Automated Quality Inspection of Precast Concrete Elements with Irregular Shapes Using Terrestrial Laser Scanner and BIM Technology

Q. Wang[a,b], J. C. P. Cheng[a] and H. Sohn[b]

[a]Department of Civil and Environmental Engineering,
The Hong Kong University of Science and Technology, Hong Kong
[b]Department of Civil and Environmental Engineering,
Korea Advanced Institute of Science and Technology, South Korea
E-mail: qwangau@ust.hk, cejcheng@ust.hk, hoonsohn@kaist.ac.kr

ABSTRACT

Nowadays precast concrete elements are widely used in buildings, bridges, and other civil infrastructure facilities because precast elements allow rapid construction and high precision quality control. However, the quality inspection of precast concrete elements primarily relies on manual inspection, which is time consuming and error-prone. Recently, terrestrial laser scanner (TLS) has been used to improve quality inspection. The authors' group has previously developed an automated dimension estimation technique for precast concrete elements, but its applicability is limited only to precast elements with rectangular shapes. This study advances our previous work so that the dimensions of precast elements with irregular shapes can also be automatically estimated. First, a density-based clustering algorithm is adopted to extract target objects from 3D point cloud data acquired by a TLS. Then, coarse registration is conducted to match each object extracted from the 3D point cloud with the as-design objects in building information models (BIM) one by one. Thirdly, all point cloud data points are registered onto different surfaces of the as-designed objects in BIM through fine registration. Lastly, the as-built dimensions of the precast concrete element are extracted and compared with the as-design ones in BIM. The effectiveness and accuracy of the developed technique are examined using point cloud data obtained from a laboratory-scale precast concrete bridge deck panel.

Keywords –

Automated Dimension Estimation; Precast Concrete Element; Terrestrial Laser Scanner; Building Information Model; Irregular Shape

1 Introduction

Nowadays precast concrete elements are widely used in buildings, bridges, and other civil infrastructure facilities because precast elements allow rapid construction and high precision quality control [1,2]. However, the quality inspection of precast concrete elements primarily relies on manual inspection, which is time consuming and error-prone [3]. Recently, terrestrial laser scanner (TLS) has been used to improve quality inspection due to its high accuracy of range measurement and high speed of data acquisition. For example, Teza et al. [4] developed a method for automatic recognition of mass loss of concrete using point cloud data from TLS. Kim et al. [5] reported a technique to simultaneously localize and quantify spalling defects on concrete surfaces. Schäfer et al. [6] measured dynamic deformation of a concrete wall in hydropower station under different settings of constant liquid levels using TLS. Moreover, several studies have integrated laser scanning with building information models (BIM). As-built geometry of a facility represented in point clouds from TLS can be compared with the as-design geometry represented in BIM to identify any discrepancy in the built facility. Bosché [7] developed an algorithm to automatically recognize 3D CAD objects in a point cloud and calculate as-built dimensions of objects for dimensional compliance control. Tang et al. [8] formulated a deviation analysis based approach to compare BIM with laser scanning data for efficient quality assessment.

The authors' group has previously developed an automated dimension estimation technique for precast concrete elements, but its applicability is limited only to precast elements with rectangular shapes [9]. This study advances our previous work so that the dimensions of precast elements with irregular shapes, particularly sides of precast concrete bridge deck panels, can also be automatically estimated. Sides of precast concrete bridge

deck panels usually have a complex polygonal outer boundary and two important structures, namely shear keys and flat ducts, as shown in Figure 1. Shear keys serve as transverse joints between two adjacent panels. For a typical female-female shear key joint, grout is filled in the space between two shear keys of two adjacent panels so that the joint can transfer traffic load from one panel to the next [10]. Flat ducts are used to place post-tensioning strands, which also connect two adjacent panels. Therefore, it is important to make sure that these structures have desired dimensions and positions in accordance with design codes.

