Segmentation and Quantification of Surface Defects in 3D Reconstructions for Damage Assessment and Inspection

Jonathan Sterckx, Michiel Vlaminck, and Hiep Luong

***Abstract*—Detecting surface defects is crucial for maintaining the integrity of critical infrastructure. Traditional RGB image-based methods are limited by their reliance on 2D information, which impairs accurate damage assessment. This paper introduces a novel approach that enhances defect detection and quantification, utilizing dense 3D reconstructions generated through techniques like photogrammetry or profilometry. We develop an improved robust spline fitting algorithm to estimate the undamaged surfaces from the 3D reconstructions. The residual distances between the observed and fitted surfaces are subsequently used to segment and quantify defects. By leveraging 3D data, our method resolves visual ambiguities and enables damage quantification using physically meaningful metrics. For 3D models based on optical sensing, our method complements RGB image-based defect detectors and classifiers, facilitating the fusion of visual and 3D information for a more comprehensive defect analysis. Validated on both synthetic and real-world datasets, our method demonstrates strong performance and practical feasibility.**

***Note to Practitioners*—Our research is driven by the growing potential of drone-based inspections using high-resolution imaging platforms, which offer significant advantages for monitoring remote or hard-to-reach infrastructure. With high-quality images, we can use photogrammetry to reconstruct accurate 3D models directly from inspection images, without requiring additional sensors or manual intervention. We leverage this 3D data to improve the robustness of automated defect detection and enable the precise quantification of defect sizes and material loss. Tracking these metrics over multiple inspections can provide valuable information for preventive and predictive maintenance, moving us closer to efficient, comprehensive structural health monitoring. While our method is highly effective across various surfaces, it is less suited to detecting widespread shallow damage, which could benefit from incorporating additional geometric constraints.**

***Index Terms*—Surface defects, 3D reconstruction, damage assessment, automated inspection.**

I. INTRODUCTION

THE automated detection of surface defects plays a crucial role in maintaining the safety and integrity of critical infrastructure, such as bridges [1], buildings [2], roads [3], power plants [4], and wind turbine blades [5]. As these structures age or are exposed to harsh operational conditions, they become increasingly vulnerable to damage that can impair their functionality and safety [6]. Early identification and accurate quantification of surface defects – such as cracks, spalling, pitting, and erosion – are vital for effective preventive and predictive maintenance, cost-efficient repairs, and ensuring long-term operational reliability [7].

The use of techniques based on RGB images (hereafter referred to simply as images) for identifying damage has gained significant popularity due to their ability to provide clear visual evidence of defects [8]. Furthermore, drones equipped with high-resolution cameras can capture detailed images of hard-to-reach remote locations, enhancing the accessibility and quality of the visual data [9]. However, image-based methods face significant challenges in defect detection and quantification. Variations in lighting, surface textures, and shadows can lead to misclassification of defects, while the reliance on 2D information limits the ability to assess depth and material loss accurately.

A fundamental drawback of image-based methods is their inability to capture the full extent of damage in three dimensions, making it difficult to measure defect volume or progression over time. However, with the advent of advanced imaging technologies and photogrammetry workflows – which reconstruct 3D structures from overlapping 2D images [10] – there is an opportunity to overcome these limitations without the need for additional specialized sensors.

In addition to enabling the calculation of physically meaningful 3D metrics, such as defect depth and material loss, incorporating 3D data could help to differentiate actual defects from visual artifacts like surface discolorations, improving detection accuracy. However, most existing 3D-based approaches prioritize anomaly detection [11], [12], [13], estimating the probability of each point in the 3D model being an anomaly. Such methods lack precise defect delineation and quantification, limiting their value for maintenance and repair decision-making. While anomaly detection identifies potential issues, defect quantification provides geometrically meaningful metrics that allow for precise tracking over multiple inspections and support the refinement of predictive models.

