



An interactive surface damage measurement system using 3D LiDAR

Heon-Hui Kim[†] · Dong-Hyun Bae¹ · Taek-Kun Nam² · Jin-Kyu Choi³

(Received January 27, 2026 ; Revised February 12, 2026 ; Accepted February 12, 2026)

Abstract: Surface damage such as dents and deformations on ship hulls directly affects structural safety, making accurate measurement essential. Conventional contact-based measurement methods are difficult to apply to large structures and are time-consuming. This study proposes a real-time surface damage measurement system using structured LiDAR point cloud data. The proposed system consists of a four-stage pipeline: preprocessing, curvature extraction, plane estimation, and damage measurement. In the preprocessing stage, sensor noise is suppressed through hole filling, Gaussian smoothing, and EMA filtering. In the curvature extraction stage, damage candidate regions are detected through PCA-based surface variation analysis. In the plane estimation stage, the reference plane is estimated using reference normal-based filtering and RANSAC, and in the damage measurement stage, robust damage quantification is performed using a percentile-based approach. Experimental results demonstrate that the proposed system achieves a measurement accuracy of 0.64 mm RMS error and a repeatability within 0.7 mm, with an average processing time of 68.7 ms enabling near real-time measurement.

Keywords: 3D LiDAR, Point cloud processing, Surface damage measurement, Curvature extraction, Plane estimation, Non-contact inspection

1. Introduction

The outer shells of large structures such as ships, aircraft, and storage tanks are exposed to surface damage in the form of protrusions or dents due to physical impacts or contact during operation. In the case of ships, contact with piers during berthing, collisions with floating objects during navigation, and careless operation of cargo handling equipment are the main causes [1], and the resulting structural deformation has been studied in terms of dynamic crushing behavior under impact loads such as ship-to-ship collisions and ship-to-bridge impacts [2]. Aircraft also experience localized deformation due to similar external impacts [3]. Large storage tanks are likewise susceptible to surface dents and protrusions resulting from mechanical contact during filling, maintenance, and logistical operations. Such surface damage can cause long-term corrosion and reduce structural strength by accompanying coating damage, so it must be properly managed through regular inspections.

Damage inspection of structures has two objectives: detection to identify the presence of damage, and quantitative measurement to calculate the depth and volume of damage numerically. Detection is

used to determine the need for follow-up actions, and quantitative measurement provides essential information for evaluating the severity of damage and determining the scope of repairs [4]. In this regard, image processing techniques have been highlighted as an effective means of enabling precise and objective quantitative assessment of surface deterioration such as corrosion [5]. Therefore, both objectives of detection and quantitative measurement must be simultaneously satisfied for effective maintenance decision-making.

Existing damage inspection methods each have their own limitations. Visual inspection is performed as the first step before other non-destructive testing techniques and has the advantages of being cost-effective and easy to apply [6]. However, inspection results are subjective as they depend heavily on the inspector's experience and proficiency, and quantitative measurement of damage depth or volume is inherently impossible [7]. Manual measurement tools such as pit gauges or ultrasonic thickness gauges enable high-precision quantitative measurement, but the point-by-point measurement method requires considerable time to inspect the entire large structure. Terrestrial Laser Scanning

[†] Corresponding Author (ORCID: <https://orcid.org/0000-0003-3410-8493>): Associate Professor, Division of Marine System Engineering, Mokpo National Maritime University, 91, Haeyangdaehak-ro, Mokpo-si, Jeollanam-do 58628, Korea, E-mail: heonhuikim@mmu.ac.kr, Tel: +82-61-240-7256

¹ M. S., Department of Marine System Engineering, Mokpo National Maritime University, E-mail: donghyun1439@gmail.com, Tel: +82-61-240-7200

² Professor, Division of Marine System Engineering, Mokpo National Maritime University, E-mail: tknam@mmu.ac.kr, Tel: +82-61-240-7255

³ Professor, Ocean Science & Technology School, Korea Maritime & Ocean University, E-mail: jk-choi@kmou.ac.kr, Tel: +82-51-410-4342

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

(TLS) technology can acquire high-density 3D point cloud data to provide shape information over a wide area [8]. However, there are practical constraints such as high equipment and operating costs, and the point cloud registration and post-processing process takes several days, making it difficult to immediately confirm results on-site [9].