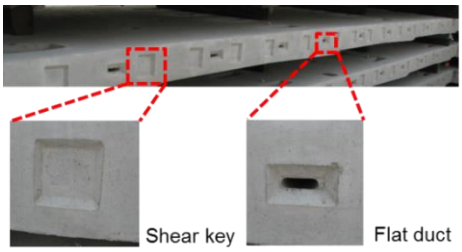

Figure 1. Examples of shear keys and flat ducts on sides of precast concrete bridge deck panels

In this study, an automated dimensional quality inspection technique for precast concrete panels using TLS and BIM technology is developed. First, a density-based clustering algorithm is adopted to extract target objects from 3D point cloud data acquired by a TLS. Then, coarse registration is conducted to match each object extracted from the 3D point cloud with the as-design objects in BIM one by one. Thirdly, all point cloud data points are registered onto different surfaces of the as-designed objects in BIM through fine registration. Lastly, the as-built dimensions of the precast concrete element are extracted and compared with the as-design ones in BIM. The effectiveness and accuracy of the developed technique are examined using point cloud data obtained from a laboratory-scale precast concrete bridge deck panel.

2 Proposed Automated Dimensional Quality Inspection Technique

An overview of the proposed automated dimensional quality inspection technique is described in Figure 2. Point cloud data of a precast concrete panel obtained from TLS and the as-design BIM of the panel are provided as input information. Point cloud data are processed through four steps – (1) noise removal, (2) coarse registration, (3) fine registration, and (4) dimension extraction, while the as-design BIM serves as the reference in the steps of coarse registration and fine registration.

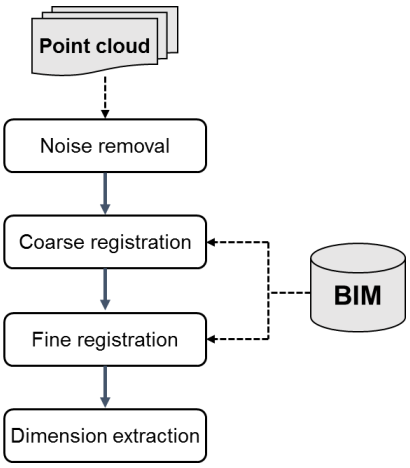

Figure 2. Overview of the proposed automated dimensional quality inspection technique

2.1 Noise Removal

Once point cloud data are acquired by a TLS, noise removal is undertaken to remove noise data and retain valid points representing the target object to facilitate subsequent processing. Figure 3 illustrates three kinds of data contained in a point cloud, in which point $(i-2)$ and point $(i-1)$ are valid points, point $(i+1)$ and point $(i+2)$ are background points, and point i is a mixed pixel. Background points and mixed pixels are two kinds of noise data. A background point occurs when a laser beam fully lies on the background rather than the target object. A mixed pixel occurs when a laser beam is separated into two parts and lies on two different surfaces, target object and background [11]. Since both of the reflected signals from two surfaces are received by TLS, the resulted pixel will be located between these two surfaces. Therefore, a mixed pixel usually has a larger distance to its adjacent point than a valid point. For example, $D_{i-1,i}$ is larger than $D_{i-2,i-1}$ and $D_{i,i+1}$ is larger than $D_{i+1,i+2}$, given that $D_{i,j}$ represents the distance between point i and point j.

Since valid points and background points have relatively high density while mixed pixels between them have low density, a density-based clustering algorithm namely Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is adopted in this study to remove the background points and mixed pixels from point cloud data [12]. DBSCAN is based on two parameters (ε and *minPts*) and three fundamental concepts (*core point*, *directly density-reachable*, and *density-reachable*). The three fundamental concepts are built on the parameters ε and *minPts*. A point p is a *core point* if there are at least *minPts* points surrounding p at a distance of ε or less from p. A point q is *directly density-reachable* from p if p is a *core point* and the distance between q and p is less

than or equal to ε. Point q is *density-reachable* from p if there exists a sequence $\{p_1, p_2 \ldots p_n\}$ of points with $p_1 = p$ and $p_n = q$, where each p_{i+1} is *directly density-reachable* from p_i. The DBSCAN algorithm visits each point in a dataset. If one point is a *core point*, a new cluster is started and all the points which are *density-reachable* from the *core point* as well as the *core point* itself are included in the cluster. If one point is not a *core point* and it is not *density-reachable* from any *core point*, it is labelled as noise.

