sensors

MDPI

Article

Image-Driven Hybrid Structural Analysis Based on Continuum Point Cloud Method with Boundary Capturing Technique

Kyung-Wan Seo [1], Junwon Park [1], Sang I. Park [2], Jeong-Hoon Song [3] and Young-Cheol Yoon [1,*]

[1] Department of Civil Engineering, Myongji College, Seoul 03656, Republic of Korea; k.w.seo@yonsei.ac.kr (K.-W.S.); william@willreal.tech (J.P.)
[2] Research Institute for Safety Performance, Korea Authority of Land & Infrastructure Safety, Jinju 52856, Republic of Korea; s.park@kalis.or.kr
[3] Department of Civil Environmental and Architectural Engineering, University of Colorado, Boulder, CO 80309, USA; jh.song@colorado.edu
* Correspondence: ycyoon@mjc.ac.kr

Abstract: Conventional approaches for the structural health monitoring of infrastructures often rely on physical sensors or targets attached to structural members, which require considerable preparation, maintenance, and operational effort, including continuous on-site adjustments. This paper presents an image-driven hybrid structural analysis technique that combines digital image processing (DIP) and regression analysis with a continuum point cloud method (CPCM) built on a particle-based strong formulation. Polynomial regressions capture the boundary shape change due to the structural loading and precisely identify the edge and corner coordinates of the deformed structure. The captured edge profiles are transformed into essential boundary conditions. This allows the construction of a strongly formulated boundary value problem (BVP), classified as the Dirichlet problem. Capturing boundary conditions from the digital image is novel, although a similar approach was applied to the point cloud data. It was shown that the CPCM is more efficient in this hybrid simulation framework than the weak-form-based numerical schemes. Unlike the finite element method (FEM), it can avoid aligning boundary nodes with regression points. A three-point bending test of a rubber beam was simulated to validate the developed technique. The simulation results were benchmarked against numerical results by ANSYS and various relevant numerical schemes. The technique can effectively solve the Dirichlet-type BVP, yielding accurate deformation, stress, and strain values across the entire problem domain when employing a linear strain model and increasing the number of CPCM nodes. In addition, comparative analysis with conventional displacement tracking techniques verifies the developed technique's robustness. The proposed technique effectively circumvents the inherent limitations of traditional monitoring methods resulting from the reliance on physical gauges or target markers so that a robust and non-contact solution for remote structural health monitoring in real-scale infrastructures can be provided, even in unfavorable experimental environments.

Keywords: hybrid structural analysis; continuum point cloud method; digital image processing; polynomial regression; essential boundary condition; boundary value problem

check for updates

Academic Editors: Stefano Berretti, Jean-Baptiste Thomas, Baptiste Magnier and Khizar Hayat

Received: 7 December 2024
Revised: 8 January 2025
Accepted: 9 January 2025
Published: 11 January 2025

Citation: Seo, K.-W.; Park, J.; Park, S.I.; Song, J.-H.; Yoon, Y.-C. Image-Driven Hybrid Structural Analysis Based on Continuum Point Cloud Method with Boundary Capturing Technique. *Sensors* **2025**, *25*, 410. https://doi.org/10.3390/s25020410

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Structural health monitoring (SHM) and deformation analysis play a vital role in maintaining the safety and functionality of infrastructure. Due to various loads, environmental influences, and aging, modern engineering structures demand reliable methods to

effectively evaluate deformation, stress, and strain for appropriate operation and maintenance [1–3]. Conventionally, these kinematic variables are measured using contact sensors or non-contact devices tracking predefined targets [4–6]. Contact sensors, such as strain gauges and fiber optic sensors, are known for their high precision and real-time feedback capabilities. However, they often meet challenges, such as the extensive resources required for data processing and issues with hostile sensor installation environments on actual infrastructures [7]. On the other hand, non-contact sensing approaches suggest interesting resolutive features, such as structural methodology couplings or nonphysical target implementation, offering viable alternatives for kinematic variable measurement [8–11].

In many practical cases, special treatment is required to extrapolate measured or computed kinematic values across an entire structure, which demands substantial computational exertion and time [12–14]. To avoid tricky situations and enrich the performance of traditional sensor utilization, considerable research has favored methods that merge FEM with non-contact sensing tools [15–18]. This combination tries to emulate the kinematic variables from displacement data gathered from direct and indirect sources [19–25]. Typically, during the combination, an FEM mesh is synchronized with the data measurement configuration, and the synchronization mainly focuses on numerically computed displacements, so kinematic variable calculation using FEM requires frequent adjustments to the mesh, especially when monitoring intricate deformations such as large deformation and crack-induced displacement. In this scenario, non-mesh techniques like mesh-free methods become essential, especially when the mechanical calculations need to be conducted at an arbitrary position outside predefined nodes or mesh grids. Mesh-free techniques, such as the boundary element method (BEM) [26,27] or the element-free Galerkin method (EFGM) [28,29], have emerged as alternatives to FEM to avoid mesh-connectivity-induced difficulties. In addition, recent studies have explored graph-based models, such as graph neural networks (GNN) [30] and vision-based techniques using single-camera systems [31], to manipulate high-fidelity numerical simulation combined with a data-driven machine learning approach and introduce a deep neural network designed to predict 3D mesh deformation directly from a single image input, combined with an initial 3D cube mesh input, respectively. However, these methods might rely on data-driven models or require complex calibration. However, the BEM is unsuitable for dealing with significant deformation inside the boundary element model, and the fundamental solution does not seem quite feasible for handling experimental data. Although the EFGM is included in the category of representative mesh-free techniques, it still requires the integration cell to integrate the weak form, making itself partially dependent on mesh structures [28,29]. In addition, its derivative approximation computation demands quite a complicated process and considerable computational resources. For this reason, a strong-form-based mesh-free method has been strongly recommended as a remedy [32–34]. It has been proven to have proper flexibility to provide digital image processing (DIP) for structural analysis based on a truly mesh-free platform, as seen in [35].

Park et al. [35] proposed a synthetic structural analysis scheme that combines the DIP with a strong-form particle method called PDM. The scheme employed displacement image data from targets attached to the real structure surface to simulate a real-scale experiment without solving governing partial differential equations. Although it manipulated PDM's node-wise displacement data to enhance the accuracy of kinematic variable tracking, it inevitably seriously relies on the target configuration and raw data quality. On the other hand, Seo et al. [36] presented an edge shape tracking technique that utilizes light detection and ranging (LiDAR) without preset targets on the specimen during the loading process. The technique recognizes the edge shape change due to the loading and transforms the change into essential boundary conditions for the numerical analysis based on the PDM.

Unlike Park et al. [35], the boundary value problem (BVP) formulated as a partial differential equation set should be solved to obtain the mechanical variables. Seo et al. [36] noted that the captured boundary deformations can be transformed into the essential boundary conditions for the Dirichlet-type BVP.

This study proposes a technique that captures the essential boundary condition using a digital image photographed from the real flexural loading test for a rubber beam and polynomial regression analysis. The BVP is equipped with the essential boundary condition and is solved by the strong-form-based mesh-free method called CPCM. Kinematic variables such as displacement, strain, and stress are obtained from the BVP solutions. Unlike Park et al. [35] and Seo et al. [36], the developed technique directly solves the BVP without involving target displacement data and utilizes the digital image for polynomial regression, respectively. The feasibility and accuracy of resultant kinematic variables are verified by comparing them with reference data and the results obtained by other relevant methods inspired by the methodologies of Park et al. [35] and Seo et al. [36].

2. Strongly Formulated Boundary Value Problem

2.1. Derivative Approximation Derivation

Derivative approximations are essential in solving the strongly formulated BVP. As referred to in [37], the PDM uses only nodes or particles to solve the BVP. The CPCM shares the same mathematical background as the PDM. It deals with a continuum body based on the neighbor cloud particles [37]. The governing equations are solved at every particle according to the particle's region, for example, the equilibrium equation for the interior nodes and the essential boundary equation for the boundary nodes. More specifically, based on the moving least squares method, the Taylor series is numerically expanded at the particles using neighbor cloud particles encompassed in the influence domain. In this process, shape functions and their derivatives are derived without any differentiation calculation. The order of the polynomial basis vector and the number of neighbor cloud particles affect the accuracy of the derivative approximation. Unlike the grid in the finite difference method, the mesh in FEM, or the integration cell in other meshless methods, the CPCM does not suffer from any restrictions of the structured grid or mesh or cell; additionally, its particle arrangement or even the number of particles can change during the analysis. Thus, the CPCM can sophisticatedly avoid edge extraction difficulties due to the persistence of the numerical model, such as the finite element mesh or the finite difference method grid.

In strongly formulated solid mechanics problems, the governing equations are often discretized in terms of displacement, leading to Navier's equation, which necessitates the second-order derivatives of displacements or the divergence of stress [34]. On the other hand, this study employs an improved numerical scheme that facilitates only first-order derivatives, as in Yoon et al. [37]. The stress divergence is discretized by consecutively using the first-order derivatives, achieving improved precision and efficiency. The following part presents a concise overview of the derivative approximation construction as detailed in [37].