Meanwhile, numerous studies on damage detection using point cloud data have been conducted. Olsen *et al.* [8] presented a damage assessment method that analyzes volumetric changes in structures using TLS point clouds. Mohammadi *et al.* [10] proposed a method of calculating surface variation through PCA-based eigenvalue analysis and using it as a damage detection indicator. Jovancevic *et al.* [3] segmented defect areas using region growing techniques with normal vectors and curvature information for aircraft outer panels, and calculated defect depth through PCA. Erkal and Hajjar [4] conducted research to detect and quantify various types of damage such as cracks, delamination, and corrosion by analyzing differences between predicted surface characteristics and actual measurements. Domestically, Song *et al.* [1] proposed a hull deformation detection method based on point cloud features, and Kim *et al.* [11] presented a hull deformation analysis method combining depth estimation and RANSAC plane estimation. However, most of these existing studies rely on KD-tree-based neighbor search targeting unstructured point clouds and presuppose offline post-processing environments, lacking consideration for immediate on-site applicability. Also, focusing on detection, systematic methodologies for quantitative damage measurement are relatively insufficient.

This paper proposes a methodology for detecting and quantitatively measuring surface damage of large planar structures on-site using low-cost 3D LiDAR sensors. The key of the proposed method is to utilize the fixed grid characteristics of structured LiDAR sensor data. With recent advances in autonomous driving and robotics, low-cost LiDAR sensors with measurement ranges of 100 m class have been commercialized, and these sensors output structured point clouds arranged at constant horizontal and vertical angular intervals. Such sensors generally provide fixed grid point clouds consisting of tens of vertical channels and hundreds of horizontal scan points. This structured grid has the advantage of enabling efficient neighbor search without building spatial search data structures like KD-trees, since each point's neighbor relationship is determined by index operations alone.

The proposed processing pipeline consists of four stages. First, in the preprocessing stage, hole filling, Gaussian smoothing, and

exponential moving average (EMA) filtering are applied utilizing the characteristics of the structured grid to suppress sensor noise and stabilize data. Second, in the curvature extraction stage, PCA analysis is performed on the covariance matrix around each point to calculate surface variation-based curvature information and normal vectors. At this time, radius-based neighbor search is implemented as grid index operations to ensure computational efficiency. Third, in the plane estimation stage, the main plane is estimated through consistency with the reference normal direction and distance-based dominant layer selection within the region of interest (ROI) specified by the user. This stage enables stable reference plane setting even in environments where damage areas are mixed. Fourth, in the damage measurement stage, the signed distance from the estimated plane to each point is calculated to quantify the depth of damage and visualize the spatial distribution.

The main contribution of this study is to realize efficient neighbor search and noise suppression without KD-trees by utilizing the fixed array characteristics of structured grid LiDAR, and to enable quantitative damage measurement through ROI-based plane estimation. Through this, damage measurement results at interactive-rate speed can be provided on-site with only low-cost LiDAR without expensive TLS equipment or long post-processing, which is expected to improve the accessibility and efficiency of large structure inspection.

The organization of this paper is as follows. Section 2 describes the detailed methodology of the proposed four-stage processing pipeline. Section 3 presents the experimental environment and measurement results, and Section 4 discusses conclusions and future research directions.

2. Proposed Damage Measurement Algorithm

2.1 System Overview

The proposed surface damage measurement system consists of a preprocessing stage, curvature extraction stage, plane estimation stage, and damage measurement stage. In the preprocessing stage, input point cloud data is stabilized utilizing the characteristics of the structured grid. First, gaps due to invalid measurements are filled with hole filling, and spatiotemporal smoothing is applied to suppress sensor noise. Spatiotemporal smoothing consists of a combination of a small Gaussian filter (3x3 kernel) for spatial noise suppression and an exponential moving average (EMA) filter for temporal stability.

