Automated segmentation and analysis of point clouds of pier foundations using Pier Inspection and Evaluation Report (PIER)

Amine Turki[1], Nolan W. Hayes[1], Balaji Selvakumar[1], Bryan P. Maldonado[1], and Diana Hun[1]

[1]Buildings and Transportation Science Division, Oak Ridge National Laboratory, United States of America *

turkia@ornl.gov, hayesnw@ornl.gov, selvakumarb@ornl.gov, maldonadopbp@ornl.gov, hunde@ornl.gov

Abstract -

Pier foundations are commonly used in locations with unstable soil or where other types of foundations are unsuitable or cost prohibitive. A pier foundation consists of vertical columns to support the structure and elevate it above the ground. Common materials for pier foundations include masonry, concrete, timber, and steel. The methods for accurate placement of pier foundations have remained relatively unchanged for decades. For simple installations, construction chalk lines are used to layout the locations of piers to ensure accurate placement and elevation. For more complex installations, surveying instruments operated by trained professionals are employed to accurately locate piers and assess correct elevation before construction. After installation, another survey may need to be performed to assess the quality of the as-built foundation. However, with the advent of terrestrial laser scanners (TLS), the means now exist for contractors to conduct their own assessments of as-built foundations. The major barrier preventing contractors from performing their own assessments of as-built foundation quality is the segmentation and analysis of point cloud data, a skill that often requires a trained user. The objective of this research is to develop a software tool (PIER: Pier Inspection and Evaluation Report) to enable automated segmentation and analysis of point clouds of pier foundations. In this paper, the automated segmentation and analysis algorithms are detailed. A mockup lay out of pier foundations was built using concrete masonry units, and the algorithms were tested to evaluate performance. Limitations of the current algorithms and future research direction are discussed.

Keywords -

foundation, pier, evaluation, automation, real-time

1 Introduction

In the United States, pier foundations are often used to support a variety of building types and infrastructure. Common applications for piers include foundations for manufactured housing, site-built housing with crawlspaces, buildings constructed on unstable soil, and bridges, especially those over water. Pier foundations are unique in that they are discontinuously distributed across the footprint of the structure. As a result, the accurate placement of pier locations and elevations is critical.

Pier foundations can take the form of many shapes and sizes. An array of materials are traditionally used to construct piers (Figure 1). For example, timber, steel, cast-in-place concrete, precast concrete, and masonry are all typically used for pier foundations across diverse building types. Due to variability in materials, sizes, and shapes, methods of placement and installation can vary significantly.

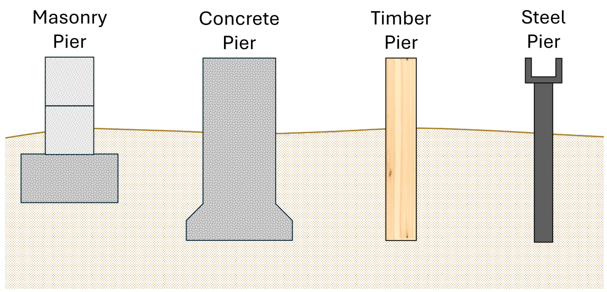

Figure 1. Types of pier foundations.

1.1 Pier foundation positioning and elevation control

The methods for accurate placement of pier foundations have remained relatively the same for decades [1, 2, 3]. For small construction sites, the layout of foundations is made using strings, chalk lines, measuring tapes, or bubble levels. Some combination of these tools is necessary to ensure squarely-set, evenly-spaced pier foundations with accurate top elevations. However, as the size of the construction site increases, the usability of these tools diminishes, and the likelihood of user error increases. More advanced tools such as laser levels can be employed with a higher degree of accuracy to ensure that the top of each pier foundation is constructed at the correct elevation. For larger construction projects, it is common to hire a professional surveyor to measure and locate foundations prior to construction. After the construction of the foundations, it is sometimes necessary to perform another survey to determine if the as-built elevations and locations are within tolerance. Recent research has developed methods to analyze piers in challenging locations, such as underwater [4]; however these methods are not typically applicable under normal conditions in open air.