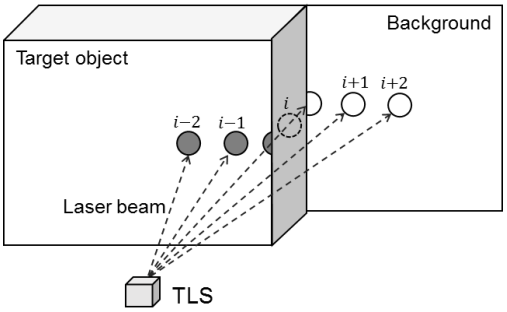

Figure 3. Three kinds of data contained in a point cloud – valid points, background points, and mixed pixels

Figure 4 shows an example of point cloud data, where filled circles represent valid points, empty solid circles represent background points, and empty dashed circles represent mixed pixels. Assume that all the valid points and background points are evenly distributed respectively and they have the same distance, which is denoted as d_0, between two adjacent points. DBSCAN is applied to this point cloud with *minPts* = 8 and $\varepsilon = d_1$, which is the distance between two adjacent valid points in diagonal direction (e.g. point A and point B as shown in Figure 4). As a result, point A becomes a *core point* because it has 8 surrounding points which have a distance smaller than or equal to ε, so that a cluster is started which will include all the valid points finally. Since point D is a mixed pixel, the distance between point C and point D, denoted as d_2, is larger than ε. Therefore, point D is not *density-reachable* from A and it is not included in this cluster. After applying DBSCAN, all the valid points are included in one cluster; all the background points are included in another cluster; and mixed pixels are labelled as noise.

However, the assumption that valid points are evenly distributed in aforementioned example is not correct. In reality, distance between two adjacent points d_0 depends on the distance from TLS to target object and the incident angle between laser beam and the normal vector of target object surface. Therefore, ε should not be smaller than d_{max}, which denotes the theoretical maximum value of the distance between any two adjacent valid points in diagonal direction. Moreover, considering that the target object surface may not be smooth, d_0 may vary according to local surface condition. Therefore, a safety factor is taken into consideration and ε is set as $1.2d_{max}$.

Once all clusters are found in a point cloud, the cluster of points that have the smallest distance to the TLS are taken as valid points and represent the target object, whereas background points are farther away from the TLS.

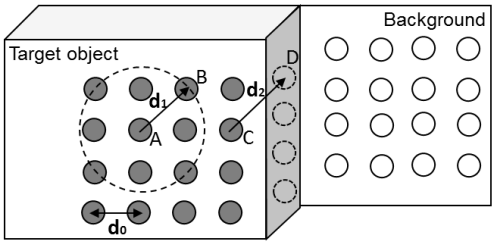

Figure 4. Noise removal based on DBSCAN

2.2 Coarse Registration

Once the as-built target object is extracted from the point cloud, coarse registration is undertaken to transform the as-built object so that it best matches its corresponding as-design object in BIM. The as-built object is transformed by a three-dimensional (3D) transformation, which includes three translations and three rotations, in order to minimize the distance between the as-built and as-design objects. Note that the as-built object is expressed by a point cloud and the as-design object is expressed by a set of surfaces in BIM.

Considering that the target object in this study is the side of a precast concrete bridge deck panel, which is a quasi-2D object although there are structures like shear keys, the problem of finding a 3D transformation can be simplified to finding a 2D transformation as follows. (1) Find the least squares fitting planes of the as-built and as-design objects. Since the as-design object is expressed by a set of surfaces rather than a point cloud, sampling points are generated on all the surfaces with identical sampling density and the least squares fitting plane of all the sampling points is taken as the fitting plane of the as-design object. (2) Transform the as-built object so that its fitting plane is overlapping with that of the as-design object. Their common fitting plane is then defined as the x-y plane of a new Cartesian coordinate system and two objects are both transformed into this new coordinate system. Therefore, the problem is simplified to finding a 2D transformation in the x-y plane which minimizes the distance between as-built and as-design objects. This 2D transformation is a combination of translation along x axis, translation along y axis, and rotation about z axis.