In the curvature extraction stage, Gaussian smoothing is first

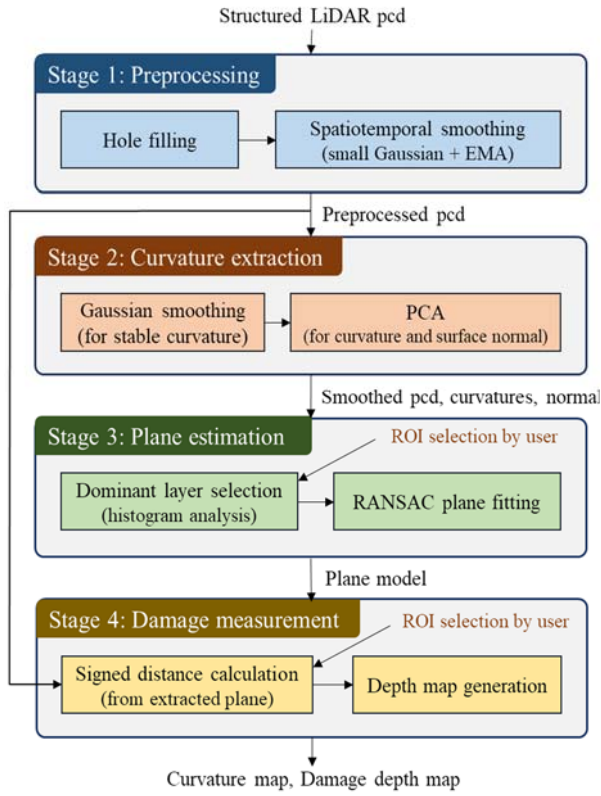


Figure 1: Overall procedure of the proposed method

applied for stable curvature and normal vector calculation, then PCA analysis is performed on the covariance matrix around each point to calculate surface variation-based curvature information and normal vectors. At this time, computational efficiency is secured by implementing radius-based neighbor search as grid index operations utilizing the fixed array characteristics of the structured grid.

In the plane estimation stage, the main plane is estimated within the ROI specified by the user. First, plane candidate point clouds are filtered by evaluating the consistency between the reference normal direction and each point's normal vector, and stable reference plane candidates are set even in environments where damage areas are mixed through dominant layer selection via distance histogram analysis. Then, the RANSAC algorithm is applied to the selected layer to estimate the final plane model.

In the damage measurement stage, the depth of damage is quantified by calculating the signed distance from the estimated main plane to each point. At this time, the point cloud from the preprocessing stage is used for distance calculation instead of the smoothed point cloud from the curvature extraction stage, to prevent the depth of damaged areas from being measured smaller

than actual due to excessive smoothing. Positive distance indicates protrusion, negative distance indicates dent, allowing simultaneous identification of damage direction and magnitude. Finally, distance information is generated as a depth map within the ROI specified by the user so that the shape and extent of damaged areas can be intuitively confirmed.

2.2 Preprocessing

The purpose of the preprocessing stage is to suppress measurement noise and restore sporadic invalid measurement points while preserving structural features such as edges and damage boundaries. Since the LiDAR sensor used in this study performs measurements with a fixed angular interval scan pattern, each point is directly represented by row index r and column index c on a 2D grid of size 56×576 . This structured representation enables neighbor search and filtering through simple index operations.

The first step of preprocessing is hole filling. LiDAR measurements intermittently produce invalid measurement points due to surface reflectance variations or occlusion, and if not handled, these can propagate as artifacts in subsequent smoothing processes. To prevent this, missing values are interpolated using the median of valid neighbors within a 3×3 window. However, to prevent erroneous interpolation in depth discontinuity regions, interpolation is performed only when the condition in **Equation (1)** is satisfied.

$$\sigma_d = \sqrt{\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (d_j - \bar{d})^2} < \tau_d \quad (1)$$

In **Equation (1)**, \mathcal{N}_i is the set of valid neighbors of the invalid point \mathbf{p}_i , $d_j = \|\mathbf{p}_j\|$ is the depth (norm) of each neighbor point, \bar{d} is the mean of neighbor depths, σ_d is the standard deviation of neighbor depths, and τ_d is the threshold. This ensures that holes near occlusion boundaries or surface discontinuities are not interpolated, thereby preserving edge features.

After hole filling, spatial Gaussian smoothing is applied for high-frequency noise reduction. To handle invalid (NaN) entries on the grid, a NaN-aware approach is used that computes weighted averages only over valid points, defined as **Equation (2)** for each coordinate channel $X \in \{x, y, z\}$.

$$X = \frac{G * (X \odot M)}{G * M} \quad (2)$$

In **Equation (2)**, G is the 2D Gaussian kernel, $*$ denotes convolution, \odot denotes element-wise multiplication, and M is the valid entry mask. This approach normalizes by the sum of weights from valid neighbors, adaptively adjusting filter support in boundary regions. The filter is implemented as a separable operation, allowing different smoothing strengths to be applied along each axis.