Because these tools rely on skilled labor to assess distances and read measurements, there is a chance for errors. Accurate placement of piers is critical because the structure above must be adequately supported at the correct locations. As a result, it is common to incorporate some tolerance for pier foundations so that if the placement of a pier is inaccurate to some degree, the subsequent construction is minimally affected. However, excessive errors can cause constructability issues, which can delay the entire construction schedule. For example, in a series of piers supporting an array of beams, if a pier is misplaced by more than the width of a beam, either the pier must be demolished and correctly reconstructed or the layout of the beam structure must be altered to accommodate the misplaced pier. Errors can often go unnoticed during the foundation construction process, causing significant delays in the construction timeline.

94% of construction firms report having difficulty finding workers to hire [5]. Current quality assessments of pier foundations for complex construction sites are conducted by professional surveyors, who are becoming harder to find and more expensive. The transfer of knowledge and continued development of skilled labor is necessary to ensure that construction demand can meet the growing needs of society. Digital tools can increase productivity and alleviate some of the strain on the diminishing workforce. However, for foundation layout and construction, the availability of digital tools to improve productivity is low. Some options exist on the market to automate some aspects of foundation evaluation; however, no option currently exists specifically to evaluate the quality of construction for pier foundations.

1.2 Laser scanning and point cloud segmentation

Modern laser scanning devices can collect more than 2 million points per second and are accurate at long ranges with peak range accuracy of 2 mm + 2 ppm or better [6]. These simple, easy-to-use scanners have the potential to expedite the measurement of as-built quality of construction and generate real-time guidance to correct issues. With further development, the affordability and accessibility of these devices will also add to their potential. The accompanying software with modern laser scanners can automate and simplify the process of registering multiple scans into a single frame, which is usually a time-consuming and difficult process. Laser scanners have the potential to replace manual measurement methods and tools used during the traditional process of construction if point cloud data and analysis can be streamlined and automated.

The major aspect of laser scanning that currently limits adoption is the segmentation of point cloud data. The fastest method of scanning for most TLS is typically a full 360° scan during which every object in line of sight is measured. Once the scan is complete, the data is imported into software and the points belonging to the objects of interest (the piers in this case) are retrieved from the rest of the points. This process is called segmentation. Automated segmentation of points clouds for varying applications is a major point of interest in current research [7, 8, 9, 10]. Automated point clouds segmentation techniques exist to perform tasks ranging from prefabricated tolerance compliance assessment [11] to automated digital twin generation [12]. The segmentation process can be turned from a several-hour manual process into a few second-long automated process by incorporating automation through artificial intelligence. Previous research by the authors developed automated segmentation and analysis algorithms for the flatness and levelness of concrete slabs [6]. The current research expands upon the existing algorithms to automate segmentation and analysis of pier foundations. This paper details automated segmentation and analysis algorithms to identify and analyze the quality of construction for pier foundations.

2 Methodology

The focus of this research is the development of automated segmentation and analysis algorithms for point clouds of pier foundations. The algorithms are incorporated into a software tool called the Pier Inspection and Evaluation Report (PIER). As part of the analysis, the user is informed in real-time about differences in the top elevations of piers and misalignment of pier locations based on an assumed orthogonal planar layout. The method takes as input a dense point cloud and does not require user interaction to automatically segment points belonging only to piers and analyze the quality of construction. For the initial design of the algorithms, masonry piers were used.

2.1 Segmentation

The segmentation process is performed to identify and isolate points belonging only to piers from a dense point cloud. This segmentation is necessary to identify and isolate pier locations before the automated analysis can be conducted. Figure 2 illustrates the steps in the segmentation procedure.

The first step is to remove points above the scanner origin. The top surfaces of piers will not be visible from the scanner unless the scanner is located at a higher elevation than all the top piers surfaces. As a result, only points below the scanner are relevant. This step helps to reduce the total number of points initially and ensure that any elements above the scanner origin are not incorrectly identified as piers. To perform this removal, the point cloud is cropped such that all points with an elevation above zero are removed from the point cloud.

It is essential to remove the points belonging to the ground around the piers. The next step consists of removing points belonging to the ground by using random sample consensus (RANSAC). RANSAC consists of randomly sampling points to generate planes and select inlier point sets that fall within a specified normal distance from that plane [13]. All points that fall within a specified distance threshold are considered inliers within the selected plane. The procedure in PIER randomly selects five points in the cloud to fit a plane. The distance threshold represents the normal distance from the plane generated and assigns points within this distance as inliers. After a specified number of iterations, the plane with the largest number of inlier points is returned as the plane representing the ground [6]. For instances where the ground around the piers is largely non-planar, the procedure may struggle to effectively remove the ground. However, it is only necessary to remove the ground around the piers to ensure there is clear separation between the points at the bases of piers and points belonging to the ground.

