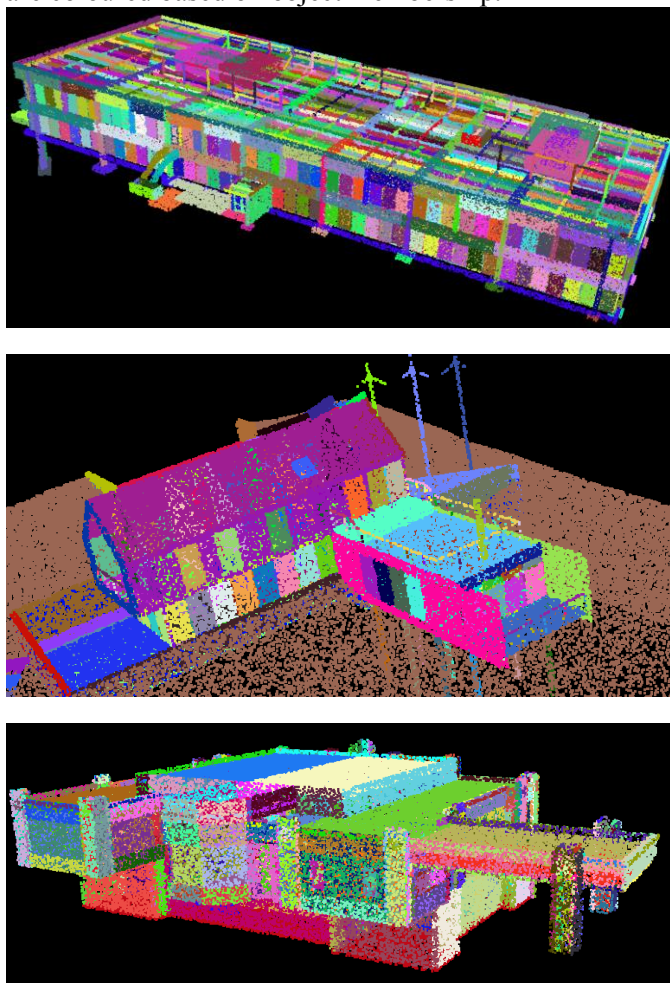


Figure 2: (a) Original point cloud and (b) color-coded point cloud based on computed feature vectors