Finally, for improved measurement stability during real-time streaming data processing, the exponential moving average (EMA) filter in **Equation (3)** is applied.

$$\mathbf{p}_t^{ema} = (1 - \alpha)\mathbf{p}_{t-1}^{ema} + \alpha\mathbf{p}_t \quad (3)$$

In **Equation (3)**, the smoothing coefficient $\alpha \in (0, 1]$ is set by the user, where smaller values provide stronger smoothing effects but increase response delay. Points transitioning from invalid to valid are directly initialized with the current measurement value.

The preprocessed point cloud from the above process is used as input for curvature extraction (Stage 2) and damage measurement (Stage 4). The additional smoothing performed for curvature calculation in Stage 2 is applied separately and is not used for distance calculation (see Section 2.5).

2.3 Curvature Extraction

The purpose of the curvature extraction stage is to compute the local surface curvature and normal vector at each point in the point cloud. Curvature information is utilized to visually identify candidate damage regions, and normal vectors are used as criteria for selecting the measurement target surface in the subsequent plane estimation stage.

For stability in curvature calculation, additional Gaussian smoothing is applied to the preprocessed point cloud. This smoothing uses a wider kernel than the preprocessing stage smoothing to stabilize local surface characteristics and is applied exclusively for curvature calculation.

At each point \mathbf{p} , neighbor points within radius r are searched, with the maximum number of neighbors limited to k_{max} for computational efficiency. For the k searched neighbor points $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k\}$, the covariance matrix is constructed as in **Equation (4)**.

$$C = \frac{1}{k} \sum_{i=1}^k (\mathbf{p}_i - \bar{\mathbf{p}})(\mathbf{p}_i - \bar{\mathbf{p}})^T \quad (4)$$

In **Equation (4)**, $\bar{\mathbf{p}}$ is the centroid of the neighbor points. The covariance matrix C is a 3×3 symmetric matrix, and three eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$ are obtained through eigenvalue decomposition. Each eigenvalue represents the variance of point distribution along the corresponding eigenvector direction, and the eigenvector corresponding to the smallest eigenvalue λ_0 becomes the normal direction of the local surface.

Surface variation proposed by Pauly *et al.* [12] is used as the curvature metric, defined as **Equation (5)**.

$$\sigma = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (5)$$

Surface variation has values between 0 and 1, approaching 0 in regions close to planar and increasing in regions with high curvature. While this metric does not provide the exact value of principal curvature, it is effective for comparing relative curvature magnitudes and has the advantage of simple computation.

Normal vectors are obtained directly from eigenvectors, but eigenvectors have undefined sign direction, making \mathbf{n} and $-\mathbf{n}$ equivalent. For consistent normal direction, the sign is adjusted to point toward the sensor origin. That is, if the dot product of the normal vector \mathbf{n} and the direction vector toward the sensor origin is negative, the normal is inverted.

2.4 Plane Estimation

The purpose of the plane estimation stage is to determine the reference plane that serves as the basis for damage measurement. Since the surface of large structures can be locally approximated as a plane, a plane is estimated from the normal surface surrounding the damaged area and used as the reference for measuring damage depth.

General RANSAC-based plane estimation has difficulty stably extracting the desired surface when noise is severe or multiple planes coexist in the scene. To address this, in this study, pre-filtering using reference normal vectors and histogram-based layer selection are applied before RANSAC.

2.4.1 Candidate Point Selection

First, the user refers to the curvature visualization results and specifies a flat region without damage on the measurement target surface as the ROI. The average of normal vectors of points within the specified ROI is defined as the reference normal vector \mathbf{n}_{ref} , calculated as **Equation (6)**.

$$\mathbf{n}_{ref} = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{p}_i \in \mathcal{R}} \mathbf{n}_i \quad (6)$$

In **Equation (6)**, \mathcal{R} is the set of valid points within the ROI, and \mathbf{n}_i is the normal vector of each point. The calculated \mathbf{n}_{ref} is normalized to a unit vector.