The next step is to prepare the point cloud data for computationally expensive algorithms. To ensure the code is not too computationally expensive and that the run time is minimized, data reduction is achieved by voxel-down sampling [14]. A grid of cubes with a specified dimension is computed. All points within each cube are averaged and reduced to a single point. In this work, a voxel size of 12.5 cm appears to be effective in reducing the point cloud while still accurately providing enough fidelity in the data for the remaining steps in the procedure [6].

Points at the edge of the point cloud are also removed to make the code run more quickly and improve robustness. To do so, convex hulls and K-D tree are used. The convex hull is an algorithm that computes the smallest convex shape of spatial data points: given $P = \{p_1, p_2, \ldots, p_n\}$, $\text{Convex Hull}(P) = \bigcap\{H \mid P \subseteq H,\ H \text{ is convex}\}$. The implementation of the convex hull algorithm in the Open3D library (Qhull algorithm) is used in this work [14]. In this implementation, a tetrahedron is initialized with vertices at extreme points (maxima and minima of x, y, and z). Then the remaining points outside the shape are checked and included as long as the convex property is maintained [15]. Computing the convex hull of the point cloud allows us to identify points composing the vertices or the borders of the hull. The next step is to find and remove the neighbors of these border points by using KD-Tree (k-dimensional tree) for computational optimization purposes. The K-D tree is a space-partitioning binary search tree for organizing points and enables the selection of points based on nearest neighbors [16]. Space is partitioned with an axis-aligned hyperplane. Embedded functions in Open3D easily allow the organization of the point cloud in a KD-Tree and selection of neighbors within a radius of the border points identified with the convex hull procedure to be removed.

After removing the ground and the points at the extremities of the point cloud, a density-based scan (DBSCAN) clustering method ([14] [17]) is used to identify the different clusters in order to isolate the point sets belonging to the piers. A set of filters are used to correctly identify which sets of clusters are pier foundations. The algorithm iterates through the projection of each cluster on xy plane to compare the area difference between the minimum oriented bounding box and the contour of the cluster. The minimum oriented bounding box algorithm determines the smallest rectangle at any orientation that encapsulates all points within the set [18]. The contour is computed using alpha shape: a generalization of convex hull that is capturing all vertices and allows concavity [19]. In addition, the contour area is compared with the user-specified dimensions of piers to ensure a robust selection of clusters belonging to piers. In most cases, the dimensions of the piers are normalized or easily quantified in the construction industry. The user input pier dimensions are not required but do improve the robustness of pier identification. All individual point clusters identified as piers are joined to create a single point cloud. At this stage, the point cloud includes only points that have been identified as piers and is ready for analysis.

2.2 Analysis

After segmenting the piers, labels are set based on the distance between the centroid of each pier and the origin of the point set. By using this naming convention, the identification of each pier remains consistent even if the user takes multiple scans and runs the segmentation and analysis multiple times. Labels are set as P_i. The current objectives of the analysis are to produce two plots: (1) a quantified illustration of the differences in elevations between the top surfaces of piers that allows the user to shim piers such that the top surface of each pier is at the same elevation, and (2) a quantified illustration of the