Quantifying defects requires a reference surface to compare the observed surface against. In this work, we assume the reference surface is unknown (i.e., no CAD model is available) and needs to be estimated. The primary challenge in estimating

---

Received 6 November 2024; revised 4 April 2025; accepted 27 July 2025. Date of publication 30 July 2025; date of current version 11 August 2025. This article was recommended for publication by Associate Editor C.-Y. Hsu and Editor B. Vogel-Heuser upon evaluation of the reviewers' comments. This work was supported by the Energy Transition Fund (ETF) of the Belgian Federal Government (BLEEPID-Project; Federal Public Service (FPS) Economy Belgium). *(Corresponding author: Jonathan Sterckx.)*

The authors are with the Department of Telecommunications and Information Processing, Ghent University, 9000 Ghent, Belgium (e-mail: jonathan.sterckx@ugent.be).

This article has supplementary downloadable material available at https://doi.org/10.1109/TASE.2025.3593967, provided by the authors.

Digital Object Identifier 10.1109/TASE.2025.3593967

this surface lies in distinguishing regular surface curvature from defect-induced variations in the observed surface. Additionally, adapting real-world data for reliable defect metric computation requires special consideration. To address these challenges, we introduce a tailored pipeline for segmenting and quantifying damage in dense 3D models, including those generated through photogrammetry workflows.

When 3D reconstructions are generated using optical tools like photogrammetry or profilometry, our workflow seamlessly integrates with existing image-based defect detectors and classifiers by applying them to the source images. This compatibility enables the fusion of 2D and 3D methods, enhancing defect segmentation, classification, and quantification.

The key contributions of this work are as follows.

- We estimate an object's undamaged surface from its 3D reconstruction by customizing the ISO-standard robust spline filter [14], an iteratively reweighted fitting method. We introduce a varying smoothing parameter and a sigmoid-based weighting function applied to point-to-plane residuals, enabling adaptive fitting to natural surface curvatures while reducing overfitting to defects.
- We specify how to leverage the ordered structure of 3D models from standard reconstruction techniques, allowing the 3D residuals to be represented on a 2D grid. This approach allows the use of efficient, well-established 2D segmentation tools for 3D defect segmentation.
- We outline the computation of a comprehensive set of practically relevant geometric defect metrics, such as volumetric material loss and affected surface area.
- We present the integration of image-based and 3D-based defect segmentation on real-world data, showcasing how their fusion improves the accuracy of damage segmentation and quantification.

We validate our method through a quantitative assessment of defect segmentation and quantification on an artificial dataset, demonstrating robust performance across diverse surfaces and defect types. To highlight its practical application, we apply our approach to two real-world cases: a photogrammetry-derived 3D reconstruction of a wind turbine blade with leading edge erosion and a profilometry-based reconstruction of a pitted and spalled gear.

II. RELATED WORK

A. 2D-Based Methods

Initial surface defect detection methods predominantly relied on conventional image analysis techniques. For instance, the Sobel edge detector and Canny edge detector have been employed for crack detection [15]. Additionally, methods like the Hough transform [16] and wavelet analysis [17] are used to extract features indicative of various types of structural damage. Despite their usefulness, these traditional approaches require manual feature extraction and are limited in their ability to handle diverse damage types and varying lighting conditions. Issues such as image blurriness, shadows, and uneven lighting can further impair the effectiveness of these techniques.

To address these challenges, deep learning-based methods were introduced for damage detection [18], [19]. Unlike traditional methods, deep learning approaches automatically extract and represent features through multiple non-linear layers, offering improved robustness in many tasks. However, these methods are not immune to problems like poor generalization to new data with different properties. Specifically, the visual differences between surface discoloration, shadows, and actual defects may vary widely across different use cases.