To find the 2D transformation for best matching the as-built and as-design geometries, an iterative search

algorithm is used, as described in the followings. (1) Project the as-built and as-design objects onto the x-y plane. Then the following steps only involve x and y coordinates. (2) Extract the boundaries of the as-built and as-design objects. Here, the boundary of the as-built object refers to a set of points, which are classified as valid points but not *core points* in the DBSCAN algorithm described in previous section. The boundary of the as-design object refers to a set of line segments, which compose the outer boundary of the as-design object. Figure 5(a) shows an example, where circles represent an as-built boundary and line segments represent an as-design boundary. (3) Start the iterative search algorithm. In each iteration, generate a different 2D transformation and transform the as-built boundary by this transformation. After that, for each point in the as-built boundary, find the least distance from it to the as-design boundary. For example, the least distance from point A, as shown in Figure 5(a), to the as-design boundary is the perpendicular distance from it to the line segment located below it. (4) After finding the least distance for each point in the as-built boundary, calculate the mean square of all the distances. When the mean square is minimized, as shown in Figure 5(b), the transformation for the best matching is found. Then, the as-built object is transformed by this transformation in order to best match the as-design object.

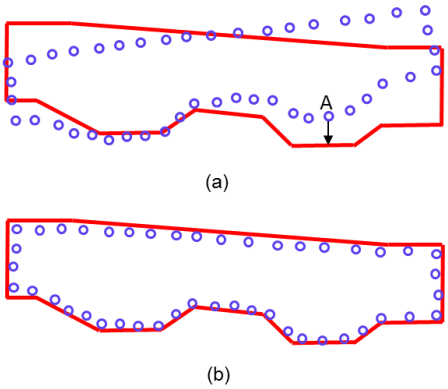

Figure 5. Coarse registration: (a) The as-built (circles) and as-design (line segments) boundaries; (b) Transformation for the best matching

2.3 Fine Registration

In fine registration, all point cloud data points of the as-built object are registered onto different surfaces of the as-design object. Fine registration mainly works for shear keys in precast concrete bridge deck panels because each shear key has multiple surfaces. The 3D view and plane view of a typical shear key are shown in Figure 6(a) and 6(b), respectively. There are 6 surfaces in the local area of a shear key as labeled in Figure 6(b).

For a point in the point cloud of the as-built object, it is impossible to be registered onto any as-design surface which is too far away from it. Thus, a margin is firstly set for each surface and only points inside the margin of a surface are possible to be registered onto this surface. As shown in Figure 6(c), the solid lines represent the margin of surface 1; the dashed lines represent the margin of surface 5; and the margins of other surfaces are set in a similar manner. If one point is inside the margin of only one surface, e.g. point A as shown in Figure 6(d), it is registered onto this surface. If one point is inside the margins of more than one surfaces, e.g. point B, additional criteria are needed to decide which surface this point should be registered onto.

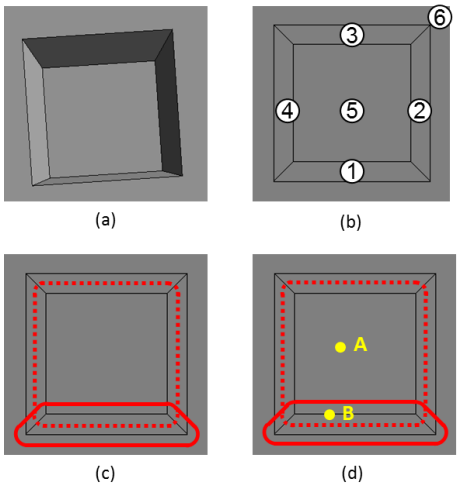

Figure 6. Fine registration for points on a shear key: (a) 3D view of a shear key; (b) Plane view of a shear key and 6 surfaces in the local area of it; (c) Margins of surface 1 and surface 5; (d) Examples of two kinds of points

The first criterion is the perpendicular distance from the point to each surface. As shown in Figure 7(a), point B is possible to be registered onto surface 1 and surface 5. The perpendicular distances from point B to two surfaces are calculated and denoted as D_{B1} and D_{B5}, respectively. The second criterion is the difference between the normal vector of the point and the normal vector of each surface. Normal vector of a point refers to the normal vector of its local least squares fitting plane. As shown in Figure 7(b), eight nearest neighbors of point B are extracted. Then, the least squares fitting plane of totally nine points is found and its normal vector n_B is taken as the normal vector of point B. Besides, the normal vectors of surface 1 and surface 5 are denoted as n_1 and n_5, respectively. Assume that the three vectors are all unit vectors, $|n_B - n_1|$ and $|n_B - n_5|$ represent the magnitudes of differences between the normal vector of

point B and the normal vectors of the two surfaces, respectively.