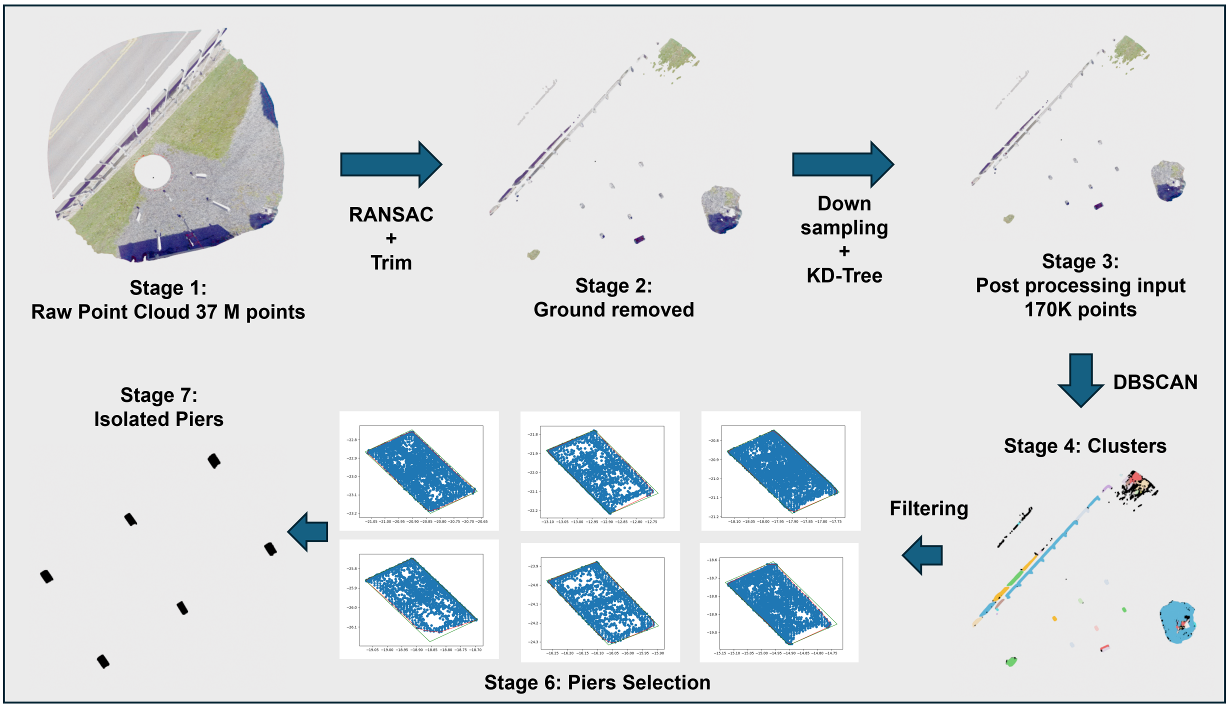

Figure 2. Steps in the automated segmentation process for pier foundations.

misalignment that informs the user about incorrect planar locations of piers.

To determine the difference in elevation between the top surfaces of piers, a point of reference must be established. The highest point of the point cloud, including only points identified as piers, is computed and referred to as h to be used as the reference. For each pier, the local maximum elevation h_i is identified, and a plane parallel to the xy-plane is created using the local maxima h_i at centroid c_i. For each of these planes, a perpendicular vector is computed, originating from the corresponding plane's centroid c_i and extending to the corresponding point where the z-coordinate matches the global maximum elevation h. This analysis results in the reported minimum distance from the top surface of the pier at the highest elevation to the top surface of each other pier as shown in Figure 3. With this information, the user can adjust piers top elevation by shimming appropriately. For example, in Figure 3, P2 is shown to be 18 cm below P6 (the datum). To correctly bring the top elevation of P2 to a uniform elevation, a shim or block with total thickness of 18 cm would need to be added to the top of the second pier. After shimming, the site can be scanned again, and the automated segmentation and analysis can be repeated to determine if the piers were correctly shimmed.

During the foundation construction, the planar locations of piers need to be accurate to avoid issues with the construction of the structure above. To give corrective

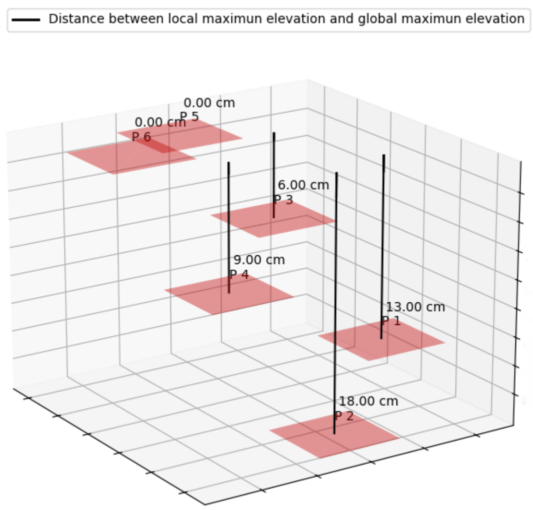

Figure 3. Elevation differences between top surfaces of piers.

guidance for planar location, the automated analysis generates an illustration of planar misalignment. To generate this guidance, the total point cloud including only piers is projected onto xy plane (Figure 4). In the current development, the algorithms assume a rectangular layout of piers. The measured diagonals formed between piers at corner locations are presented to show the quality of the rectangle produced by pier layout. If the measured distance of diagonals are substantially different, the user can know that the interior angles of the rectangle forming the layout of the piers are not orthogonal indicating an error in the construction.