A Taylor polynomial at an arbitrary position $\boldsymbol{x}$ against the reference point $\boldsymbol{y}$ based on the mth polynomial vector can be written as follows:

$$u_i(\mathbf{x};\mathbf{y}) = \mathbf{p}_m^T(\mathbf{x},\mathbf{y})\mathbf{a}_i(\mathbf{y}) = \left(\frac{(\mathbf{x}-\mathbf{y})^{\boldsymbol{\alpha}_1}}{\boldsymbol{\alpha}_1!}, \cdots, \frac{(\mathbf{x}-\mathbf{y})^{\boldsymbol{\alpha}_K}}{\boldsymbol{\alpha}_K!}\right) \cdot \begin{pmatrix} D_{\mathbf{x}}^{\boldsymbol{\alpha}_1} u_i(\mathbf{y}) \\ \vdots \\ D_{\mathbf{x}}^{\boldsymbol{\alpha}_K} u_i(\mathbf{y}) \end{pmatrix} \tag{1}$$

where $D_{\boldsymbol{x}}^{\boldsymbol{\alpha}} = \partial_{x_1}^{\boldsymbol{\alpha}_1} \cdots \partial_{x_n}^{\boldsymbol{\alpha}_n}$ represents the differential operator expressed in terms of multi-index natation; then, $\boldsymbol{\alpha}_1 = (0, \cdots, 0)$ and $\boldsymbol{\alpha}_K = (0, \cdots, m)$. $\boldsymbol{p}_m^{\mathrm{T}}(\boldsymbol{x}, \boldsymbol{y})$ represents the mth order polynomial basis vector, and $\mathbf{a}_i(\boldsymbol{y})$ is the derivative coefficient vector for spatial component $i = (0, \cdots, n)$. n represents the space dimension, and $K = \frac{(m+n)!}{m!n!}$. When there are N particles included in the influence domain of the reference point $\boldsymbol{y}$, Equation (1) is related to the nodal solution $u_{iI} = u_i(\boldsymbol{x}_I)$ by using the residual of the moving least squares method as follows:

$$J = \sum_{I=1}^{N} \omega_I(\mathbf{y}) \left(\mathbf{p}_m^{\mathrm{T}}(\mathbf{x}_I, \mathbf{y}) \mathbf{a}_i(\mathbf{y}) - u_{iI} \right)^2 \tag{2}$$

where $\omega_I(\boldsymbol{y}) = \omega\left(\frac{|\boldsymbol{x}_I - \boldsymbol{y}|}{r}\right)$ is the weight function, and r denotes the radius of influence domain that determines how many neighbor particles are included in the derivative approximation calculation. Minimizing Equation (2) is achieved by considering a stationary condition ($\frac{\partial J}{\partial \mathbf{a}} = 0$), yielding $\mathbf{a}_i(\boldsymbol{x})$, as follows:

$$\mathbf{a}_i(\boldsymbol{x}) = \left(\sum_{I=1}^{N} \omega_I(\boldsymbol{x}) \boldsymbol{p}_m(\boldsymbol{x}_I, \boldsymbol{x}) \boldsymbol{p}_m^{\mathrm{T}}(\boldsymbol{x}_I, \boldsymbol{x}) \right)^{-1} \cdot (\omega_1(\boldsymbol{x}) \boldsymbol{p}_m(\boldsymbol{x}_1, \boldsymbol{x}), \cdots, \omega_N(\boldsymbol{x}) \boldsymbol{p}_m(\boldsymbol{x}_N, \boldsymbol{x})) \cdot \boldsymbol{u}_i^h \tag{3}$$

where $\boldsymbol{u}_i^h = (u_{i1}, \cdots, u_{iN})^{\mathrm{T}}$ is a vector that collects nodal solutions, and the $\boldsymbol{y}$ of Equation (2) can be replaced by $\boldsymbol{x}$ via the moving process [15,16]. Recall that the $\mathbf{a}_i(\boldsymbol{x})$ contains all the derivative approximations from the zeroth order to mth order, corresponding to the Taylor polynomial in Equation (1). By rearranging Equation (3) in a matrix form, we have the following expression for the shape function, its derivatives, and the nodal solution:

$$\mathbf{a}_i(\boldsymbol{x}) = \begin{pmatrix} \phi_1^{[\boldsymbol{\alpha}_1]}(\boldsymbol{x}) & \cdots & \phi_N^{[\boldsymbol{\alpha}_1]}(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ \phi_1^{[\boldsymbol{\alpha}_K]}(\boldsymbol{x}) & \cdots & \phi_N^{[\boldsymbol{\alpha}_K]}(\boldsymbol{x}) \end{pmatrix} \cdot \begin{pmatrix} u_{i1} \\ \vdots \\ u_{iN} \end{pmatrix} \tag{4}$$

Recall from Equation (1) that $\mathbf{a}_i(\boldsymbol{x}) = \left(D_{\boldsymbol{x}}^{\boldsymbol{\alpha}_1} u(\boldsymbol{x}), \cdots, D_{\boldsymbol{x}}^{\boldsymbol{\alpha}_K} u(\boldsymbol{x}) \right)$. In the above, $\phi_I^{[\boldsymbol{\alpha}_1 = (0,0)]}(\boldsymbol{x})$ and $\phi_I^{[\boldsymbol{\alpha}_K = (0,\mathrm{m})]}(\boldsymbol{x})$ denotes the zeroth and $\boldsymbol{\alpha}_K$th order shape functions for particle I at $\boldsymbol{x}$ for the two-dimensional case, respectively. In the above, since the first row is the collection of the zeroth order shape functions, the i directional displacement value at $\boldsymbol{x}$ point is then calculated by $u_i^h(\boldsymbol{x}) = \sum_{I=1}^{N} \phi_I^{[(0,0)]}(\boldsymbol{x}) u_{iI}$. It can be stressed that calculation of $\mathbf{a}_i(\boldsymbol{x})$ requires no actual differentiation; all derivative approximations up to the mth order are simultaneously computed by matrix inversion in Equation (3). More detailed derivation procedures for these derivative approximations can be found in [34,37].

2.2. Governing Equation Discretization and Kinematic Variable Calculation

During the structural loading test, deformation data of the specimen edge are gathered from constant-time-interval photoshoots using a high-resolution digital camera. Subsequently, DIP and regression analysis are carried out to quantify the essential boundary value due to deformation. As a result, all boundary values can be prescribed, and the Dirichlet-type BVP can be defined entirely for the given problem domain. Commonly, the BVP comprises both essential and natural boundary conditions in solid mechanics problems, but the BVP is set up only using an essential boundary condition in this study. Figure 1 shows the computational domain discretized with interior and boundary nodes. The equilibrium equation is solved at each interior node, and the captured essential boundary conditions are assigned to the boundary nodes. After solving the BVP using the CPCM, kinematic variables such as displacement and strain are computed for all interior nodes

using Equation (4). In addition, the stress can be easily calculated in the same manner as in FEM using a constitutive equation.

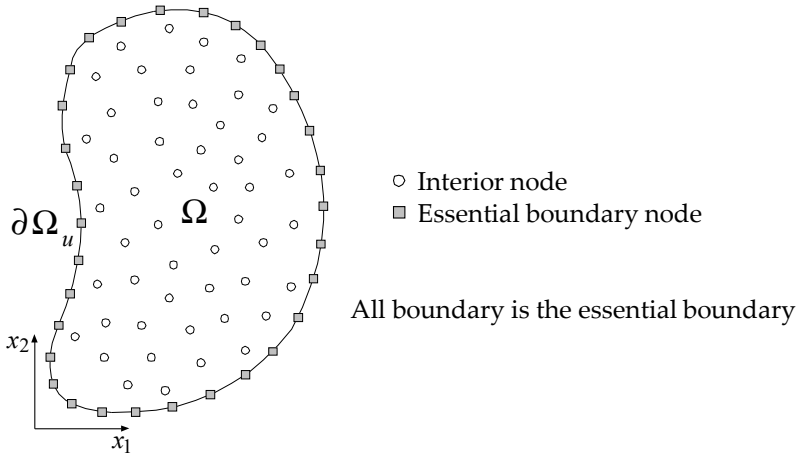


Figure 1. Dirichlet-type BVP configuration modeled with interior and boundary nodes.

A brief explanation of the CPCM formulation for a solid body is given here; more details can be found in Yoon et al. [38]. When the deformation occuring in the body is small, and the problem is static, the equilibrium equation to be solved for the interior nodes is written as

$$\sigma_{ij,j} + b_i = 0 \text{ in } \Omega \tag{5}$$

where σ_{ij} represents the Cauchy stress tensor, and b_i stands for the body force. Since the stress tensor includes first-order derivatives of displacement, the equilibrium equation finally involves the second-order derivatives. Since elastic material is considered in this study, the constitutive equation is given as

$$\sigma_{ij} = \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij} \tag{6}$$

where μ and λ are Lamé constants, ε_{ij} is the strain tensor, and δ_{ij} is a second-order identity tensor. In the small deformation problem, the strain is expressed as the symmetric part of the displacement gradient as follows:

$$\varepsilon_{ij} = \frac{1}{2}\left(u_{i,j} + u_{j,i}\right) \tag{7}$$

When the body force is absent, the equilibrium equation of Equation (5) can be expressed in a matrix form as below:

$$\boldsymbol{B}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B}(\boldsymbol{x}) \boldsymbol{u}(\boldsymbol{x}) = 0 \tag{8}$$

where $\boldsymbol{u}(\boldsymbol{x}) = (u_{11}(\boldsymbol{x}), \cdots, u_{2N}(\boldsymbol{x}))^{\mathrm{T}}$. The above equation needs to be assembled over all the interior nodes during the construction of the stiffness matrix of the total system. The strain and stress are calculated based on Equations (6) and (7) using the computed displacements and derivative approximations, respectively; i.e., $\varepsilon_{ij} = \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})$ and $\sigma_{ij} = \boldsymbol{D}\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})$, where the $\boldsymbol{B}(\boldsymbol{x})$ matrix is constructed using the first-order derivatives of the shape function, and $\boldsymbol{D}$ matrix describes the material properties given in Equation (6), respectively, as follows:

$$\boldsymbol{B}(\boldsymbol{x}) = \sum_{I=1}^{N} \begin{bmatrix} \varphi_I^{[(1,0)]}(\boldsymbol{x}) & 0 \\ 0 & \varphi_I^{[(0,1)]}(\boldsymbol{x}) \\ \varphi_I^{[(0,1)]}(\boldsymbol{x}) & \varphi_I^{[(1,0)]}(\boldsymbol{x}) \end{bmatrix} \tag{9}$$

$$\boldsymbol{D} = \begin{bmatrix} 2\mu + \lambda & \lambda & 0 \\ \lambda & 2\mu + \lambda & 0 \\ 0 & 0 & \mu \end{bmatrix} \tag{10}$$