B. 3D-Based Methods

Recent research has increasingly focused on leveraging 3D data for anomaly detection in point clouds and 3D models. Bergmann et al. proposed 3D-ST, an unsupervised anomaly detection and localization method for 3D point clouds. This model, which uses a student-teacher architecture with a self-supervised pre-training approach sensitive to local rotations, achieves strong results on the MVTec 3D-AD dataset. However, the method's high VRAM requirements make it impractical for processing detailed point clouds [11]. Horwitz and Hoshen introduced BTF, a method that combines handcrafted 3D representations (FPFH) with a deep color-based approach (PatchCore) for anomaly detection and segmentation in RGB and 3D data. While BTF achieves state-of-the-art performance, it lacks the capacity to handle high-resolution point clouds effectively [20]. Liu et al. presented Real3D-AD, a high-precision anomaly detection dataset for point clouds, and Reg3D-AD, a registration-based anomaly detection method that uses a feature memory bank. Despite its promising performance, this approach requires defect-free reference samples, limiting its applicability to cases with known, undamaged baselines [12]. Cao et al. proposed CPMF, which combines local geometric descriptors with global semantic information from multi-view 2D projections to enhance point cloud anomaly detection. This fusion of 3D and 2D data offers strong detection capabilities but remains constrained by its reliance on a memory bank [13].

3D anomaly detection assigns a confidence score to each point, indicating the likelihood of a defect. While useful, many applications may require quantifying defect characteristics (e.g., area, depth, material loss) for maintenance decisions. Few studies, to our knowledge, focus on 3D defect quantification. Beckman et al. developed a method to detect and quantify concrete spalling using Faster R-CNN paired with a depth camera. This approach enables segmentation and volume estimation of spalling; however, the depth camera's low resolution limits accuracy, and the setup is impractical for remote inspections of inaccessible areas. Additionally, the method is restricted to flat surfaces, reducing its applicability in real-world infrastructure [21]. Guerra et al. proposed a method to characterize and quantify surface imperfections on 3D models using geometric and topological descriptors. Although effective in a single case study, this approach relies on polynomial surface representations and simple thresholding for defect segmentation, which limits its flexibility and generalizability for more complex surfaces and diverse defect types [22].

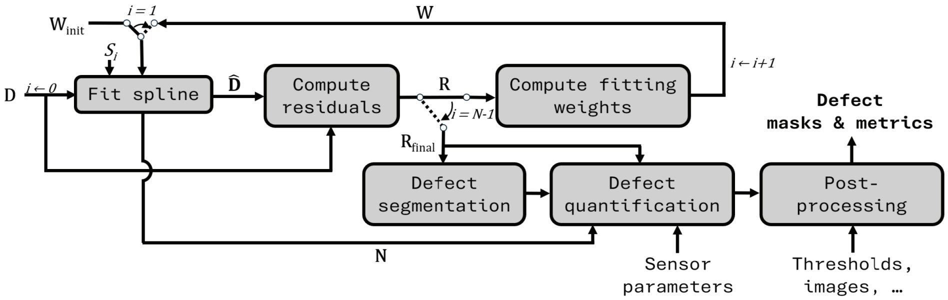


Fig. 1. Pipeline overview: A smoothing spline is fitted to a depth map **D** using initial weights $\mathbf{W}_{\text{init}}$, producing the estimated defect-free depth map $\hat{\mathbf{D}}$. Residuals **R** between observed and fitted models update fitting weights **W** for N iterations, with varying smoothing parameter S_i. The final residual map $\mathbf{R}_{\text{final}}$ is segmented into a binary defect mask. Each defect is quantified using $\mathbf{R}_{\text{final}}$, normal map **N**, and sensor parameters. Optional post-processing removes false detections or refines results.

This paper addresses shortcomings of the works of Beckman et al. and Guerra et al. by adopting a more flexible surface representation beyond flat or polynomial models. We introduce a pipeline designed to address the added complexities this flexibility brings, along with a segmentation approach that surpasses the binary thresholding of Guerra et al. and a structured method for computing defect metrics from depth maps with non-uniform spatial coverage.

III. METHODS

The quantitative analysis of surface defects on an inspected object involves the following steps. First, the object's original, undamaged state is estimated based on a dense 3D reconstruction of the damaged object. Then, the residual distances between the 3D model of the damaged object and its estimated undamaged counterpart are computed. Next, regions with significant residuals, which indicate surface defects, are segmented. Finally, for each identified defect, key metrics such as depth, surface area, and the volume of missing material are quantified. Figure 1 gives an overview of the workflow.