Additionally, local guidance is presented to the user on how to correct the location of each pier to achieve an orthogonal rectangular layout. The contour and minimum oriented bounding box of the total point cloud including only piers are computed. For each pier, a minimum oriented bounding box is computed for all points on the xy plane. The closest point on the segment of the total bounding box from each pier extremity is given by their respective orthogonal projection (Figure 5). For example, let H be the minimum oriented bounding box of the total point cloud, and let a_i, b_i, c_i, d_i be the vertices generating the minimum oriented bounding box of the each pier P_i. For each a_i, b_i, c_i, d_i, the minimum distance between these points and H is computed and presented to the user. For example, as shown in Figure 5, the pier in the top left corner needs to be adjusted nearly 14 cm to be correctly set to create a planar rectangular layout with the remaining piers.

If the displacement between one of the points a_i, b_i, c_i, d_i for each P_i and H is less than 2 mm, it is not going to be plotted. Also if this displacement is higher than the dimension of the piers specified by the user, it is not going to be computed since it does not bring valuable information.

2.3 Limitations

In this version, the automated segmentation and analysis algorithms include some limitations. Each limitation will be addressed in future research.

First, this initial study focuses on the general performance and feasibility of the proposed algorithm for a single case study. Future research will focus on multiple demonstrations to evaluate robustness and accuracy of outputs.

The final alignment plot can be misleading if the errors are too large. Since the code is looking for the displacement needed so that all the piers are within an optimized rectangle, if large errors were made during placement, the assumption for a rectangular layout is not valid, and the algorithms generate incorrect guidance.

The number of points attributed to the piers is low compared to the remaining elements such as the ground. During the step where the ground is removed from the point cloud, a few failures occur where a pier is inadvertently removed in some instances.

In the current form, the automated analysis algorithms do not provide guidance on the levelness of the top surface of each pier. Information of the levelness of the top surface is critical to ensuring that constructability issues are not created. A feature to extract and report on the levelness of the top surface of each pier will be added to the algorithms.

In the current development, the algorithms assume a rectangular layout of piers. The measured diagonals formed between piers at corner locations are presented to show the quality of the rectangle produced by piers layout. In future developments, a digital twin is planned to be incorporated to generate guidance for complex layouts. However, in this case, alignment of the digital twin with the physical space will need to be performed to generate applicable correct guidance for users.

Due to the automated nature of the analysis procedure for correct planar rectangular layout, the algorithm assumes that the most exterior pier within the minimum oriented bounding box is the reference point. As a result, all guidance is to move the remaining piers outwards from the center of the layout. This assumption may not always be correct as a certain pier may need to be used as the reference. In future development, the user will be allowed to choose which piers should be treated as the reference point and remain unmoved.

3 Demonstration

The automated segmentation and analysis algorithms were tested on a mockup pier foundation construction located on the Oak Ridge National Laboratory site. Six masonry piers were placed in a planar rectangular layout with design dimensions of 7.5 m by 3.6 m. The mockup pier foundation was scanned using a terrestrial laser scanner.

The scan was made with a Leica MS60 which was placed in front of the piers and was set to do a 360° scan of the area to have a full view of the mockup construction site. The same scan parameters as in the previous paper [6] were used. Figure 6 shows the raw .pcd file of the scan that was loaded into the PIER digital tool. The masonry piers are clearly visible by their "shadows" cast from the laser scanning procedure.

The pier segmentation and analysis algorithms were written using Python. Table 1 shows the list of parameters and values used to analyze the mockup foundation. Voxel Size is the dimension of the voxel cube used for downsampling. The Plane Distance Threshold is the minimum normal distance between the RANSAC plane and points to be considered as inliers. Cluster Neighbor Radius is the maximum distance between points to be considered members of the same cluster. Cluster Minimum Points is the minimum number of points that a cluster must contain. KD-Tree Minimum Points is the minimum number of neighbors a point must have in order to not be considered a border point in the point cloud. KD-Tree search radius is the maximum distance where neighbors are taken into