Here, a double-criteria principle is adopted to improve the accuracy of fine registration. Among all the surfaces onto which one point is possible to be registered, only if one surface has both the smallest distance to the point and the smallest magnitude of difference between normal vectors, the point is registered onto this surface. Otherwise, the point is not registered onto any surface. For point B, if $D_{B1} < D_{B5}$ and $|n_B - n_1| < |n_B - n_5|$, point B is registered onto surface 1; if $D_{B1} > D_{B5}$ and $|n_B - n_1| > |n_B - n_5|$, point B is registered onto surface 5; otherwise, point B is not registered onto any surface.

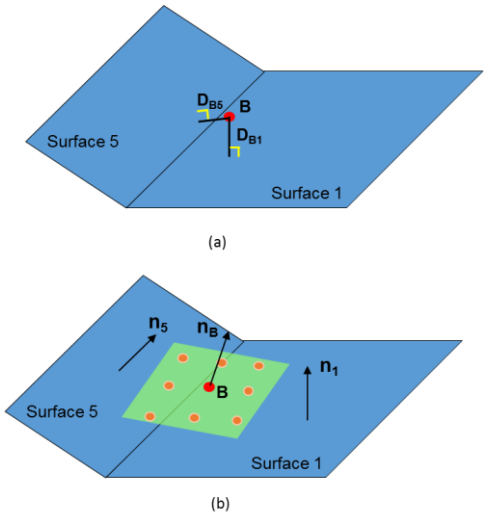

Figure 7. Two criteria for fine registration: (a) Distances from point B to two surfaces; (b) Normal vectors of point B and two surfaces

2.4 Dimension Extraction

To extract the dimensions of the as-built object, the coordinates of as-built corner points need to be obtained firstly. Usually, for a 3D object, every corner point can be extracted as the intersection point of three or more surfaces, which is called intersection point based approach. However, in this study, point cloud data are only available for one side of a precast concrete bridge deck panel. Point cloud data of other sides are not available, and therefore corner points on the outer boundary of a panel cannot be accurately identified by intersecting multiple surfaces. In this case, an edge point based approach is adopted to extract the coordinates of as-built corner points on the outer boundary.

2.4.1 Intersection Point Based Approach

Intersection point based approach works in the case that one corner point is the intersection point of multiple surfaces whose point cloud data are all available, e.g. corner points of shear keys. Figure 8 shows the local area of a typical shear key, which has 6 surface labelled from 1 to 6, and 8 corner points labelled from A to H. In the step of fine registration, all point cloud data points of the as-built object are registered onto different surfaces of the as-design object. Therefore, the position of each as-built surface can be obtained as the least squares fitting plane of all the points that are registered onto their corresponding as-design surfaces. Then, each as-built corner point can be extracted as the intersection point of three as-built surfaces. For example, corner point A is the intersection point of surfaces 1, 4 and 6.

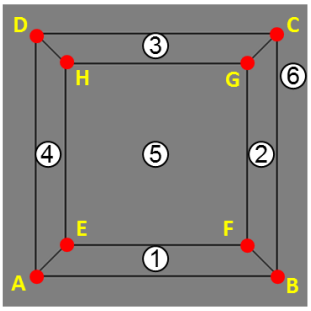

Figure 8. Intersection point based approach for corner points of a shear key

2.4.2 Edge Point Based Approach

Edge point based approach is needed when point cloud data of the surfaces that intersect at one corner point are not fully available, e.g. corner points on the outer boundary of a panel. Figure 9 illustrates the edge point based approach, where L_{11} and L_{12} are two as-design edges and their intersection point A_1 is an as-design corner point. Circles represent a point cloud of the target object and filled circles are extracted as edge points because they are the last valid point in each row or column. Then, the least squares fitting lines of edge points for each edge, denoted as L_{21} and L_{22} respectively, are obtained and taken as as-built edges. Finally, as-built corner point A_2 is obtained as the intersection point of two as-built edges.