If a CAD model is available, residual distances can be computed by aligning the observed model using a robust trimmed ICP algorithm [23], [24]. Surface imperfections can then be quantified via residual distance calculations. However, we focus on the more challenging case where no ground truth exists or the CAD model is inaccurate (e.g., due to stress-induced deformation).

A. Acquisition & Data Structure of the 3D Models

The 3D models used in this work are assumed to be ordered, meaning they can be represented as a structured 2D grid where each (row, column) index corresponds to a fixed spatial location, and depth values are stored in a regular format, such as a depth map. This structure enables efficient processing techniques like bivariate spline fitting and convolution-based filtering, which are difficult to apply to unstructured point clouds. Two common methods producing structured 3D data are photogrammetry and profilometry.

A photogrammetry pipeline reconstructs 3D surfaces from multiple 2D images. Initially, a sparse 3D reconstruction is obtained by matching keypoints between images, followed by the joint estimation of the camera poses and the triangulated 3D position of the matched keypoints. Then, dense stereo matching produces depth maps – which provide the distance from the 3D point captured in each image pixel to the image plane – which naturally align with the image grid. Given a depth map $\mathbf{D} \in \mathbb{R}^{H \times W}$, where each element d_{vu} corresponds to the depth at pixel coordinates (u, v) (with $u \in \{1, \ldots, W\}$ and $v \in \{1, \ldots, H\}$) and a 3×3 camera intrinsic matrix **K**, the 3D point in camera coordinates corresponding to pixel (u, v) is obtained using the function ϕ, defined in Equation 1.

$$\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad (u, v) \mapsto d_{vu}\mathbf{K}^{-1}(u, v, 1)^T \tag{1}$$

The set of all 3D points obtained from the depth maps of a 3D reconstruction constitutes the point cloud. Since 3D points can also be projected back into image space using standard projection formulas, depth maps and point clouds are mathematically equivalent representations of the same geometric information. For further details, we refer to [25].

Techniques like structured light scanning or laser profilometry capture depth in a fixed 2D array, directly forming an ordered depth map.

When appropriate preprocessing is applied, other sources like LiDAR can also be structured into a grid-like format, enabling similar processing techniques.

B. Estimating the Original Surface by a Smoothing Spline

Since most objects cannot be described by a simple parametric equation, we opt for non-parametric model fitting. Given the widespread use of splines in the design and modeling of various engineered surfaces [26], [27], [28], we choose to recover the original surface by fitting a bivariate smoothing spline $f(u, v)$ to the depth map representation **D** of the observed 3D model of the inspected object. If the 3D model consists of multiple depth maps, they can be processed sequentially and fused in overlapping regions, though this work does not focus on the fusion step.

Splines are piecewise polynomials that provide a flexible way to model complex surfaces. They are formed by fitting polynomial segments between specific points called knots, ensuring continuity and smoothness across these segments. The placement of knots determines the shape and smoothness of the spline, allowing it to adapt to the surface details [29].

A smoothing spline is constructed by placing knots until the desired level of smoothness is achieved. This process is automated by the FITPACK algorithm [30], which iteratively adds knots until the following inequality is satisfied:

$$\sum_{v=1}^{H}\sum_{u=1}^{W} w_{vu}(d_{vu} - f(u,v))^2 \leq S \tag{2}$$

Here, S is the smoothing parameter, and w_{vu} determines the fitting weight assigned to the point represented by pixel (u, v) in the depth map. The weight map $\mathbf{W}$ is defined as the matrix of fitting weights, structured to align with the depth map.

An appropriate choice of S depends on the data: setting it too low causes overfitting, while a high value results in excessive smoothing. To recover the object's original surface, S should capture global curvature while avoiding defect irregularities. However, tuning S alone is not enough; an iterative approach is essential for minimizing defect influence while ensuring the spline fits healthy surface points.