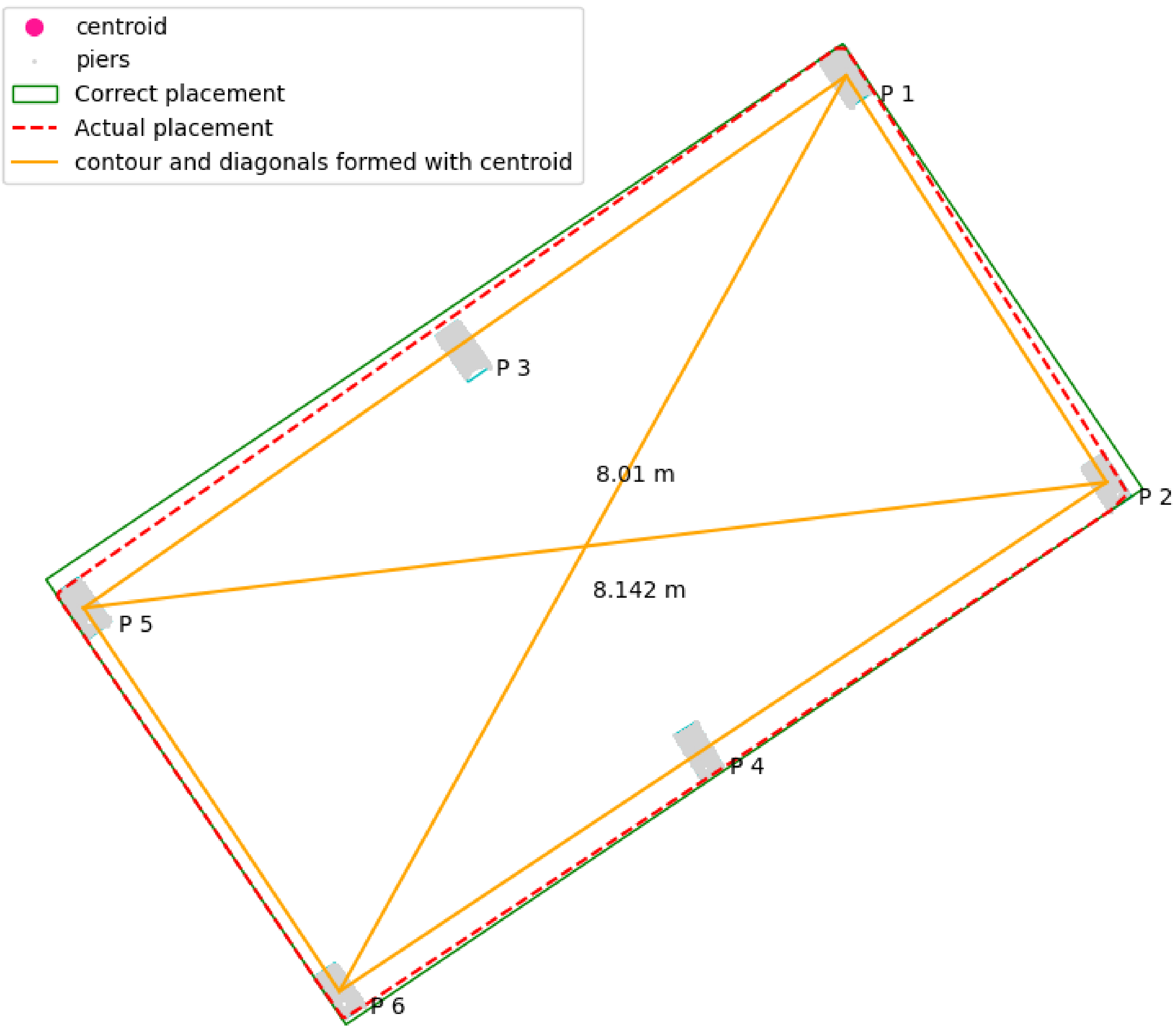

Figure 4. Planar alignment guidance for pier locations

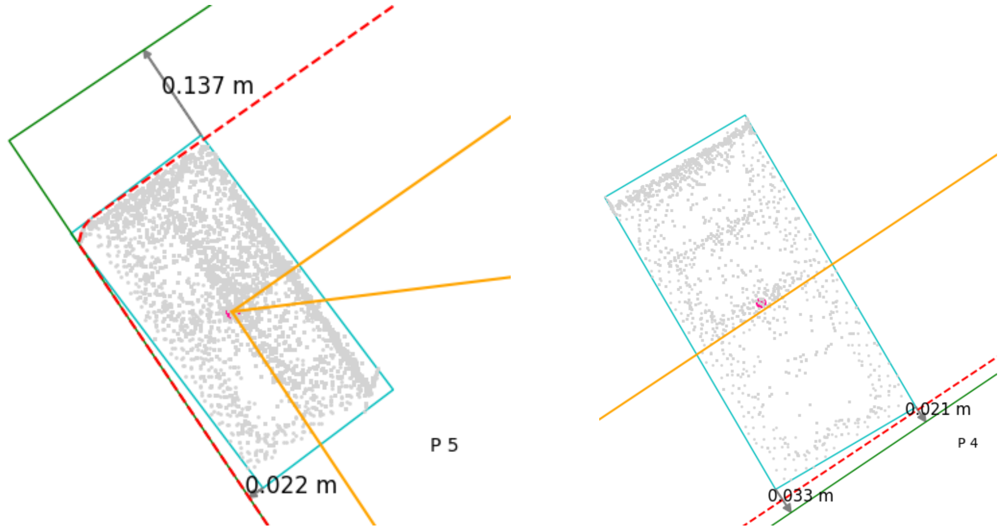

Figure 5. Local planar correction to locate each pier

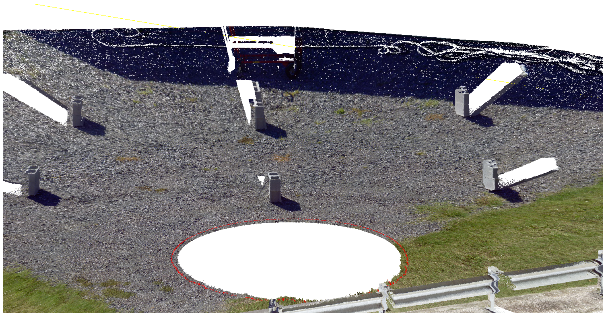

Figure 6. A portion of the full-dome scan of the site with mockup pier foundations.

account.

Table 1. Parameters for the demonstration.

Parameter	Value
Voxel Size	12.5 cm
Plane Distance Threshold	20 cm
Kd-Tree Search Radius	20 cm
Kd-Tree Minimum Points	1000
Cluster Minimum Points	25
Cluster Neighbor Radius	50 cm

For the MS60 laser scanner used in this study, the single-view dense point cloud with a specified resolution was collected in approximately 45 minutes. However, many TLS scanners boast faster scanning rates and can capture point cloud data with higher resolution in a fraction of the time (3 minutes or less).

Table 2 gathers the running time for each main task required to segment and analyze the piers foundation on a standard laptop computer. Tasks before segmentation included the PCD file reading, trimming, and pre-processing with RANSAC, voxel-down sampling, and KD-Tree. The clustering by density-based scan (DBSCAN) and the cluster filtering are reported collectively within the segmentation task. Finally elevation and alignment guidance tasks represent the computation and analysis of piers elevation and placement.