The essential boundary condition for the Dirichlet-type BVP is given by

$$u_i = \overline{u_i} \text{on } \partial\Omega_u \tag{11}$$

where $\overline{u_i}$ denotes the prescribed boundary value. Note that the value of the structure's edge deformation obtained from the DIP and regression analysis is assigned using Equation (12) as the essential boundary condition. Namely, Equation (11) is implemented for all the boundary nodes by the following matrix form:

$$\boldsymbol{I}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}) = \overline{u_i} \tag{12}$$

where $\boldsymbol{I}(\boldsymbol{x})$ is an interpolation operator that includes the zeroth order shape function in its diagonal slots and is discretized with N neighbor particles as follows:

$$\boldsymbol{I}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x}) = \sum_{I=1}^{N} \begin{bmatrix} \varphi_I^{[(0,0)]}(\boldsymbol{x}) & 0 \\ 0 & \varphi_I^{[(0,0)]}(\boldsymbol{x}) \end{bmatrix} \begin{pmatrix} u_{1I} \\ u_{2I} \end{pmatrix} \tag{13}$$

The CPCM assembles all discrete governing equations given in Equations (8) and (12) for all the nodes. The assemblage for all nodes yields a final system of equations for Dirichlet-type BVP. When dealing with the essential boundary condition, unlike other weak-form-based mesh-free methods, any enforcement using the constraint formulation, such as the Lagrange multiplier method and penalty method, is not necessary [28]. In addition, unlike the FEM, it is not required to send the known boundary values to the right-hand side of the system together with the corresponding stiffness components. The total system of equations is written in an algebraic equation form as follows:

$$\begin{bmatrix} \boldsymbol{A}_{I=1}^{N}(\boldsymbol{B}_I^T \boldsymbol{D}_I \boldsymbol{B}_J) \\ \boldsymbol{A}_{K=1}^{M}(\boldsymbol{I}_K) \end{bmatrix} \begin{bmatrix} u_{11} \\ \vdots \\ u_{2N} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{A}_{K=1}^{M}(\overline{\boldsymbol{u}}_K) \end{bmatrix} \tag{14}$$

where $\boldsymbol{A}_{I=1}^{N}$ refers to an assemblage of the equilibrium equations for interior nodes 1 to N, and $\boldsymbol{A}_{K=1}^{M}$ represents an assemblage of the essential boundary equations for boundary nodes 1 to M. The stiffness matrix is constructed by assembling discrete equations according to the original node number. The assembly does not need to be conducted separately for the internal and boundary nodes. Therefore, the discrete governing equations can be easily assembled according to the original node number.

3. Experimental and Simulation Verifications

3.1. Experimental Configuration and Image-Driven Edge Shape Capturing

Digital images used for experimental verification were obtained from the experiment conducted by Park et al. [35]. In the experiment, a simply supported rubber beam under a three-point bending condition was loaded and digitally photographed (see Figures 2 and 3). In the simulation, Young's modulus of E = 2 MPa and Poisson's ratio of v = 0.45 were assumed. For the digital image acquisition, a Canon 5D Mark IV (Tokyo, Japan) full-frame camera with a resolution of 30.1 million pixels (6720 pixels wide × 4480 pixels long) equipped with a Canon EF 24-70 F2.8L II lens (Tokyo, Japan) was used. Snapshots of the experimental setup are given in Figure 3a,b. In this setting, the unit pixel size was approximately 0.063 mm. The external load increased steadily at 0.11 mm/s, reaching

2046.5 *N*, while the maximum deflection was limited to 50 mm. Images were taken and stored every 2 s to capture the specimen's deformation. Although the images, including the ordered square targets, were acquired in this study, the target images are not used for edge tracking; the specimen's edge images are selectively used. Note that the BVP analysis of this study employs the digital image of the deformed beam only to capture the edge shape change. The image taken when the central deflection was 2.64 mm is used for simulation since the rubber beam is regarded to be in an elastic stage up to that point. In addition, a similar experiment can be found in Seo et al. [36].

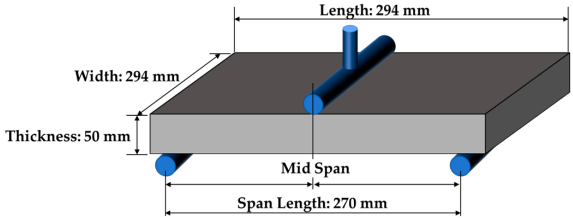
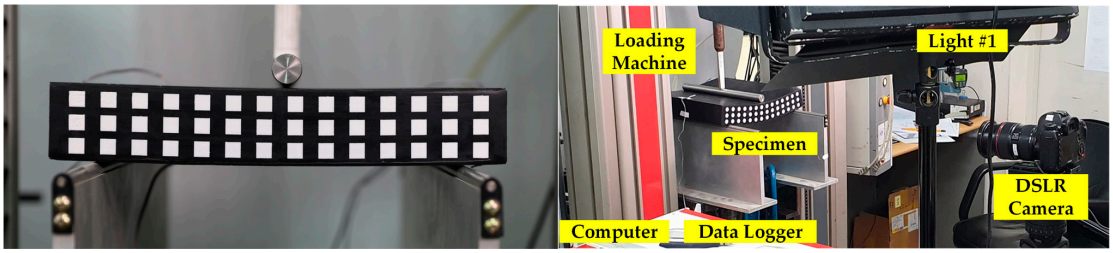


Figure 2. Configuration of three-point bending test for rubber beam.



(**a**) (**b**)

Figure 3. Experimental setup for digital image acquisition of the 3-point bending test: (**a**) specimen setup; (**b**) device setup.

Figure 4 illustrates the boundary capturing procedure for estimating the edge points of the deformed specimen from the digital image and applying essential boundary values to the CPCM analysis. The digital image is handled in jpg file format, and the entire process was carried out in a Python 3.11 environment using the OpenCV 4.5.5 [39], Open3D 0.16.0 [40], and SciPy 1.7.3 [41] libraries. As shown in Figure 5a,b, Step 1 involves ROI (region of interest) determination processing from the initial digital image for the specimen. Step 2 explains how to determine the edge points of the deformed specimen image. First, the Canny edge detection algorithm [42] was applied. The results are shown in Figure 5c,d. The edge points were successfully captured to be implemented in the CPCM model as essential boundary conditions. However, it is important to note that Canny's edge detection algorithm may be sensitive to noise and lighting variations, potentially affecting its robustness in real-world applications. This process estimates edges by detecting pixel value change, i.e., after removing other outliers, edge points are specified regardless of the attached targets on the specimen surface. As a result, edge points for the boundary of the deformed specimen were extracted, as shown in Figure 5e,f. Step 3 involves applying essential boundary values to the CPCM boundary nodes (See Figure 5g,h). Figure 5i,j represents the CPCM analysis models before and after loading simulation. When comparing these with the results obtained using the LiDAR point cloud data [1], it is confirmed that the Canny edge estimation algorithm extracts more qualified edge points (pixels) from the digital image. Regression analysis can reduce the overfitting risk by interpolating the deformed edge shape. In addition, note that fewer errors occur as the degree of the polynomial regression equation increases.

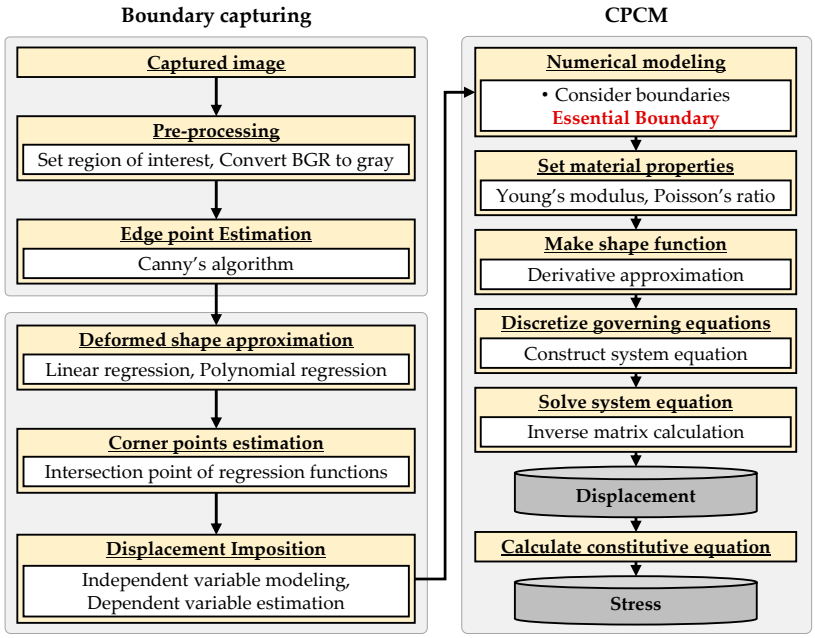


Figure 4. The procedure of estimation and assignment of the essential boundary value from the digital image to the CPCM simulation.