Next, from the entire point cloud, only points with normal directions similar to \mathbf{n}_{ref} are selected as candidates. Points whose normal vector \mathbf{n} and \mathbf{n}_{ref} dot product satisfies the condition in **Equation (7)** form the candidate set \mathcal{P}_c .

$$\mathbf{n} \cdot \mathbf{n}_{ref} \geq \tau_{dot} \quad (7)$$

In **Equation (7)**, τ_{dot} is the dot product threshold, where larger values select only points with directions matching \mathbf{n}_{ref} . This process removes surfaces with different orientations from the measurement target (e.g., adjacent structures, floor surfaces).

2.4.2 Histogram-Based Layer Selection and Plane Fitting

The candidate point set \mathcal{P}_c may still contain multiple parallel surfaces. To separate these, points are projected along the \mathbf{n}_{ref} direction to construct a 1D histogram. The projection value s of each point \mathbf{p} is given by **Equation (8)**.

$$s = \mathbf{p} \cdot \mathbf{n}_{ref} \quad (8)$$

A histogram is constructed with bins of fixed interval Δs for the projection values s , and the bin containing the most points is selected. Points belonging to that bin form the dominant layer \mathcal{P}_L .

From the dominant layer \mathcal{P}_L , a plane is estimated using the RANSAC (Random Sample Consensus) algorithm. The plane is represented by the unit normal vector $(a, b, c)^T$ and the distance from the origin d as in **Equation (9)**.

$$ax + by + cz + d = 0 \quad (9)$$

In each RANSAC iteration, three points are randomly selected from \mathcal{P}_L to define a plane. For the remaining points, the signed distance $d(\mathbf{p})$ to the plane is calculated using **Equation (10)**.

$$d(\mathbf{p}) = \frac{ax + by + cz + d}{\sqrt{a^2 + b^2 + c^2}} \quad (10)$$

Points satisfying $|d(\mathbf{p})| \leq \varepsilon$ are determined as inliers, where ε is the distance threshold. This process is repeated N_{iter} times, and the plane with the most inliers is selected as the final model.

The normal direction of the estimated plane is adjusted so that its dot product with \mathbf{n}_{ref} is positive, maintaining consistent directionality.

2.5 Damage Measurement

In the damage measurement stage, the damage depth of each point is calculated using the preprocessed point cloud from Stage 1 and the reference plane estimated in Stage 3. Since the point cloud data with suppressed noise and temporal stabilization through preprocessing is used as input for distance calculation, more reliable damage measurement is possible.

The signed distance from each point $\mathbf{p} = (x, y, z)^T$ to the reference plane is calculated using **Equation (10)**. Since the plane's normal vector has been adjusted to point toward the sensor origin, $d(\mathbf{p}) > 0$ indicates that the point is closer to the sensor than the plane (protrusion), while $d(\mathbf{p}) < 0$ indicates it is farther from the sensor (dent).

The calculated signed distance values are reconstructed into a 2D array according to the LiDAR grid structure to form the distance map $D(r, c)$. Here, (r, c) are the row and column indices on the grid. Applying a colormap to the distance map for visualization allows intuitive identification of the location and extent of damaged areas.

For quantitative measurement of specific damage areas, the user specifies a region of interest (ROI) on the distance map. Within the specified ROI, the upper percentile d_{high} and lower percentile d_{low} of distance values are extracted, and damage type and amount are determined according to **Equation (11)**.

$$d_{damage} = \begin{cases} d_{high}, & \text{if } |d_{high}| \geq |d_{low}| \\ d_{low}, & \text{otherwise} \end{cases} \quad (11)$$

In this study, the 90th percentile was used for d_{high} and the 10th percentile for d_{low} . The method of directly using maximum and minimum values can be sensitive to sensor noise or outliers, increasing measurement error. The percentile-based method mitigates such peak errors, enabling more stable damage estimation.

If the sign of d_{damage} is positive, it is determined as protrusion, if negative, as dent, and the absolute value $|d_{damage}|$ represents the damage depth of that area.