The mean total run time was approximately 27.20 seconds. Most tasks within the algorithms have consistent run time except for pre-processing and segmentation. This is due to the random nature of RANSAC. Since the tests were not conducted on a flat surface, some parts of the ground may not be removed and taken into account during the segmentation which increases the computation time. Another method to identify the ground may be introduced in the future in order to address this issue.

Among 40 iterations, two runs were considered failures. One failure was due to RANSAC removing the entire collection of points belonging to a single pier. The other failure consisted of an additional element in the construction site being incorrectly identified as a pier. This variability is mainly due to the ground removal with RANSAC and the clustering method that use unsupervised machine learning, resulting in different cluster identifications between attempts. Another cause of failure is if there is an object on site that has approximately the same dimension as the piers since the algorithm is filtering the different identified clusters based on the dimension specified by the user.

Table 2. Run times. 40 iterations.

Task	Run Times (sec)			
	Min	Max	Mean	Std
Read PCD	3.77	4.11	3.89	0.09
Trim	0.01	0.03	0.01	0.01
Pre processing	4.27	5.80	5.10	0.36
Segmentation	13.06	25.25	16.33	3.08
Elevation	0.21	0.38	0.28	0.04
Alignment	1.78	1.04	1.4	0.15
Entire Procedure	23.47	36.52	27.20	3.14

4 Conclusions and next steps

Pier Inspection and Evaluation Report (PIER) is a digital tool that uses algorithms to automate the analysis and inspection of point clouds of pier foundations. Tests were conducted on masonry piers at Oak Ridge National Lab site. The results show that PIER can provide quick and accurate feedback on piers installation.