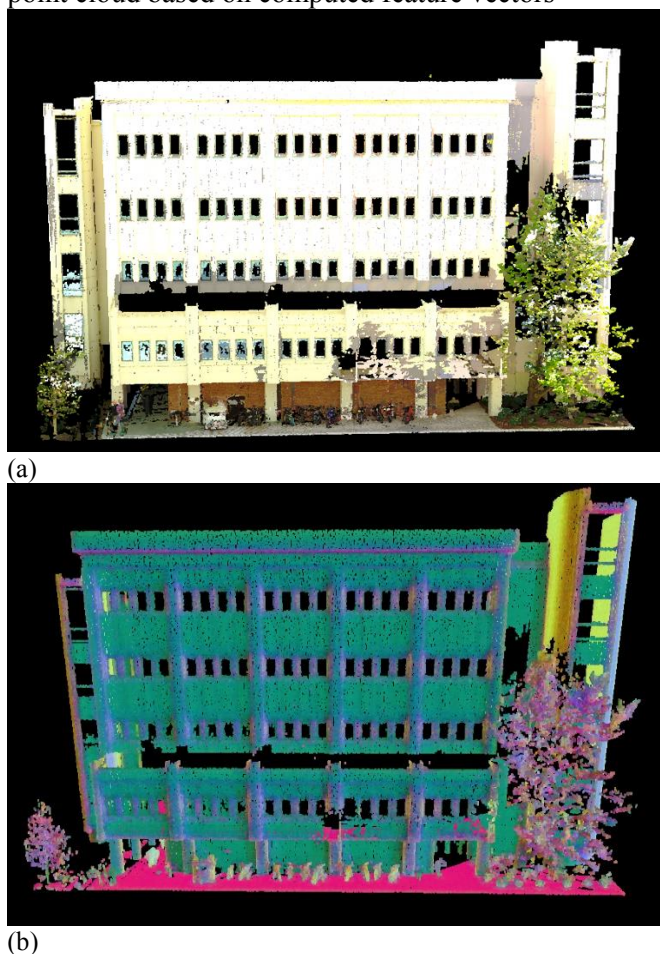


Figure 2 shows an example of the results of applying the proposed feature extractor applied on the façade of a university building. Figure 2a shows the original laser-scanned point cloud whereas Figure 2b shows the color-coded point cloud after feature extraction. For visualization purposes, the 50-dimensional feature vectors are reduced to 3 dimensions using Principal Component Analysis (Locantore *et al.* 1999) such that it can be displayed in RGB colours. As shown in Figure 2b, the feature-enriched point cloud constitutes a more semantically meaningful data representation compared to the original point cloud. For example, the ground is labeled red, the walls are labeled green, the windows are labeled blue, and the trees are labeled purple.

3.2 Point Cloud Segmentation

The next step in processing the point cloud data is to group neighbouring points together into cohesive segments which have object-level semantics. The K-means clustering technique is first used to determine latent class labels for each point. The latent class label differentiates between different classes of points, such as points from vertical wall segments, points from horizontal floor segments, and points on edges. These latent classes are not hand-coded but inferred from the dataset as a form of unsupervised learning. K-means clustering involves the following steps: (i) randomly initialize K cluster centres, (ii) assign the label for each point based on the minimum Euclidean distance between its feature vector and the cluster centre, (iii) update the cluster centre to be the mean feature vector of its member points, and (iv) repeat until the labels remain unchanged. Figure 3 shows the resulting point cloud after K-means clustering ($K = 50$) where each point is coloured according to its latent class label.

Next, a region growing method is used to form point cloud segments between points that have the same latent class label. Seed points are iteratively selected from the point cloud and segments are created by merging all neighbouring points that have the same latent class label as the seed point. Figure 4

shows the point cloud segmentation result, where each point cloud segment is visualized in a different colour.

Figure 3: Color-coded point cloud after K-means clustering

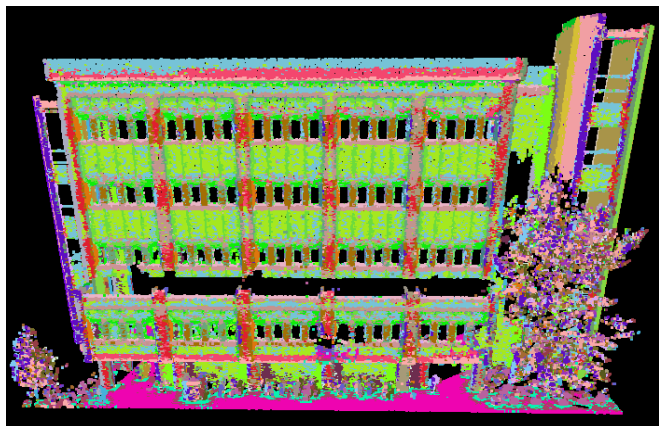
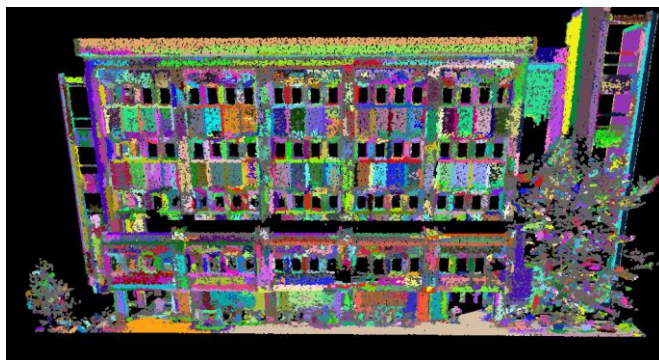


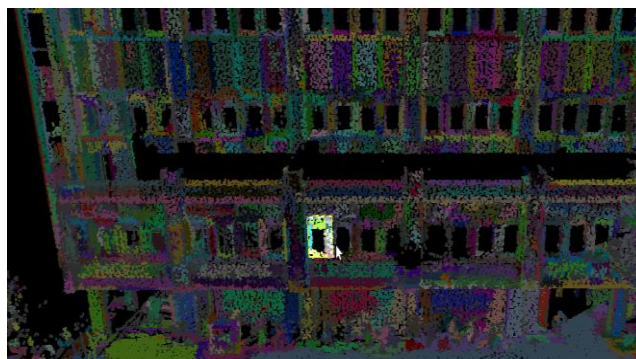
Figure 4: Point cloud segmentation based on region growing. Each point cloud segment is visualized in a different colour.