3. Experiment Results

3.1 Experimental Setup

In this study, the ML-X 120D model LiDAR sensor from SOSLAB was used. This sensor outputs a structured point cloud

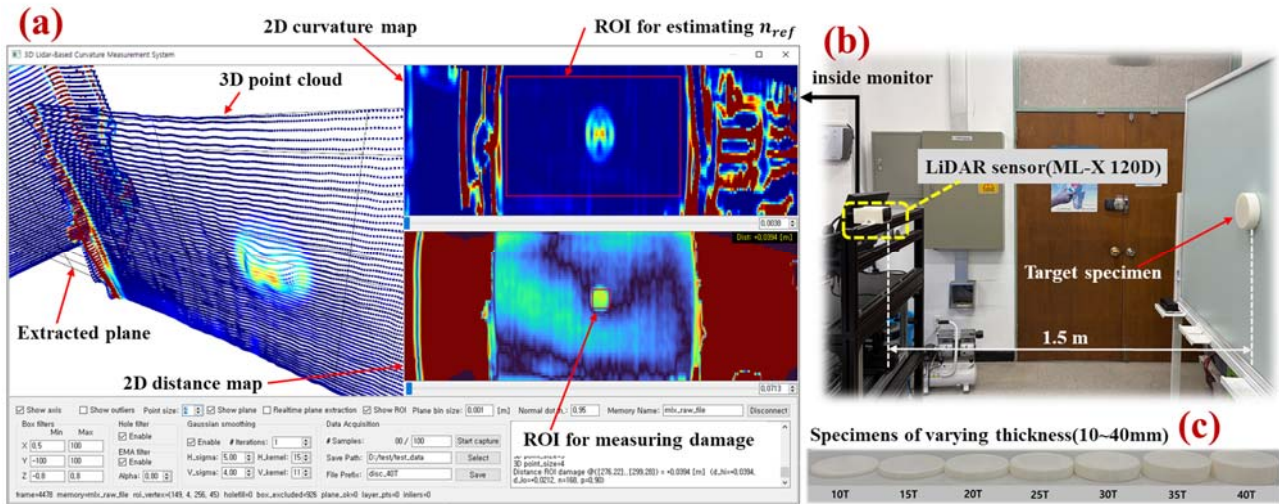


Figure 2: Experimental setup. (a) GUI program for damage measurement, (b) test environment, (c) specimens of varying thickness

Table 1: Algorithm parameters

Parameter	Value	Description
τ_d	0.5	Hole filling depth threshold [m]
α	0.7	EMA smoothing coefficient
τ_dot	0.9	Normal dot product threshold
Δs	0.002	Histogram bin size [m]

with a resolution of 56×576 and has a response frequency of 20 Hz. The horizontal field of view (FOV) is 120 degrees and the vertical FOV is 35 degrees. **Figure 2(b)** shows the experimental setup using the target LiDAR sensor, and **Figure 2(a)** shows the 3D point cloud output pattern acquired from the sensor through the 3D output screen within the GUI interface.

A white board was used as the reference plane, and test specimens were fabricated using a 3D printer. **Figure 2(c)** shows the test specimens used in the experiment, which are cylindrical shapes with a diameter of 140 mm, with R5 fillets applied to the top surfaces. This study fabricated specimens of various heights (10 mm to 40 mm, 5 mm intervals) to evaluate measurement accuracy. The LiDAR sensor was placed at a distance of approximately 1.5 m from the specimen and aligned so that the optical axis was perpendicular to the reference plane. For each specimen, 100 consecutive frames of data were acquired to evaluate measurement repeatability and statistical significance.

The main algorithm parameters used in the experiment are shown in **Table 1**. Additionally, the RANSAC iteration count was set to 1000 and the distance threshold to 0.005 m.

3.2 Evaluation Method

Figure 2(a) shows the interface of the developed measurement

system. The left side displays the 3D point cloud view, the upper right shows the curvature map, and the lower right shows the distance map in real-time. Users can specify an ROI on the distance map to measure the damage amount in that area.

The conventional maximum/minimum-based measurement method has a problem of being sensitive to peak errors caused by sensor noise or outliers. To solve this, this study adopted a percentile-based measurement method, applying the 90% percentile to d_{high} and the 10% percentile to d_{low} through experiments.

For measurement accuracy evaluation, Mean Error (average difference between measured and actual values), Standard Deviation (standard deviation of repeated measurements), and RMS Error (overall accuracy) were used.