In future work, an alternative way to remove the ground to reduce the variability of the segmentation will be implemented. The selection algorithm for identifying point clusters of piers will be improved to increase robustness. Futhermore, a feature to extract and report on the levelness of the top surface of each pier will be added to the algorithms.

Additionally, future work will include research on the robustness and accuracy of the proposed methods through multiple demonstrations at construction sites where piers are being constructed. PIER will be demonstrated and verified to produce accurate results and guidance.

5 Acknowledgements

This research was supported by the DOE Office of Energy Efficiency and Renewable Energy, Building Technologies Office, under the guidance of Sven Mumme, and used resources at the Building Technologies Research and Integration Center, a DOE-EERE User Facility at Oak Ridge National Laboratory.

References

[1] IRC 2024. International Residential Code (IRC). Building code, International Code Council, 2024.

[2] ACI SPEC-336.1-01. Specification for the Construction of Drilled Piers. Standard, American Concrete Institute, 2001.

[3] ACI SPEC-336.1-24. Construction of Drilled Piers - Specification. Standard, American Concrete Institute, 2024.

[4] Tao Wu, Shitong Hou, Weihao Sun, Jing Shi, Fujun Yang, Jian Zhang, Gang Wu, and Xiaoyuan He. Visual measurement method for three-dimensional shape of underwater bridge piers considering multirefraction correction. *Automation in Construction*, 146:104706, 2023. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2022.104706.

[5] Associated General Contractors. New Survey Shows How Failure To Invest In Construction Education Makes It Hard For Firms To Build - AGC News — news.agc.org. https://news.agc.org/labor-hr/new-survey-shows-how-nations-failure-to-invest-in-construction-education-training-programs-makes-it-hard-for-firms-to-build/, 2024. [Accessed 16-12-2024].

[6] Nolan W Hayes, Bryan P Maldonado, Mengjia Tang, and Diana Hun. Flat and Level Analysis Tool (FLAT) for real-time automated segmentation and analysis of concrete slab point clouds. In *Proceedings of the 41st International Symposium on Automation and Robotics in Construction*, pages 838–846, Lille, France, June 2024. International Association for Automation and Robotics in Construction (IAARC). ISBN 978-0-6458322-1-1. doi:10.22260/ISARC2024/0109.

[7] Shuangping Li, Bin Zhang, Junxing Zheng, Dong Wang, and Zuqiang Liu. Development of automated 3d lidar system for dimensional quality inspection of prefabricated concrete elements. *Sensors*, 24(23), 2024. ISSN 1424-8220. doi:10.3390/s24237486.

[8] Yan Xu, Yi Luo, and Jian Zhang. Laser-scan based pose monitoring for guiding erection of precast concrete bridge piers. *Automation in Construction*, 140:104347, 2022. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2022.104347.

[9] Min-Koo Kim et al. Automated dimensional quality assessment for formwork and rebar of reinforced concrete components using 3d point cloud data. *Automation in Construction*, 112:103077, 2020.

[10] Mads A Kuhlmann-Jørgensen et al. Plastr: Planning for autonomous sampling-based trowelling. *IEEE Robotics and Automation Letters*, 8(8):5069–5076, 2023.

[11] Pingbo Tang et al. 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, 2010.

[12] Bryan P. Maldonado et al. Automatic point Cloud Building Envelope Segmentation (Auto-CuBES) using Machine Learning. In *Proceedings of the 40th International Symposium on Automation and Robotics in Construction*, pages 48–55, Chennai, India, 2023.

[13] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. doi:10.1145/358669.358692.

[14] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[15] C. Bradford Barber, David P. Dobkin, and Hannu T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996. doi:10.1145/235815.235821. URL `http://www.qhull.org`.

[16] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. doi:10.1145/361002.361007.

[17] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, 1996.

[18] Joseph O'Rourke. Finding minimal enclosing boxes. *International Journal of Computer & Information Sciences*, 14(3):183–199, Jun 1985. ISSN 1573-7640. doi:10.1007/BF00991005.

[19] Kevin Bellock. alphashape: A python implementation of alpha shapes. `https://github.com/bellockk/alphashape`, n.d.