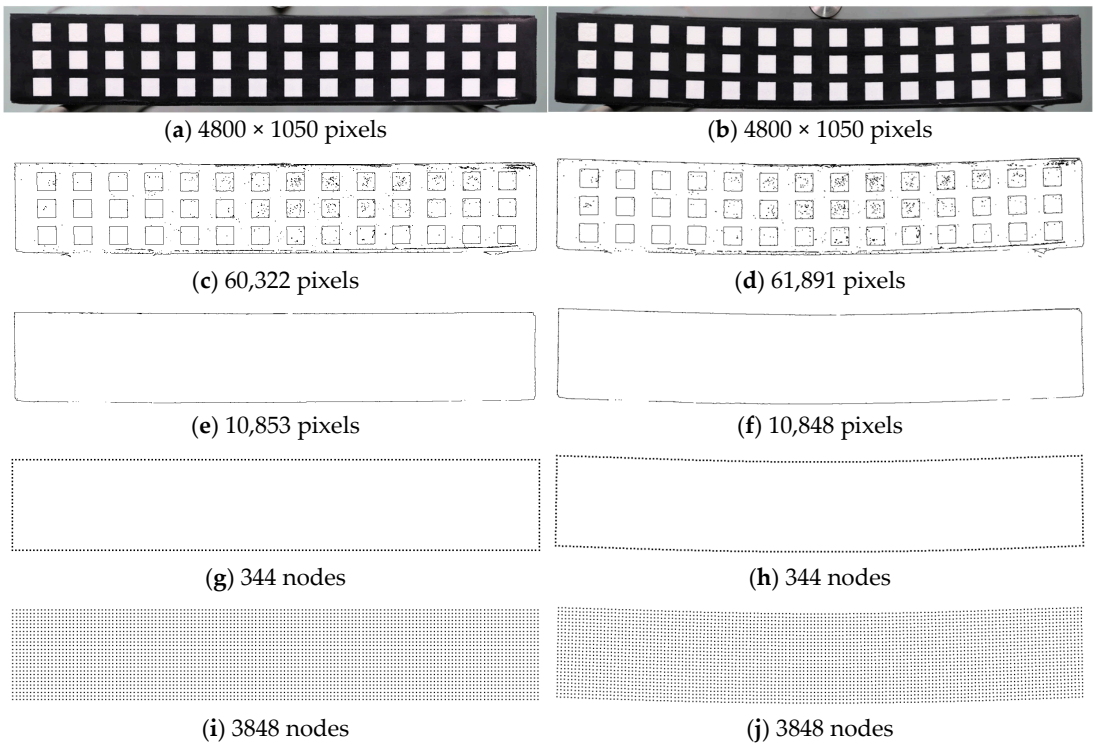


Figure 5. DIP-based boundary capturing results by the steps outlined in Figure 4: (**a**) the initial state image (step 1); (**b**) the deformed shape image (step 1); (**c**) 2D edge pixels (step 2 for the initial state); (**d**) 2D edge pixels (step 2 for the deformed specimen); (**e**) 2D edge pixels (step 3 for the initial state); (**f**) 2D edge pixels (step 3 for the deformed specimen); (**g**) captured essential boundary (step 3 for the initial state boundary nodes); (**h**) captured essential boundary (step 3 for boundary nodes after loading simulation); (**i**) total CPCM model for initial state; (**j**) total CPCM model after loading simulation.

3.2. Essential Boundary Capturing Using Polynomial Regression

The extracted edge points need to be transformed into the essential boundary values for boundary nodes of the CPCM model. A one-dimensional polynomial regression equation is applied to track the shape changes of the deformed boundary as follows:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n = \boldsymbol{\beta}^{\mathrm{T}} \cdot \boldsymbol{p}(x) \tag{15}$$

where x is the independent variable, meaning the relative coordinate along an edge axis; n is the highest degree of the polynomial regression equation; $\boldsymbol{\beta}^{\mathrm{T}} = (\beta_0, \beta_1, \ldots, \beta_n)$ is the coefficient vector of the regression equation; and $\boldsymbol{p}^{\mathrm{T}}(x) = (1, x, \ldots, x^n)$ is the one-dimensional polynomial vector to describe the shape of the deformed boundary.

Seo et al. [36] proposed a simple interpolation method for determining the displacement of boundary nodes of the CPCM model in two-dimensional BVP. The method employs a linear strain model to estimate the coordinates of edge-axis nodes of the deformed specimen. When the edge-axis is coaxial with the x-axis, the independent variable coordinates $\boldsymbol{x}_{ind}^{i}$ of the regression equation can be expressed as follows:

$$\boldsymbol{x}_{ind}^{i} = \boldsymbol{x} + \boldsymbol{u}\left(x^i\right) \tag{16}$$

where the superscript denotes the number of the edge nodes, starting from zero and ending at n, and $\boldsymbol{x}_{ind}^{i}$ represents the coordinates after deformation at the ith node along the edge-axis. Figure 6 illustrates the method of calculating the edge-axis displacement when considering the linear strain at an arbitrary position for the end values, $u\left(x^0\right)$ and $u(x^n)$, with zero strain at x^n. For more straightforward computation, the strain is assumed to be constant within a section divided into equal size, and the value at the center of the section is taken.

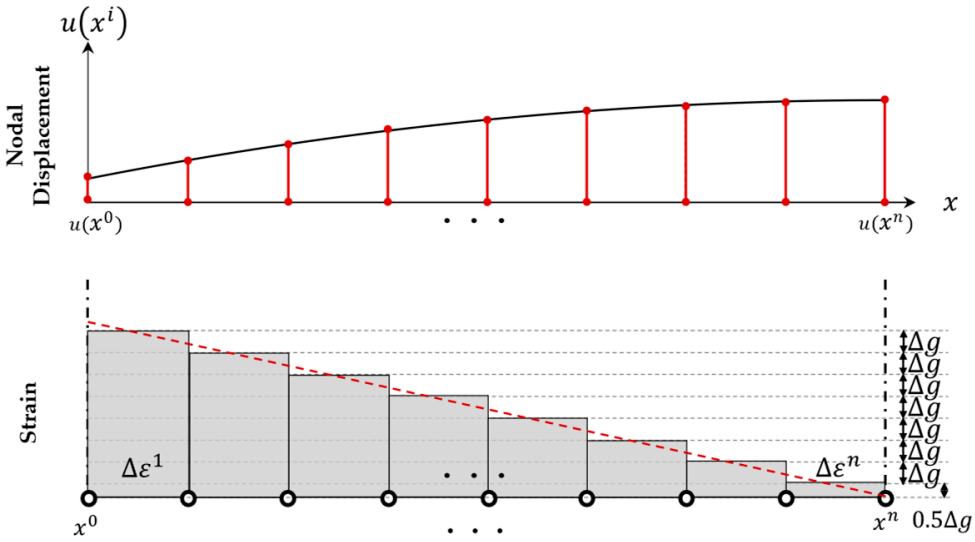


Figure 6. Edge displacement interpolation along the boundary nodes based on linear strain model with zero strain at x^n and end deformations $u\left(x^0\right)$ and $u(x^n)$.

The edge-axis displacement can then be expressed as a quadratic function of x^i, as follows:

$$u\left(x^i\right) = a\left(x^i\right)^2 + bx^i + c, \left(x^0 \le x^i \le x^n\right) \tag{17}$$

where a, b, and c are coefficients of the quadratic polynomial $u\left(x^i\right)$, and in fact, c denotes the displacement at the starting point $u\left(x^0\right)$. For example, if the edge-axis is coaxial with the x-axis and the spacing of boundary nodes of the CPCM model is uniform, the nodal displacement $u\left(x^k\right)$ of the kth node can be calculated as the cumulative sum of strains by Equation (18). More details can be found in Seo et al. [36]

$$u\left(x^k\right) = u\left(x^0\right) + \int_{x^0}^{x^k} \frac{du\left(x^i\right)}{dx} dx \approx u\left(x^0\right) + \Delta\varepsilon^1 + \ldots + \Delta\varepsilon^k = u\left(x^0\right) + \sum_{j=1}^{k} \Delta\varepsilon^j \quad (18)$$

3.3. BVP Analysis Results Using the CPCM Equipped with Boundary Capturing Technique

This section provides image-driven hybrid structural analysis results using the CPCM with the boundary capturing technique. The results are also compared with those obtained by the conventional methods, including the target-based DIP method, and the accuracy of the displacement boundary capture based on the polynomial regression equation is demonstrated together. The first analysis was performed using a complete numerical simulation using commercial structural analysis software, ANSYS 2020, under the same conditions as the three-point bending experiment. This full ANSYS simulation result is for comparison and a posteriori error analysis. The second analysis was conducted by applying the essential boundary condition captured by the regression equation using the edge points extracted from the ANSYS analysis result regarding the deformed boundary shape, where ANSYS only provides the shape of boundary deformation. The CPCM performed the third analysis, equipped with the boundary capturing technique consisting of DIP and polynomial regression analysis. The edges of the beam specimen were numbered in the same manner as in Seo et al. [36]. As seen in Figure 7, lines 1 and 4 are approximated by the second- to fifth-degree polynomial regression equations; the linear and uniform strain models are selected for the edge displacement capturing.

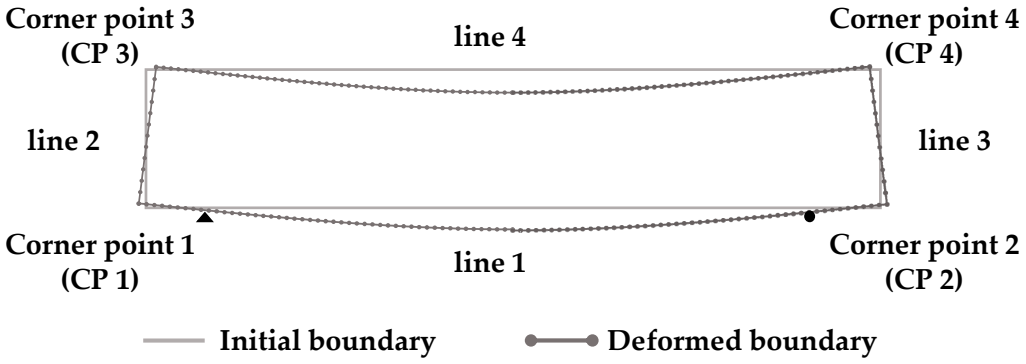


Figure 7. Essential boundary configuration for the rubber beam specimen (edge and corner point numbering).