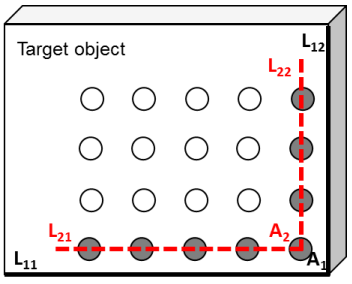

Figure 9. Edge point based approach

However, as-built edges identified as described above are probably not the actual edges. Point cloud data shown in Figure 9 are all valid points, which are fully inside the target object. The points that lie at edges are removed as noise because they become mixed pixels in the point cloud. So, there must be a difference between the found edges and the actual edges, which is called edge loss. Tang et al. [13] developed an edge loss compensation model but it is only applicable to horizontal and vertical edges. To provide a generic solution to edge loss problem, an edge estimation algorithm is developed in this study.

Figure 10 shows a simple case where the investigated edge, i.e. the right-hand edge of the target object, is vertical. Empty solid circles and filled solid circles represent valid points, which are fully inside the target object. Filled solid circles represent edge points because they are the last valid points in each row. However, true edge cannot be located at the least squares fitting line of these edge points, denoted as L0, because edge points should be fully inside the target object. Instead, true edge must be to the right of L1, which is obtained by translating L0 along the direction of its normal vector $\overrightarrow{n_1}$ by the radius of a laser beam (denoted as r), to ensure that all edge points are fully inside the target object. L1 is called the lower bound of true edge. Next, consider the points which are removed as noise, i.e. mixed pixels and background points. Dashed circles represent a group of virtual points just next to edge points, which are created assuming that the distance between two adjacent points keeps the same. However, these virtual points do not exist in the post-processed point cloud because they become mixed pixels or background points. It can be inferred that these virtual points are partially or fully outside the target object. Therefore, the true edge must be to the left of L2, which is obtained by finding the least squares fitting line of all the virtual points and translating it along $\overrightarrow{n_1}$ by r. L2 is called the upper bound of true edge. To conclude, true edge must be between lower bound L1 and upper bound L2, and the center line of them is taken as the estimated edge.

With the same principle, a generic edge with any orientation, as shown in Figure 11, can be estimated as follows. (1) Find the last valid point in each row or column as edge points, which are shown as filled circles. Then find the least squares fitting line of these edge points and translate it along its normal vector $\overrightarrow{n_2}$ by r, so that L3 is obtained as the lower bound of true edge. Note that the direction of $\overrightarrow{n_2}$ is always outward from the target object. (2) For each edge point, create a virtual point just next to it assuming that the distance between two adjacent points keeps the same, which are shown as dashed circles. Find the least squares fitting line of these virtual points and translate it along $\overrightarrow{n_2}$ by r, so that L4 is obtained as the upper bound of true edge. (3) The center line of lower bound L3 and upper bound L4 is obtained as the estimated edge shown in dashed line.

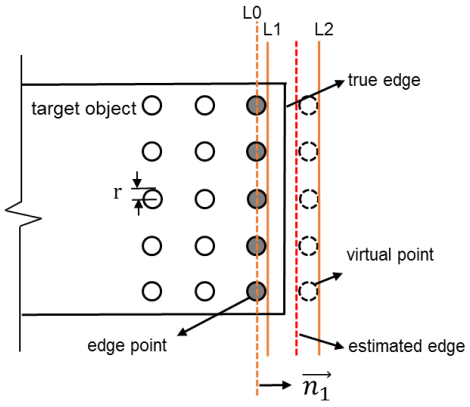

Figure 10. Edge estimation for a vertical edge

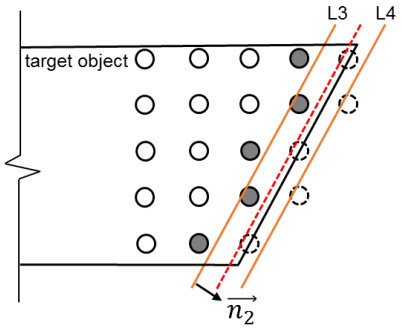

Figure 11. Edge estimation for a generic edge

Once as-built edges are obtained from the as-built object using the proposed edge estimation algorithm, the coordinates of as-built corner points can be extracted as the intersection points of as-built edges. Further, as-built dimensions can be easily calculated from the coordinates of as-built corner points.