3.3 Experimental Results

Figure 3 shows the distribution of measured values for each specimen at the $p=0.9$ setting as a boxplot. These are the results of 100 repeated measurements for each specimen, confirming generally stable measurement distributions under all conditions.

Table 2 summarizes the detailed measurement results for each specimen. The overall RMS error was measured as 0.64 mm. The

Table 2: Damage measurement accuracy results [unit: mm]

Specimen	G.T.	Mean	Std	Error
10T	10.0	11.27	0.21	+1.27
15T	15.0	15.14	0.69	+0.14
20T	20.0	19.79	0.31	-0.21
25T	25.0	24.63	0.30	-0.37
30T	30.0	30.46	0.12	+0.46
35T	35.0	35.03	0.23	+0.03
40T	40.0	40.19	0.26	+0.19

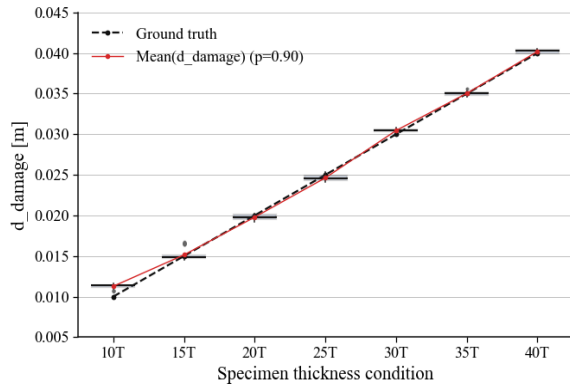


Figure 3: Boxplot of measured values by specimen condition

Table 3: Processing time by stage

Stage	Time (ms)
Preprocessing	8.4 ± 4.7
Curvature extraction	48.1 ± 4.5
Plane estimation	0.2 ± 0.6
Distance calculation	0.2 ± 0.4
Total	68.7 ± 9.3

10T specimen showed a relatively large error with a mean error of 1.27 mm. This is considered to be because as the damage depth decreases, the relative influence of sensor noise increases, showing a tendency for overestimation in percentile-based measurement. For specimens 15T and above (N=600), an RMS error of 0.45 mm confirmed stable measurement performance at sub-millimeter level.

Table 3 summarizes the processing time for each stage. Measurements were performed on 100 frames in an environment with an Intel Core i7-13700KF and 16 GB RAM.

The total processing time averaged 68.7 ms, corresponding to a processing rate of approximately 14.5 Hz. This is a level close to the LiDAR sensor's response frequency of 20 Hz, confirming that near real-time measurement is possible. Since the curvature extraction stage accounts for approximately 70% of the total, complete real-time processing is expected to be achievable through additional optimization of this part.

3.4 Discussion

Through this experiment, the performance of the proposed LiDAR-based damage measurement system was verified. The introduction of the percentile-based measurement method improved robustness against peak noise, and measured values showed excellent linearity with respect to specimen height. The 100 repeated measurements for each specimen showed standard deviations within 0.7 mm, confirming high repeatability.

The current average processing time of 68.7 ms corresponds to a processing rate of approximately 14.5 Hz, achieving near real-time performance close to the LiDAR sensor's response frequency of 20 Hz. Complete real-time processing is expected to be achievable through additional optimization of the curvature extraction stage. This experiment was conducted in a controlled laboratory environment, and additional verification is needed for application in actual ship inspection sites.

4. Conclusion

This study proposed a real-time surface damage measurement system using structured LiDAR point cloud data. The proposed system consists of a four-stage pipeline of preprocessing, curvature extraction, plane estimation, and damage measurement, enabling non-contact quantitative measurement of localized damage such as dents, deformations, and protrusions on large structures like ship hulls.

In the preprocessing stage, hole filling, Gaussian smoothing, and EMA filter were applied to suppress sensor noise and ensure temporal stability. In the curvature extraction stage, surface variation was calculated through PCA-based covariance analysis to detect damage candidate regions. In the plane estimation stage, plane candidate points were selected through reference normal-based filtering and histogram analysis, and robust plane parameters were estimated using RANSAC. In the damage measurement stage, a percentile-based method was applied to perform damage estimation robust against peak noise.

Experimental results showed that the proposed system achieved a measurement accuracy of 0.64 mm RMS error and repeatability within 0.7 mm. Measured values showed excellent linearity with respect to specimen height, and the total processing time averaged 68.7 ms, achieving a processing rate of approximately 14.5 Hz.