It is expected that the deformation due to the bending motion dominantly occurs along the top and bottom surfaces, so the combination of the high-order regression equations and the linear strain model is used in this case. However, since it is relatively small along both side surfaces, the linear regression equations are chosen to approximate the displacement of lines 2 and 3 together with the uniform strain model. A quadratic polynomial vector and Gaussian weight function are used in the CPCM simulation, and 2 mm node spacing is applied to the CPCM model. The fifth-degree polynomials are applied for regression analysis for the top and bottom surfaces (line 1 and line 4), while the first-degree polynomials are used for both side surfaces (line 2 and line 3). Figures 8–10 present the simulation results for kinematic variables, such as displacement and stress, obtained by four methods to compare the computed values. Table 1 summarizes four different combinations, and the

combinations of the analysis tools and boundary capturing methods are listed. Method 1 indicates the case of full ANSYS simulation. The edge shape of the deformed specimen can be captured using the combination of DIP and polynomial regression using the deformed specimen image and the ANSYS analysis result (Method 2, 3, and 4). In Method 2, the ANSYS analysis result was only used as the data source for boundary capturing. Note that the main developments of this study are focused on Methods 3 and 4. Selection of the strain models differentiates Methods 3 and 4, i.e., Method 3 denotes the case with the linear strain model, but Method 4 involves the uniform strain model.

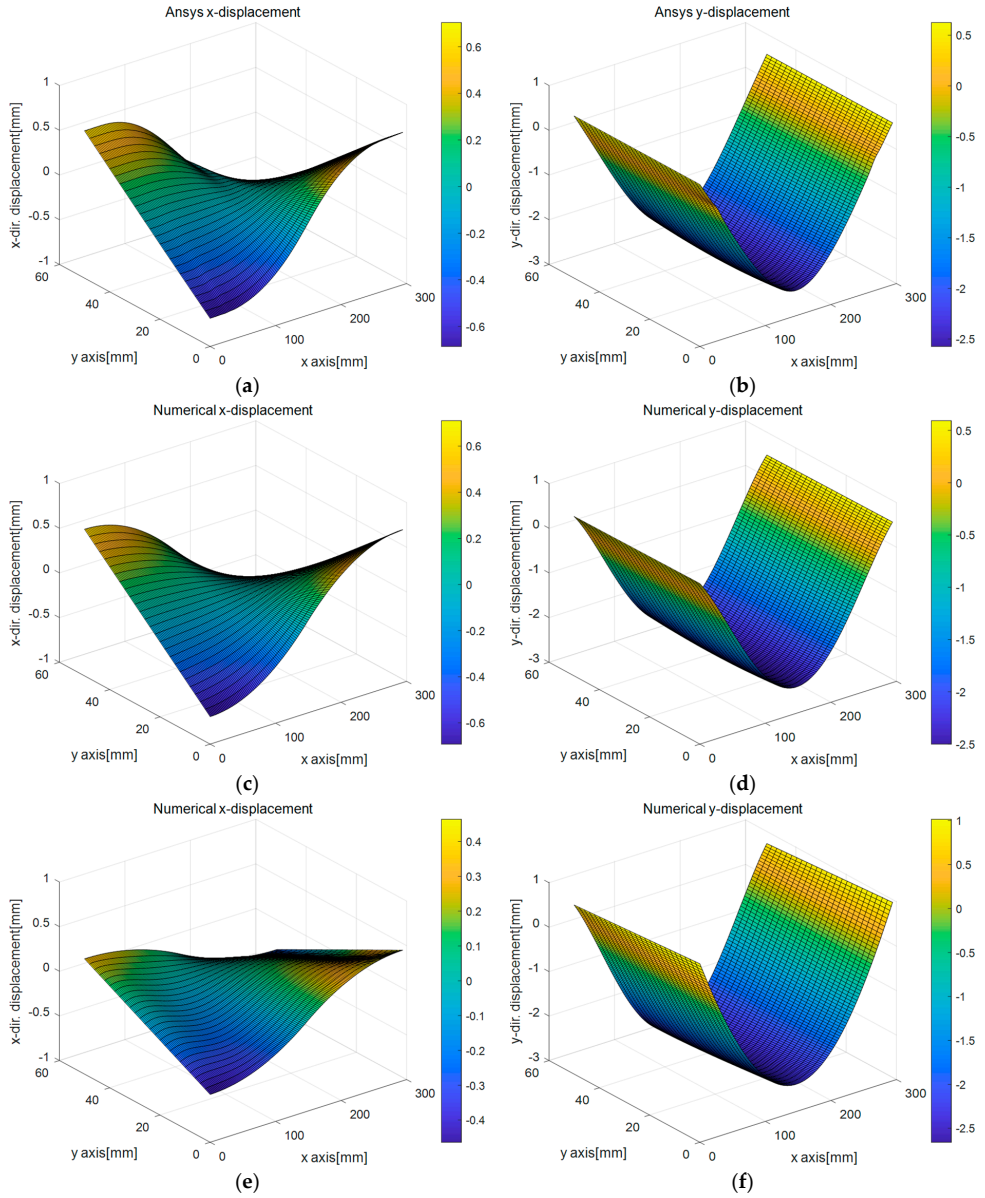


Figure 8. *Cont.*

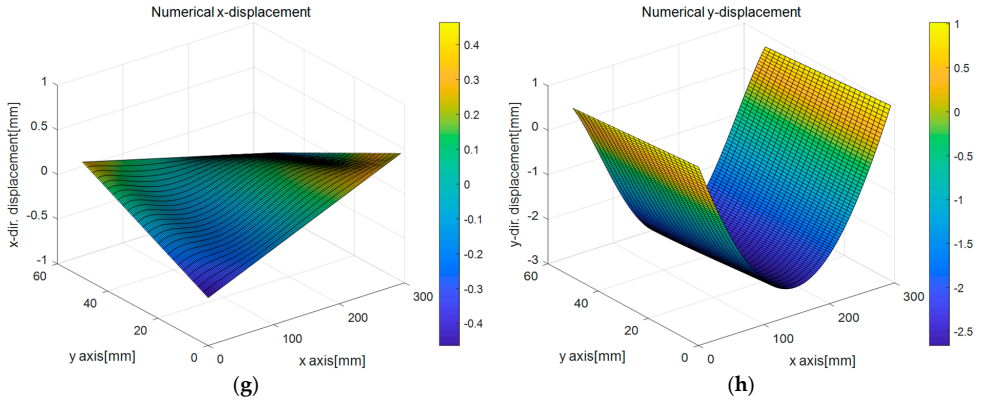


Figure 8. Surface plots for computed displacements obtained by different simulation methods and boundary capturing schemes for 3-point rubber beam bending test: (**a**) u_x (Method 1), (**b**) u_y (Method 1), (**c**) u_x (Method 2), (**d**) u_y (Method 2), (**e**) u_x (Method 3), (**f**) u_y (Method 3), (**g**) u_x (Method 4), (**h**) u_y (Method 4).