3.3 Exemplar Selection

The exemplar building element used for building element retrieval is obtained by having the user select an instance of the building component of interest from the point cloud from the output of point cloud segmentation. A group of neighbouring point cloud segments corresponding to an object is first selected from the global point cloud. A 3D bounding box is then drawn around selected points, as shown in Figure 5. The selected points are also highlighted to enable easy visualization.

Figure 5: Selection of exemplar building element from the user interface



3.4 Candidate Element Matching

The final step in building element retrieval is to find candidate elements in the global point cloud and determine positive matches to the selected exemplar. Candidate elements are first extracted by sliding a 3D bounding box around groups of points that have the same dimensions as the exemplar. Then, positive matches are determined by computing feature correlation scores for each candidate element and executing a peak finding routine. The feature correlation score, C_i , is computed with respect to the exemplar element as shown in Equation 1, where f indicates the point feature vector, p^i indicates points on the candidate element, and p^E indicates points on the exemplar element. The closer the feature similarity between two building elements, the higher the feature correlation score.

$$C_i = -\sum_k \|f(p_k^i) - f(p_k^E)\|^2 \quad (1)$$

After the feature correlation is computed, a peak finding algorithm is executed to determine positive matches among the candidate elements. A peak is defined as a locally-maximal value of feature correlation for a candidate element compared to its neighbouring elements. Figure 6 shows the result of the peak finding algorithm on the feature correlation scores. K-means clustering is used to group the matches into 3 groups: (i) strong matches (ii) weak matches and (iii) non-matches. Strong matches indicate detected building elements that have high similarity to the query element. Whereas weak matches indicate detected elements that are similar to the query element but differ slightly in point cloud geometry due to incomplete data, occlusion and other factors. Non-matches are the background elements that do not match the query element.

Finally, 3D bounding boxes are drawn around the positive matches and displayed through the user interface (Figure 7). The total number of positive matches is also shown in the user interface.

Figure 6: Peak finding on feature correlation scores to determine building element matches

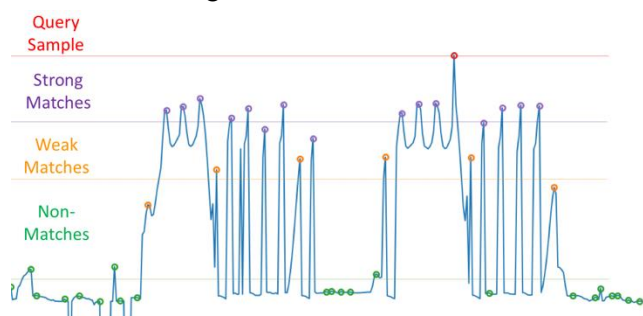
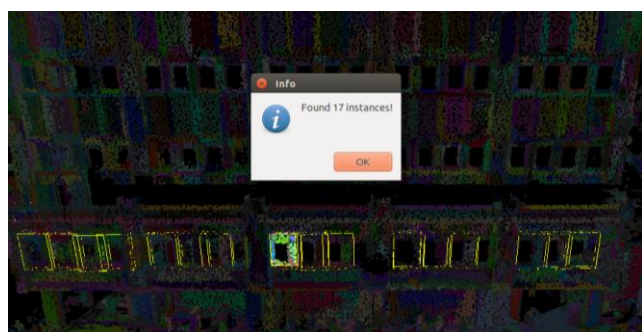


Figure 7: Visualization of building element retrieval results on the user interface



4. Results

4.1 Software Implementation

The machine learning-based feature extractor (discussed in Section 3.1) was implemented in Python and *Tensorflow* (Martin Abadi *et al.* 2015). The model was pre-trained offline using GPU acceleration. On the other hand, the user interface was developed in C++ using *wxWidgets* (Smart *et al.* 2005). The user interface contains menu items and dialog boxes to enable a smooth query and retrieval process. The graphics are rendered and displayed in *OpenGL* which allows users to visualize and interact with 3D point clouds.