Future research will focus on implementing real-time processing through additional optimization of the curvature extraction algorithm, and verification for application in actual ship inspection sites.

Acknowledgement

This research was supported by the Regional Innovation System & Education(RISE) program through the Jeollanamdo RISE center, funded by the Ministry of Education(MOE) and the Jeollanamdo, Republic of Korea.(2025-RISE-14-002).

Author Contributions

Conceptualization, Heon-Hui Kim; Methodology, H.-H. Kim, T.-K. Nam, and J.-K. Choi; Software, H.-H. Kim and D.-H. Bae; Validation, H.-H. Kim and T.-K. Nam; Formal Analysis, H.-H. Kim and J.-K. Choi; Investigation, H.-H. Kim and D.-H. Bae; Resources, H.-H. Kim and D.-H. Bae; Data Curation, H.-H. Kim and D.-H. Bae; Writing—Original Draft Preparation, H.-H. Kim; Writing—Review & Editing, H.-H. Kim, T.-K. Nam, and J.-K. Choi; Visualization, D.-H. Bae and H.-H. Kim; Supervision, H.-H. Kim; Project Administration, H.-H. Kim; Funding Acquisition, H.-H. Kim.

References

- [1] S.-H. Song, G.-H. Lee, K.-M. Han, and H.-S. Jang, "Application of point cloud based hull structure deformation detection algorithm," *Journal of the Society of Naval Architects of Korea*, vol. 59, no. 4, pp. 235-242, 2022 (in Korean).
- [2] Y. I. Park, J.-S. Cho, and J.-H. Kim, "A study on the effect of dynamic material strain on the dynamic crushing behavior of steel structures subjected to impact loads," *Journal of Advanced Marine Engineering and Technology*, vol. 47, no. 1, pp. 16–22, 2023.
- [3] I. Jovančević, H.-H. Pham, J.-J. Orteu, R. Gilblas, J. Harvent, X. Maurice, and L. Brèthes, "3D point cloud analysis for detection and characterization of defects on airplane exterior surface," *Journal of Nondestructive Evaluation*, vol. 36, no. 4, Article 74, 2017.
- [4] B. G. Erkal and J. F. Hajjar, "Laser-based surface damage detection and quantification using predicted surface properties," *Automation in Construction*, vol. 83, pp. 285-302, 2017.
- [5] Y. Kim, S.-D. Yun, I. Park, B. Kim, and J. Yang, "Analyzing cut-edge corrosion in hot dip galvanized and Zn-Al-Mg alloy using image processing," *Journal of Advanced Marine Engineering and Technology*, vol. 49, no. 6, pp. 457–462, 2025.
- [6] C. Fernández-Isla, P. J. Navarro, and P. M. Alcover, "Automated visual inspection of ship hull surfaces using the wavelet transform," *Mathematical Problems in Engineering*, vol. 2013, no. 1, p. 101837, 2013.
- [7] B. Lin and X. Dong, "Ship hull inspection: a survey," *Ocean Engineering*, vol. 289, p. 116281, 2023.
- [8] M. J. Olsen, F. Kuester, B. J. Chang, and T. C. Hutchinson, "Terrestrial laser scanning-based structural damage assessment," *Journal of Computing in Civil Engineering*, vol. 24, no. 3, pp. 264-272, 2010.
- [9] C. Wu, Y. Yuan, Y. Tang, and B. Tian, "Application of terrestrial laser scanning (TLS) in the architecture, engineering and construction (AEC) industry," *Sensors*, vol. 22, no. 1, p. 265, 2022.
- [10] M. E. Mohammadi, R. L. Wood, and C. E. Wittich, "Non-temporal point cloud analysis for surface damage in civil structures," *ISPRS International Journal of Geo-Information*, vol. 8, no. 12, p. 527, 2019.
- [11] M. Kim, S. Song, and H. Jang, "Image-based analysis and visualization of hull deformation regions: an integrated approach using depth estimation and curvature analysis," *Journal of the Society of Naval Architects of Korea*, vol. 62, no. 2, pp. 77-87, 2025 (in Korean).
- [12] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," *Proceedings of IEEE Visualization*, pp. 163-170, 2002.