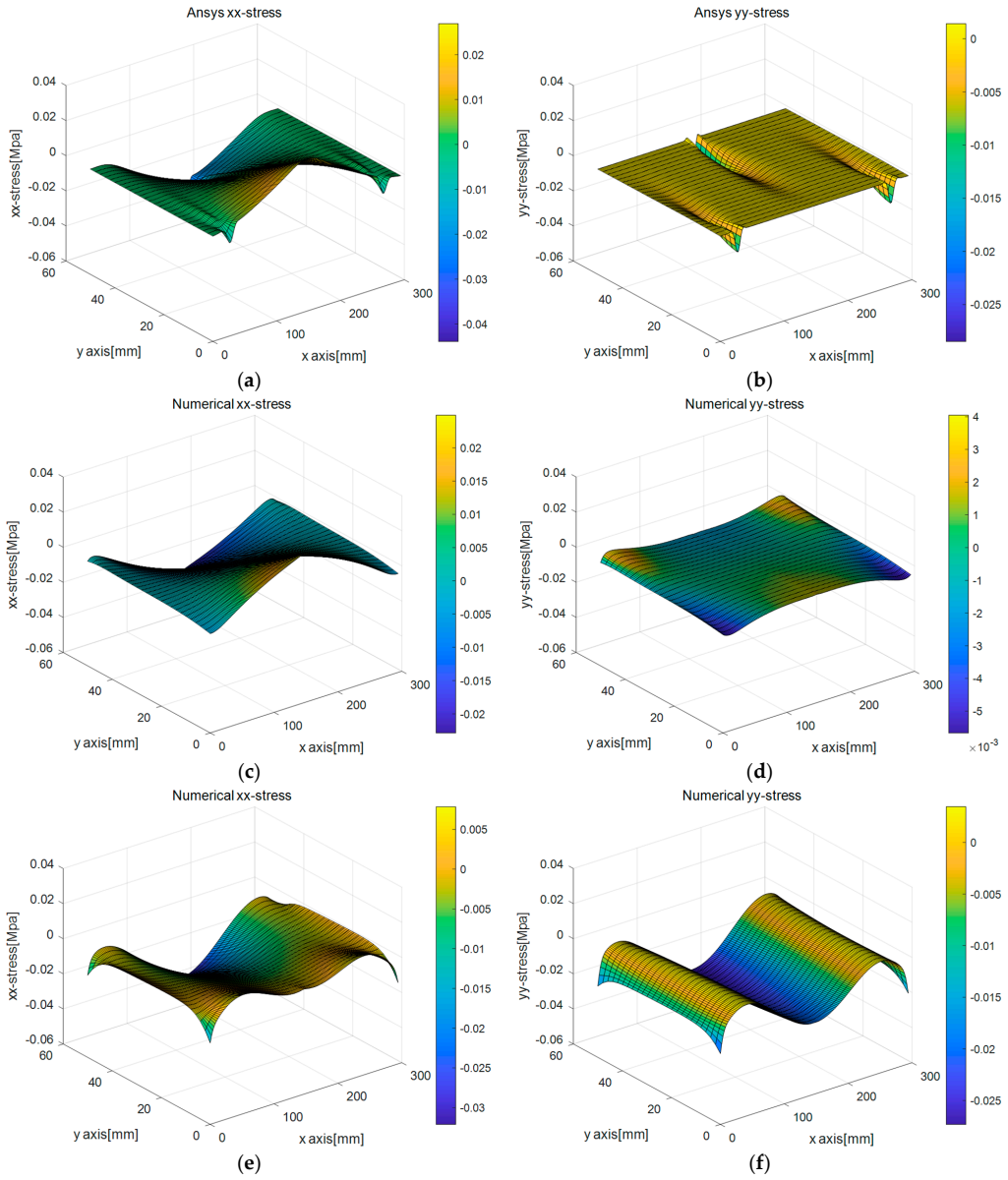


Figure 9. *Cont.*

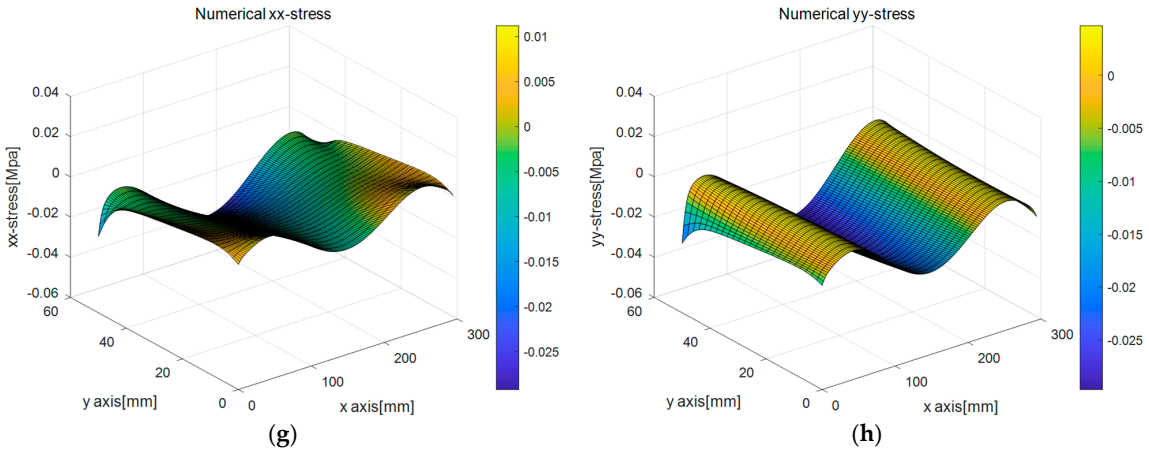


Figure 9. Surface plots for computed Cauchy stresses by different simulation methods and boundary capturing methods for 3-point rubber beam bending test: (**a**) σ_{xx} (Method 1), (**b**) σ_{yy} (Method 1), (**c**) σ_{xx} (Method 2), (**d**) σ_{yy} (Method 2), (**e**) σ_{xx} (Method 3), (**f**) σ_{yy} (Method 3), (**g**) σ_{xx} (Method 4), (**h**) σ_{yy} (Method 4).

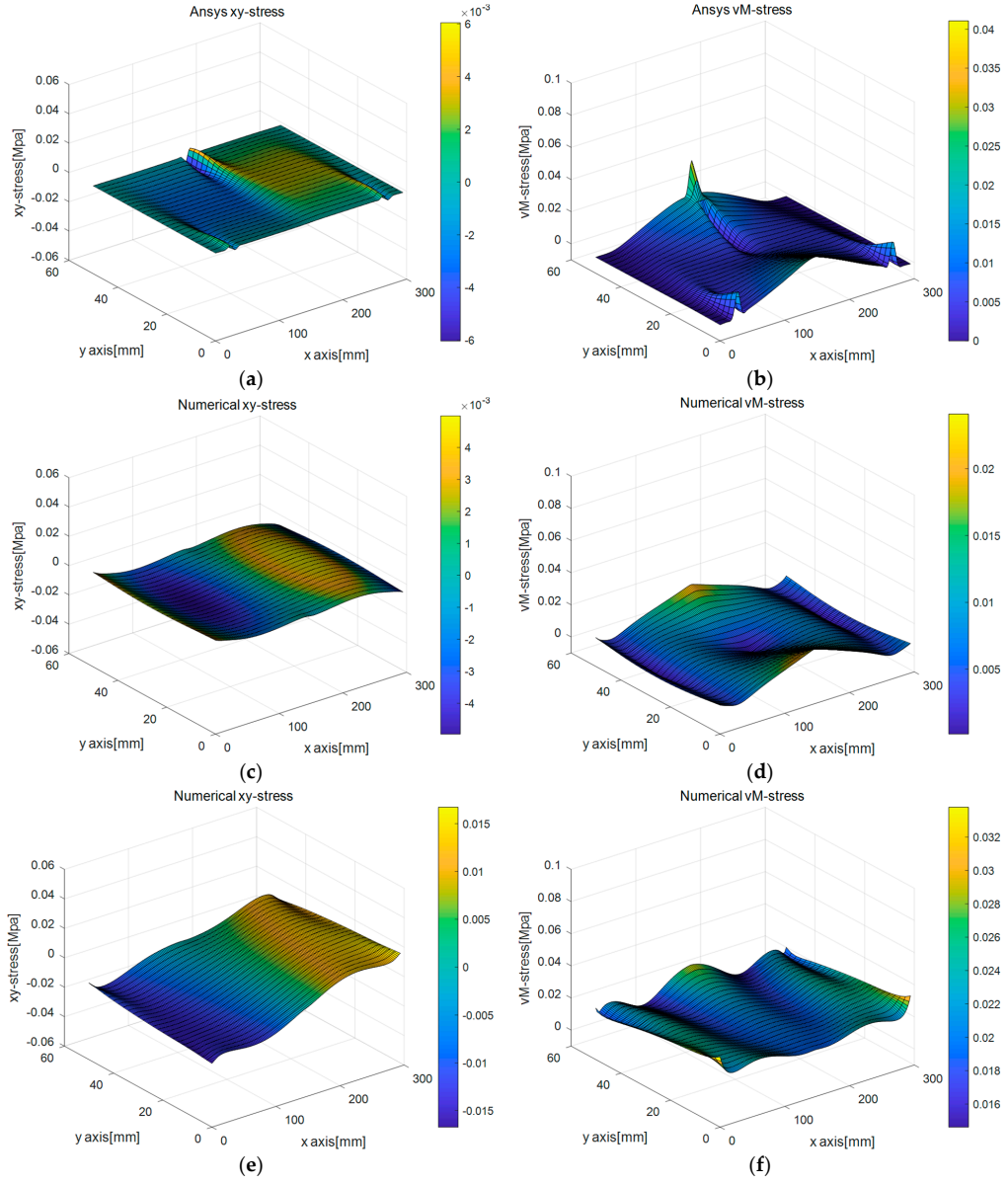
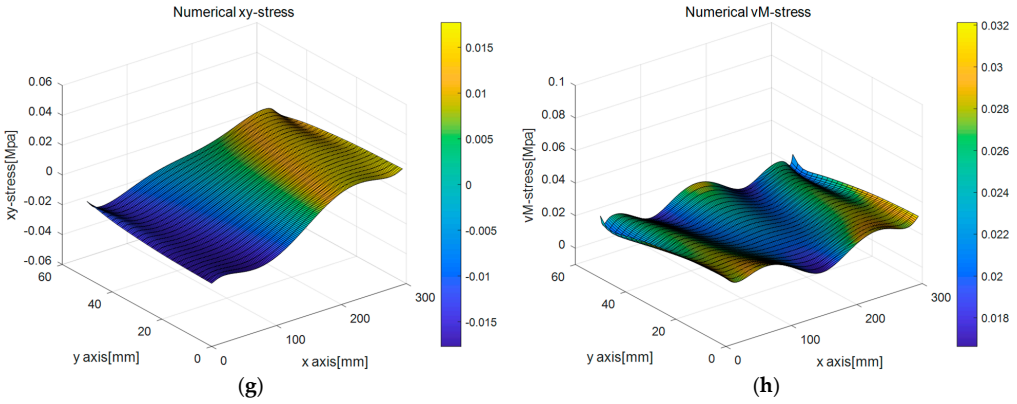


Figure 10. *Cont.*



(**g**) (**h**)

Figure 10. Surface plots for computed Cauchy and von Mises stress by different simulation methods and boundary capturing methods for 3-point rubber beam bending test: (**a**) σ_{xy} (Method 1), (**b**) von Mises stress (Method 1), (**c**) σ_{xy} (Method 2), (**d**) von Mises stress (Method 2), (**e**) σ_{xy} (Method 3), (**f**) von Mises stress (Method 3), (**g**) σ_{xy} (Method 4), (**h**) von Mises stress (Method 4).

Table 1. Combinations of analysis methods and boundary capturing techniques.

Method	Analysis Tool	Boundary Capturing Source	Regression Analysis
Method 1	Full ANSYS analysis	No boundary capturing	No regression analysis
Method 2	CPCM	ANSYS analysis result	Linear strain
Method 3	CPCM	DIP and regression analysis	Linear strain
Method 4	CPCM	DIP and regression analysis	Uniform strain

The surface plot for the x-directional displacement result is presented in Figure 8a–d. One reason for the erroneous behaviors found in Figure 8e–g is related to intersecting the regression functions in determining the displacements at surface line ends. This phenomenon has also been reported in the case of processing LiDAR-scanned point cloud data [36]. Although the DIP-based regression analysis has a small chance of overfitting, the slope of the estimated linear regression functions for line 2 and line 3 affects the error. The slope of the linear regression function used for line 2 and line 3 estimation in Methods 3 and 4 may differ from that used for line 2 and 3 in Method 2 since the method utilizes the ANSYS analysis result as boundary capturing source. The real experimental conditions differ from the ideally assumed conditions of ANSYS simulation. For example, in the numerical simulation, the material and the support are assumed to be linear, elastic, and frictionless at the supports. Still, real situations inevitably provoke material nonlinearity, geometric nonlinearity, and slipperiness to some extent [35]. Figure 8e,g show how the regression function for essential boundary assignment negatively affects the kinematic variable computation result; note that the regression function is determined from the intersection points, which are x and y coordinates of the corner points. Figure 8b,d,f,h represent surface plots for the computed y-directional displacement. It is seen that the displacements computed by the developed technique show better resolution than those obtained by DIP using LiDAR point cloud data by Seo et al. [36]. This implies that the edge points extracted from the digital image much better reflect the actual deformed shape of the loaded beam than those obtained from the point cloud data processing.