4.2 Object retrieval results

The object retrieval performance was evaluated on laser-scanned point clouds in E57 format of a five-storey building at Georgia Institute of Technology. The point cloud scene, which spans an area of 35m x 16m x 28m, originally consists of 760,000 points but is then downsampled to a resolution of 0.1m, resulting in 200,000 points. Table 1 shows the number of retrieved elements for five different categories of building elements in terms of true positives, false positives, and false negatives. From these results, the overall precision is 96% whereas the overall recall rate is 82%. The proposed method performed well for most building elements except windows. This is because windows are transparent to laser scanning and as a result, only the window frame can be detected.

Table 1: Precision and recall for each category of building element

Building Element	Number	Precision	Recall
Small windows	60	97%	63%
Large windows	8	100%	75%
Columns	4	100%	100%
Wall segments	40	85%	73%
Light fixtures	5	100%	100%
Overall	117	96%	82%

4.3 Computation time

This section performs an evaluation of the proposed building element retrieval method based on computation time, measured using a desktop computer with an Intel Xeon E3-1200 CPU and a NVIDIA GTX1080 GPU. Table 2 shows the computation time measured in seconds based on an input point cloud downsampled to 200,000 points. Although pre-processing steps such as feature extraction and region growing are time-consuming, the actual element matching and computation of building element retrieval results is relatively fast. This is advantageous because the pre-processing steps only have to be performed once for each building point cloud whereas the element retrieval has to be performed multiple times for different building elements.

Table 2: Computation time for each step of building element retrieval

Step	Computation time (s)
Feature extraction	26.9
K-means clustering	16.0
Region growing	1.1
Element matching	0.04

5. Conclusion

This paper proposed a semi-automated building element retrieval method to identify similar elements to a user-provided exemplar from a point cloud scene. The point cloud is first processed with a machine learning-based feature extractor that can derive feature vectors at each point from geometrical information. Next, the point cloud is grouped into segments using K-means clustering and region-growing algorithms. A user-selected exemplar is provided as input to the retrieval algorithm, which computes feature correlation scores and

executes a peak finding algorithm to determine positive matches among the candidate elements. Compared to conventional 3D object retrieval methods, the proposed method does not require pre-built CAD models and is less sensitive to noise, occlusion, and other scanning artefacts. Object retrieval experiments on laser-scanned point clouds of the façade of a university building showed that the method achieved an overall precision of 96% and recall rate of 82%. Although the point cloud pre-processing steps have relatively high computation time, the actual retrieval step remains reasonably efficient. The proposed method has promising applications in building construction and management such as construction progress monitoring, deviation detection, and restoration of historical buildings.

6. Acknowledgments

The work reported herein was supported by the United States Air Force Office of Scientific Research (Award No. FA2386-17-1-4655) and by a grant (18CTAP-C144787-01) funded by the Ministry of Land, Infrastructure, and Transport (MOLIT) of Korea Agency for Infrastructure Technology Advancement (KAIA). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force and MOLIT.