3 Validation

The effectiveness and accuracy of the proposed automated dimensional quality inspection technique were examined using the point cloud obtained from a laboratory-scale specimen of precast concrete bridge deck panels.

3.1 Experimental Specimen and Setup

Figure 12 shows the experimental specimen and setup. The specimen has a dimension of around 700 mm × 120 mm and it has four identical shear keys with outer dimension of 80 mm × 80 mm, inner dimension of 60 mm × 60 mm, and depth of 20 mm. The point cloud data of the specimen were acquired by a FARO Focus 3D TLS, which provided a range accuracy of ±2 mm at scanning

distance of 25 m [14]. The distance from the TLS to the specimen was 4 meters and the angular resolution was 0.018°.

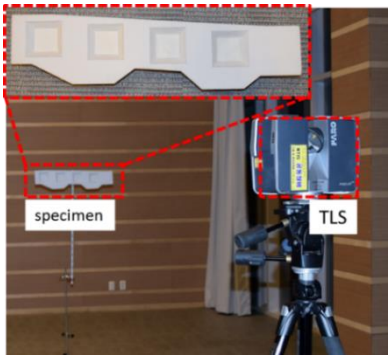

Figure 12. Experimental specimen and setup

3.2 Data Processing

Figure 13 shows the four steps of data processing for point cloud data obtained from the specimen.

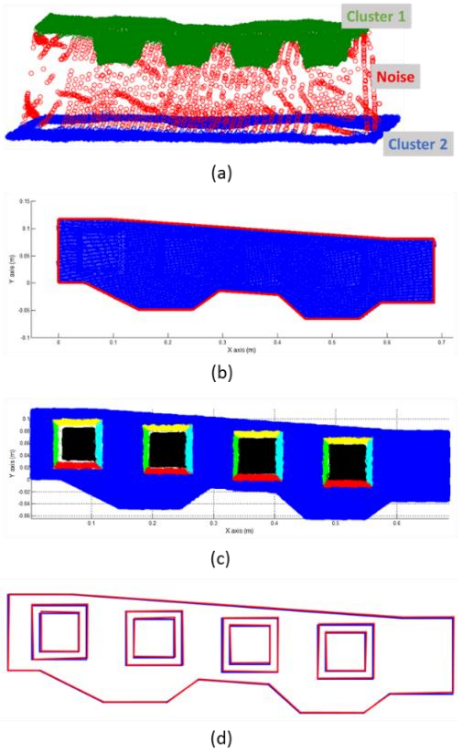

Figure 13. Data processing for the point cloud data from the specimen: (a) Noise removal using DBSCAN; (b) Result of coarse registration; (c) Result of fine registration; (d) As-design and as-built edges

Figure 13(a) shows the result of noise removal using DBSCAN, where valid points shown in green are included in one cluster, background points shown in blue are included in another cluster, and mixed pixels shown in red are labelled as noise. Figure 13(b) shows the result of coarse registration, where red lines represent the outer boundary of the as-design object and the as-built object shown in blue matches well with it after coarse registration. Figure 13(c) shows the result of fine registration, where points registered onto the same surface are shown in the same color. Lastly, Figure 13(d) shows as-built edges in blue and as-design edges in red.

3.3 Dimensional Quality Inspection Results

The experimental specimen had 14 corner points along the outer boundary and 8 corner points for each shear key. The distances between all 46 as-design corner points and their corresponding as-built corner points were calculated one by one. The result showed that the average distance between the as-built and as-design corner points was 0.95 mm, and the maximum distance was 2.95 mm.

Furthermore, the specimen had 14 sides along the outer boundary and 8 sides for each shear key, including outer and inner boundaries. The as-design and as-built dimensions of all 46 sides were calculated from the coordinates of as-design and as-built corner points, respectively. The discrepancies between all as-design dimensions and their corresponding as-built dimensions were calculated one by one. The result showed that the average discrepancy in dimensions was 0.59 mm (2.8%), and the maximum discrepancy was 3.4 mm (7.2%).