Figure 9a–h shows the surface plot for the Cauchy stress tensor (σ_{xx}, σ_{yy}) obtained by the aforementioned four different methods. As seen in Figure 9e–h, Methods 3 and

4 yield relatively poor resolution compared to Methods 1 and 2 because the resolution of capturing the essential boundary via digital image or point cloud data combined with the regression process is moderately lower than in the cases employing pure numerical simulation results. However, note that the resolution of boundary capturing is around the submillimeter level, which is practically acceptable for real-scale structural analysis. In addition, using a uniform strain model does not make a remarkable difference in the simulation accuracy. Figure 10a–h gives surface plots for shear stress and von Mises stresses, and it is noticed that they show quite similar trends to those shown in the Cauchy stress results of Figure 9a–h.

Convergence behavior as an error occurrence indicator needs to be investigated since many parts of the developed methods consist of numerical schemes. However, the DIP is difficult to evaluate in terms of the convergence concept. The conventional posteriori error analysis process, often used in computational mechanics, is adopted. The general convergence behavior of the CPCM was well investigated by Lee et al. [32] and Yoon and Song [37]. Here, the convergence rates of the CPCM combined with boundary capturing are examined. It is noted that boundary capturing provokes dominant error behavior because it generates much bigger errors than the CPCM calculation. Captured digital images should adequately reflect the actual deformed boundary of the loaded specimen. The Canny edge detection algorithm extracts sufficient edge points to involve less error and avoid overfitting in the regression process compared to the case of point cloud data processing. For the error estimation, the relative error was calculated in the following manner:

$$E_{L^2} = \frac{\left[\int_\Omega (\boldsymbol{u}_{CPCM} - \boldsymbol{u}_{ex})^T (\boldsymbol{u}_{CPCM} - \boldsymbol{u}_{ex}) d\Omega\right]^{1/2}}{\left[\int_\Omega \boldsymbol{u}_{ex}{}^T \boldsymbol{u}_{ex} d\Omega\right]^{1/2}} \tag{19}$$

where u_{CPCM} denotes the displacement calculated by the CPCM analysis, and u_{ex} refers to the reference displacement calculated from the ANSYS analysis results. Since no exact solution is available, the FE analysis results with a refined mesh were employed as a reference solution.

Figure 11a,b shows the convergence rates for the developed techniques regarding the relative displacement errors in a semi-log scale. Various degrees of the polynomial basis vector for CPCM and the polynomial regression function are considered in the error estimation. In the error calculation, node spacings of 8 mm (total 258 nodes), 6 mm (total 473 nodes), 4 mm (total 1089 nodes), and 2 mm (total 3848 nodes) are used. The second- to fifth-degree polynomials were applied for the regression analysis of the upper and lower surfaces (line 1 and line 4), and the linear and uniform strain models were used for boundary capturing. In Figure 11a,b, the essential boundary condition values are extracted from the ANSYS analysis results and digital images processed regression function, respectively. It is noticed that the error decreases with a moderate slope as the number of nodes increases. The absolute magnitudes of the errors are significantly small when edge points are extracted using the ANSYS analysis results because the ANSYS yields a smoother geometry of the deformed boundary than the digital image of the deformed specimen implemented in regression analysis. In addition, as seen in Figure 11a, the error size is smaller when the linear strain model is applied for the polynomial regression since it can more accurately reflect the curved edge induced by beam bending. However, the linear and uniform strain models decrease trends in the error curves as the degree of polynomial regression increases (see Figure 11b). In addition, the absolute error size significantly decreased for linear and uniform strain models when the fourth-degree polynomial regression was used. Still, there was no remarkable difference between the fourth- and fifth-degree polynomials. Although the convergence rates of the original CPCM show mathematically optimal rates according

to the order of polynomial basis (see [32] and [37]), the developed method does not show a promising convergent behavior because the boundary capturing method dominates the apparent error behavior with a much bigger absolute error size than that of the original CPCM.

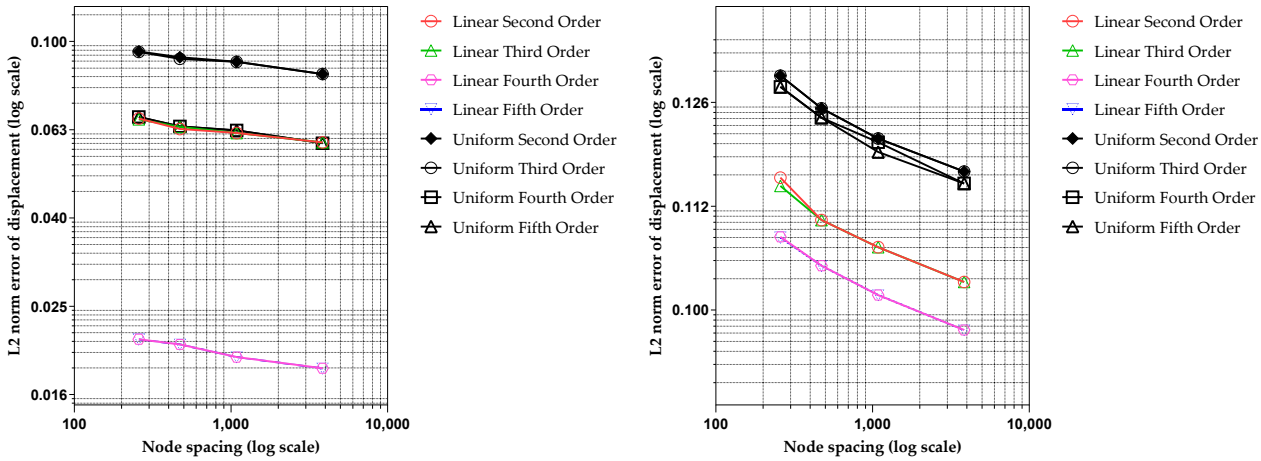


Figure 11. Convergence behavior of L^2 norms error (semi-log scale) for 3-point bending BVP according to various boundary capturing options such as strain model, regression function degree, and boundary capturing data source: (**a**) ANSYS-model-based boundary capturing (Methods 1 and 2); (**b**) digital-image-based boundary capturing (Method 3 and 4).

Regardless of the strain model, increasing the degree of polynomial regression function does not lead to overfitting but stably reduces the error. This phenomenon results from the fact that the edge points extracted from digital images are sufficient and represent a deformed beam shape well. However, it is noteworthy that only a slight difference appears between applying the second-degree polynomial with the linear strain model and the fifth-degree polynomial with the uniform strain model. Namely, when the linear strain model is used, even with a lower degree of the regression function, the error can still become smaller than the uniform strain model combined with the higher regression function. This might imply that the linear strain model can be effectively recommended in the practical boundary capturing process.

4. Conclusions

This study presented an image-driven hybrid structural analysis technique combining DIP for boundary capturing and the CPCM for structural analysis. The technique solves the Dirichlet-type BVP with only essential boundary conditions to analyze a deformed structure. It consists of a boundary capturing method using a DIP-based regression scheme and a strong-form-based particle method called the CPCM. Note that previously, Park et al. [35] proposed a displacement-tracking-based structural analysis where the resolution of the numerical solution depended on the density of the attached target and DIP performance because the CPCM only plays the role of interpolator using the estimated target displacements. In addition, Seo et al. [36] presented a BVP solving scheme equipped with LiDAR point cloud data and using the boundary capturing method, but the final accuracy of the scheme could not be achieved up to the mathematically optimal level because of the tricky point cloud process and the weak soundness of the LiDAR scan data.

On the other hand, to capture the deformed boundary, the new method presented in this study utilizes not the attached target or LiDAR data but the digital image. It effectively improves the boundary capturing resolution, although the basic framework of this study's

DIP technique and regression analysis are still related to those of Park et al. [35] and Seo et al. [36]. As a result, the accuracy of the BVP solving was effectively improved by the image-driven boundary capturing, which naturally led to the achievement of more accurate mechanical variables such as displacement, strain, and stress.

The effects of the regression function order and the strain model type, including the general parametric setting for the CPCM simulation, were investigated through the three-point bending test simulation of a rubber beam. For further verification, the results of the ANSYS analysis were utilized in two ways: as a reference solution for the error estimation of the convergence study and as source data for boundary capturing for comparative study in Section 3.3. The error analysis revealed that boundary capturing for the deformed specimen edge using DIP is rarely sensitive to the order of the regression function. However, in the case of boundary capturing using LiDAR-scanned point cloud data, the resolution was quite sensitive to the integrity of point cloud data [36] because the scatteredness of point cloud data for deformed edge easily makes the polynomial regression analysis quite vulnerable. On the contrary, this study showed that the pixel data derived from digital images of the deformed specimen boundary can be more consistently extracted due to the stable nature of the data source and the manipulative edge detection algorithms. Better resolutions of kinematic variables obtained by the new method were visually verified through the surface plots in Section 3.3. In addition, the convergence study demonstrated the integrity of the process as a numerical simulator. It was shown that a numerically optimal combination of the boundary capturing model and the parametric setting of the CPCM might be quantitatively determined through error analysis.

As a result, the developed methodology, which combines the image-driven regression analysis for boundary capturing with the CPCM for Dirichlet-type BVP solving, is new. Since the method enables simultaneous displacement measurement and structural analysis, it might provide new ideas for future structural health monitoring for real-scale infrastructures like concrete bridges. Furthermore, combined with deep learning techniques, the method can be dramatically improved and is expected to be applied to various engineering fields.