7. References

- Arandjelovi R and Zisserman A (2012) Three things everyone should know to improve object retrieval. *2012 IEEE Conference Computer on Vision and Pattern Recognition*, IEEE, pp.2911-2918.
- Armesto-gonzález J *et al.* (2010) Terrestrial laser scanning intensity data applied to damage detection for historical buildings. *Journal of Archaeological Science*, Elsevier Ltd, 37(12), 3037–3047.
- Bosché F (2010) Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics*, Elsevier Ltd, 24(1), 107–118.
- Bosché F *et al.* (2013) Tracking the Built Status of MEP Works: Assessing the Value of a Scan-vs.-BIM System. *Journal of Computing in Civil Engineering*, 0(ja), 5014004.
- Bosche F, and Haas, C T (2008) Automated retrieval of 3D CAD model objects in construction range images. *Automation in Construction*, 17(4), 499–512.
- Bueno M *et al.* (2018) 4-Plane congruent sets for automatic registration of as-is 3D point clouds with 3D BIM models. *Automation in Construction*, Elsevier, 89(December 2017), 120–134.
- Chen J and Cho Y K (2018) Point-to-point Comparison Method for Automated Scan-vs-BIM Deviation Detection. *Proceedings - 17th International Conference on Computing in Civil and Building Engineering*.
- Chen J *et al.* (2019a) “Multi-view Incremental Segmentation of 3D Point Clouds for Mobile Robots.” *IEEE Robotics and Automation Letters*, in press.
- Chen J *et al.* (2018) Region Proposal Mechanism for Building Element Recognition for Advanced Scan-to-BIM Process. *ASCE Construction Research Congress*, New Orleans, LA.
- Chen J *et al.* (2017). Unsupervised Recognition of Volumetric Structural Components from Building Point Clouds. *Computing in Civil Engineering 2017*, (June), 34–42.
- Chen J *et al.* (2016) Principal Axes Descriptor for Automated Construction-Equipment Classification from Point Clouds.” *Journal of Computing in Civil Engineering*, 1–12.
- Chen J *et al.* (2019b). Deep Learning Approach to Point Cloud Scene Understanding for Automated Scan to 3D Reconstruction. *ASCE Journal of Computing in Civil Engineering*, in press.
- Dai F and Lu M (2010) Assessing the Accuracy of Applying Photogrammetry to Take Geometric Measurements on Building Products.” *Journal of Construction Engineering and Management*, 136(February), 242–250.
- Dimitrov, A., and Golparvar-Fard, M. (2015). “Segmentation of building point cloud models including detailed architectural/structural features and MEP systems.” *Automation in Construction*, Elsevier B.V., 51(C), 32–45.
- Fekete, S *et al.* (2010). “Geotechnical and operational applications for 3-dimensional laser scanning in drill and blast tunnels.” *Tunnelling and Underground Space Technology*, Elsevier Ltd, 25(5), 614–628.
- Kalaspudi, V. S., Tang, P., and Turkan, Y. (2017). “Computationally efficient change analysis of piece-wise cylindrical building elements for proactive project control.” *Automation in Construction*, Elsevier, 81(February), 300–312.
- Locantore, N., Marron, J. S., Simpson, D. G., Tripoli, N., Zhang, J. T., Cohen, K. L., Boente, G., Fraiman, R., Brumback, B., Croux, C., Fan, J., Kneip, A., Marden, J. I., Peña, D., Prieto, J., Ramsay, J. O., Valderrama, M. J., Aguilera, A. M., Locantore, N., Marron, J. S., Simpson, D. G., Tripoli, N., Zhang, J. T., and Cohen, K. L. (1999). “Robust principal component analysis for functional data.” *Test*, 8(1), 1–73.
- Abadi M *et al.* (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.
- Park J *et al.* (2016) Framework of Automated Construction-Safety Monitoring Using Cloud-Enabled BIM and BLE Mobile Tracking Sensors. *Journal of Construction Engineering and Management*, American Society of Civil Engineers, 5016019.
- Rebolj D *et al.* (2017) Point cloud quality requirements for Scan-vs-BIM based automated construction progress monitoring. *Automation in Construction*, 84(September), 323–334.
- Roca D *et al.* (2013) Low-cost aerial unit for outdoor inspection of building facades. *Automation in Construction*, Elsevier B.V., 36, 128–135.
- Schroff F and Philbin J (2015) FaceNet: A Unified Embedding for Face Recognition and Clustering. *Proc.*

Computer Vision and Pattern Recognition (CVPR), IEEE.

Smart J *et al.* (2005) *Cross-Platform GUI Programming with wxWidgets*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Son H and Kim C (2017) Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity. *Advanced Engineering Informatics*, Elsevier, 34(November), 114–124.

Tang, P *et al.* (2010) Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7), 829–843.

Turkan Y *et al.* (2014a). “Tracking of secondary and temporary objects in structural concrete work.” *Construction Innovation: Information, Process, Management*, 14(2), 145–167.

Turkan Y *et al.* (2014b) Tracking of secondary and temporary objects in structural concrete work - ProQuest. *Construction Innovation*, 14(2), 145–167.

Wang C *et al.* (2015) Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. *Automation in Construction*, Elsevier B.V., 56, 1–13.

Wohlkinger W and Vincze M. (2011) Ensemble of shape functions for 3D object classification. *2011 IEEE International Conference on Robotics and Biomimetics*, 2987–2992.

Xiong X *et al.* (2013) Automatic creation of semantically rich 3D building models from laser scanner data. *Automation in Construction*, Elsevier B.V., 31, 325–337.