4 Conclusion

This study presents an automated dimensional quality inspection technique for precast concrete elements with irregular shape using TLS and BIM technology. First, a density-based clustering algorithm is adopted to extract target objects from 3D point cloud data acquired by TLS. Then, coarse registration is conducted to match each as-built object extracted from the 3D point cloud with the as-design objects from the BIM one by one. Third, all point cloud data points are registered onto different surfaces of the as-design objects in BIM through fine registration. Lastly, the as-built dimensions of the precast concrete element are extracted and compared with the as-design ones in BIM.

The effectiveness and accuracy of the proposed technique were examined using point cloud data obtained from a laboratory-scale precast concrete bridge deck panel. It shows that, the average distance between the as-built and as-design corner points was 0.95 mm, whereas the average discrepancy between the as-built and as-design dimensions was 0.59 mm. The proposed technique is proved to be effective and accurate.

However, this study has two limitations. First, the proposed technique was evaluated using a laboratory-scale precast concrete panel only, and needs to be further examined using full-scale precast concrete elements that are larger in size. Second, the background condition in laboratory setting was plain while the background condition in real-world setting could be complicated. Further evaluation of the proposed technique under real-world setting will be conducted in the future.

Acknowledgement

This research was supported by a grant (13SCIPA01) from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement (KAIA).

References

[1] Glass J. *The future for precast concrete in low-rise housing*. Leicester: British Precast Concrete Federation, 2000.

[2] Blais P. Y. and Couture M. Precast, prestressed pedestrian bridge-world's first reactive powder concrete structure. *PCI journal*, 44:60-71, 1999.

[3] Phares B. M., Washer G. A., Rolander D. D., Graybeal B. A. and Moore M. "Routine highway bridge inspection condition documentation accuracy and reliability." *Journal of Bridge Engineering*, 9(4):403-413, 2004.

[4] Teza G., Galgaro A. and Moro F. Contactless recognition of concrete surface damage from laser scanning and curvature computation. *NDT & E International*, 42(4):240-249, 2009.

[5] Kim M. K., Sohn H. and Chang C. C. Localization and Quantification of Concrete Spalling Defects Using Terrestrial Laser Scanning. *Journal of Computing in Civil Engineering*, 2014.

[6] Schäfer T., Weber T., Kyrinovič P. and Zámečniková M. Deformation measurement using terrestrial laser scanning at the hydropower station of Gabčíkovo. In *INGEO 2004 and FIG Regional Central and Eastern European Conference on Engineering Surveying*, Bratislava, Slovakia, 2004.

[7] Bosché F. 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*, 24(1):107-118, 2010.

[8] Tang P., Anil E., Akinci B. and Huber D. Efficient and effective quality assessment of as-is building information models and 3D laser-scanned data. In *Proceedings of the ASCE International Workshop on Computing in Civil Engineering*, Miami, USA, 2011.

[9] Kim M. K., Sohn H. and Chang C. C. Automated dimensional quality assessment of precast concrete panels using terrestrial laser scanning. *Automation in Construction*, 45:163-177, 2014.

[10] Wacker J. M., Eberhard M. O. and Stanton J. F. *State-of-the-art report on precast concrete systems for rapid construction of bridges* (No. WA-RD 594.1). Washington State Department of Transportation, 2005.

[11] Hebert M. and Krotkov E. 3-D measurements from imaging laser radars: How good are they?. In Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on, pages 359–364, Osaka, Japan, 1991.

[12] Ester M., Kriegel H. P., Sander J. and Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, No. 34, pp. 226-231), Portland, USA, 1996.

[13] Tang P., Akinci B. and Huber D. Quantification of edge loss of laser scanned data at spatial discontinuities. *Automation in Construction*, 18(8):1070-1083, 2009.

[14] FARO Technologies Inc. FARO Laser Scanner Focus 3D. On-line: http://www.faro.com/en-us/products/3d-surveying/faro-focus3d/overview, Accessed: 25/01/2015.