Author Contributions: Conceptualization, K.-W.S., J.P. and Y.-C.Y.; methodology, K.-W.S., J.P. and Y.-C.Y.; validation, K.-W.S., S.I.P., J.-H.S. and Y.-C.Y.; formal analysis, J.P., J.-H.S. and Y.-C.Y.; investigation, K.-W.S., J.P. and S.I.P.; resources, K.-W.S. and Y.-C.Y.; data curation, K.-W.S., J.P., S.I.P. and Y.-C.Y.; writing—original draft preparation, K.-W.S. and J.P.; writing—review and editing, K.-W.S., S.I.P., J.-H.S. and Y.-C.Y.; visualization, K.-W.S.; supervision, J.-H.S. and Y.-C.Y.; project administration, Y.-C.Y.; funding acquisition, J.-H.S. and Y.-C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by a National Research Foundation of Korea (NRF) grant funded by the Republic of Korea government (MSIT), grant number 2021R1F1A104825913, and the fourth author (JH) acknowledges support from the Department of Energy, National Nuclear Security Administration, Predictive Science Academic Alliance Program (PSAAP), under award number DE-NA0003962.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Glisic, B. Concise Historic Overview of Strain Sensors Used in the Monitoring of Civil Structures: The First One Hundred Years. *Sensors* **2022**, *22*, 2397. [CrossRef] [PubMed]
2. Wang, G.; Ke, J. Literature Review on the Structural Health Monitoring (SHM) of Sustainable Civil Infrastructure: An Analysis of Influencing Factors in the Implementation. *Buildings* **2024**, *14*, 402. [CrossRef]
3. Frangopol, D.M.; Soliman, M. Life-cycle of structural systems: Recent achievements and future directions. *Struct. Infrastruct. Eng.* **2015**, *12*, 1–20. [CrossRef]
4. Kot, P.; Muradov, M.; Gkantou, M.; Kamaris, G.S.; Hashim, K.; Yeboah, D. Recent Advancements in Non-Destructive Testing Techniques for Structural Health Monitoring. *Appl. Sci.* **2021**, *11*, 2750. [CrossRef]
5. Hassani, S.; Mousavi, M.; Gandomi, A.H. Structural Health Monitoring in Composite Structures: A Comprehensive Review. *Sensors* **2021**, *22*, 153. [CrossRef]
6. Feng, D.; Feng, M.Q. Vision-based multipoint displacement measurement for structural health monitoring. *Struct. Control. Health Monit.* **2016**, *23*, 876–890. [CrossRef]
7. Lynch, J.P. An overview of wireless structural health monitoring for civil structures. *Philos. Trans. A Math. Phys. Eng. Sci.* **2007**, *365*, 345–372. [CrossRef]
8. Payawal, J.M.G.; Kim, D.-K. Image-Based Structural Health Monitoring: A Systematic Review. *Appl. Sci.* **2023**, *13*, 968. [CrossRef]
9. Zhuang, Y.; Chen, W.; Jin, T.; Chen, B.; Zhang, H.; Zhang, W. A Review of Computer Vision-Based Structural Deformation Monitoring in Field Environments. *Sensors* **2022**, *22*, 3789. [CrossRef]
10. Xu, Y.; Brownjohn, J.; Kong, D. A non-contact vision-based system for multipoint displacement monitoring in a cable-stayed footbridge. *Struct. Control. Health Monit.* **2018**, *25*, e2155. [CrossRef]
11. Khuc, T.; Catbas, F.N. Completely contactless structural health monitoring of real-life structures using cameras and computer vision. *Struct. Control. Health Monit.* **2017**, *24*, e1852. [CrossRef]
12. Sohn, H.; Farrar, C.R.; Hemez, F.M.; Czarnecki, J.J. A Review of Structural Health Review of Structural Health Monitoring Literature 1996-2001. In Proceedings of the 3rd World Conference on Structural Control, Como, Italy, 7–12 April 2002.
13. Al-Adly, A.I.F.; Kripakaran, P. Physics-informed neural networks for structural health monitoring: A case study for Kirchhoff–Love plates. *Data-Centric Eng.* **2024**, *5*, e6. [CrossRef]
14. Farrar, C.R.; Worden, K. *Structural Health Monitoring: A Machine Learning Perspective*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
15. Conde-Carnero, B.; Riveiro, B.; Arias, P.; Caamaño, J.C. Exploitation of Geometric Data provided by Laser Scanning to Create FEM Structural Models of Bridges. *J. Perform. Constr. Facil.* **2016**, *30*, 04015053. [CrossRef]
16. Sun, Y.; Pang, J.H.L.; Wong, C.K.; Su, F. Finite element formulation for a digital image correlation method. *Appl. Opt.* **2005**, *44*, 7357–7363. [CrossRef]
17. Lu, Y.; Zhu, W. Convolution finite element based digital image correlation for displacement and strain measurements. *Comput. Methods Appl. Mech. Eng.* **2024**, *419*, 116597. [CrossRef]
18. Jo, H.C.; Sohn, H.-G.; Lim, Y.M. A LiDAR Point Cloud Data-Based Method for Evaluating Strain on a Curved Steel Plate Subjected to Lateral Pressure. *Sensors* **2020**, *20*, 721. [CrossRef]
19. Yue, Z.Q.; Chen, S.; Tham, L.G. Finite element modeling of geomaterials using digital image processing. *Comput. Geotech.* **2003**, *30*, 375–397. [CrossRef]
20. Duarte, A.P.C.; Silva, B.A.; Silvestre, N.; de Brito, J.; Júlio, E. Mechanical characterization of rubberized concrete using an image-processing/XFEM coupled procedure. *Compos. Part B Eng.* **2015**, *78*, 214–226. [CrossRef]
21. Wu, Z.; Ji, X.; Liu, Q.; Fan, L. Study of microstructure effect on the nonlinear mechanical behavior and failure process of rock using an image-based-FDEM model. *Comput. Geotech.* **2020**, *121*, 103480. [CrossRef]
22. Jo, H.C.; Kim, J.; Lee, K.; Sohn, H.-G.; Lim, Y.M. Non-contact strain measurement for laterally loaded steel plate using LiDAR point cloud displacement data. *Sens. Actuators A Phys.* **2018**, *283*, 362–374. [CrossRef]
23. Cabaleiro, M.; Riveiro, B.; Arias, P.; Caamano, J.C. Algorithm for beam deformation modeling from LiDAR data. *Measurement* **2015**, *76*, 20–31. [CrossRef]
24. Ghahremani, K.; Khaloo, A.; Mohamadi, S.; Lattanzi, D. Damage Detection and Finite-Element Model Updating of Structural Components through Point Cloud Analysis. *J. Aerosp. Eng.* **2018**, *31*, 04018068. [CrossRef]
25. Jafari, B.; Khaloo, A.; Lattanzi, D. Deformation Tracking in 3D Point Clouds Via Statistical Sampling of Direct Cloud-to-Cloud Distances. *J. Nondestruct. Eval.* **2017**, *36*, 65. [CrossRef]
26. Banerjee, P.K.; Butterfield, R. *Boundary Element Methods in Engineering Science*; McGraw-Hill Book Co. (UK): London, UK, 1981.
27. Samaan, M.F.; Rashed, Y.F. BEM for transient 2D elastodynamics using multiquadric functions. *Int. J. Solids Struct.* **2007**, *44*, 8517–8531. [CrossRef]
28. Belytschko, T.; Lu, Y.Y.; Gu, L. Element-free Galerkin methods. *Int. J. Numer. Methods Eng.* **1994**, *37*, 229–256. [CrossRef]
29. Belytschko, T.; Krongauz, Y.; Organ, D.; Fleming, M.; Krysl, P. Meshless methods: An overview and recent developments. *Comput. Methods Appl. Mech. Eng.* **1996**, *139*, 3–47. [CrossRef]

30. Li, Q.; Wang, Z.; Li, L.; Hao, H.; Chen, W.; Shao, Y. Machine learning prediction of structural dynamic responses using graph neural networks. *Comput. Struct.* **2023**, *289*, 107188. [CrossRef]
31. Shao, Y.; Li, L.; Li, J.; Li, Q.; An, S.; Hao, H. 3D displacement measurement using a single-camera and mesh deformation neural network. *Eng. Struct.* **2024**, *318*, 118767. [CrossRef]
32. Oñate, E.; Idelsohn, S.; Zienkiewicz, O.C.; Taylor, R.L.; Sacco, C. A stabilized finite point method for analysis of fluid mechanics problems. *Comput. Methods Appl. Mech. Eng.* **1996**, *139*, 315–346. [CrossRef]
33. Aluru, N.R. A point collocation method based on reproducing kernel approximations. *Int. J. Numer. Methods Eng.* **2000**, *47*, 1083–1121. [CrossRef]
34. Lee, S.-H.; Yoon, Y.-C. Meshfree point collocation method for elasticity and crack problems. *Int. J. Numer. Methods Eng.* **2004**, *61*, 22–48. [CrossRef]
35. Park, J.; Kim, K.-H.; Yoon, Y.-C.; Lee, S.-H. Experiment-Based Synthetic Structural Analysis Combining Digital Image Processing and a Strong Form Meshfree Method. *Appl. Sci.* **2020**, *10*, 8053. [CrossRef]
36. Seo, K.-W.; Yoon, Y.-C.; Lee, S.-H. LiDAR Point Cloud Data Combined Structural Analysis Based on Strong Form Meshless Method Using Essential Boundary Condition Capturing. *Sensors* **2023**, *23*, 6063. [CrossRef] [PubMed]
37. Yoon, Y.-C.; Song, J.-H. Extended particle difference method for weak and strong discontinuity problems: Part I. Derivation of the extended particle derivative approximation for the representation of weak and strong discontinuities. *Comput. Mech.* **2014**, *53*, 1087–1103. [CrossRef]
38. Yoon, Y.-C.; Schaefferkoetter, P.; Rabczuk, T.; Song, J.-H. New strong formulation for material nonlinear problems based on the particle difference method. *Eng. Anal. Bound. Elem.* **2019**, *98*, 310–327. [CrossRef]
39. Bradski, G. The openCV library. *Dr. Dobb's J. Softw. Tools* **2000**, *120*, 122–125.
40. Zhou, Q.-Y.; Park, J.; Koltun, V. Open3D: A Modern Library for 3D Data Processing. *arXiv* **2018**, arXiv:1801.09847.
41. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]
42. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.