C. Improvements to Robust Spline Fitting

To fit a spline while excluding defects, we use an iterative approach with a robust weighting function to minimize outlier influence. Initially, all data points are weighted equally, identifying outliers with large residuals. In each iteration, we adjust weights based on residuals, progressively down-weighting defect points to refine the fit. This method, known as a robust spline filter in ISO standards [14], helps isolate undamaged regions and exclude surface irregularities. We propose the following modifications to improve its effectiveness for our use case.

1) Variable Smoothing: Unlike traditional robust spline fitting, our approach adapts the smoothing parameter S during each iteration. Initially, S is set to a larger value to prevent overfitting to outliers. As defect points are down-weighted and identified, S is gradually reduced, allowing the spline to fit more precisely to the remaining healthy surface points. This adaptive adjustment ensures the spline evolves from a broad, smooth fit to an accurate representation of the underlying surface while avoiding overfitting to defects.

As a practical heuristic, we can initialize S by sweeping the number of knots (placed uniformly in the (u, v)-grid), fitting a spline for each knot count, and identifying the "knee" [31] in the function relating the sum of squared residuals (SSR) to the number of knots. S is then initialized as the SSR at the knee and scaled by 0.9 for 10 iterations. This approach is detailed and demonstrated in the supplementary material, where its effectiveness is also highlighted. Manual fine-tuning can further refine the reference surface estimation when higher accuracy is required.

2) Computation of Residuals: The residual distance between a 3D data point $\phi(u, v)$ and the fitted spline is typically defined as $|d_{vu} - f(u, v)|$. However, we instead use the signed point-to-plane distance r_{vu}, defined as the signed Euclidean distance between the data point and the closest point on the spline, constrained by the condition that the surface normal at this point intersects the data point.

While this approach is computationally more demanding, it ensures that the residual accurately represents defect depth perpendicular to the fitted surface. This is crucial not only for precise defect quantification but also for maintaining consistency—ensuring that defects of equal depth yield the same residual distance, regardless of their location on the surface.

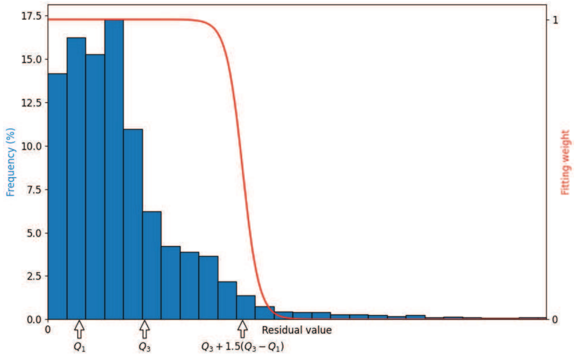


Fig. 2. Example distribution of residuals and its corresponding weighting function.

Equation 3 formulates the absolute point-to-plane distance between a data point $\phi(u, v)$ and the spline surface $\psi(s, t) = f(s, t)\mathbf{K}^{-1}(s, t, 1)^T$.

$$\begin{aligned} |r_{vu}| = &\min_{s,t} \|\phi(u,v) - \psi(s,t)\|_2 \\ &\text{subject to} \quad \big(\phi(u,v) - \psi(s,t)\big)\cdot \mathbf{n}(s,t) = 0, \\ &\text{where} \quad \mathbf{n}(s,t) = \frac{\partial\psi(s,t)}{\partial s} \times \frac{\partial\psi(s,t)}{\partial t} \end{aligned} \tag{3}$$

The residual map $\mathbf{R} \in \mathbb{R}^{H\times W}$ and normal map $\mathbf{N} \in \mathbb{R}^{H\times W\times 3}$ are arrays containing the signed point-to-plane distances (r) and corresponding spline unit normal vectors $(\mathbf{n}/\|\mathbf{n}\|_2)$, respectively, formatted to align with the structure of the depth map. The sign is positive if the z-component of the data point, $\phi_z(u, v)$, exceeds that of the corresponding point on the spline. This sign distinction differentiates between protruding and recessing defects.

3) Weighting Function: A key aspect of our approach is the weighting function, which assigns a fitting weight w_{vu} to each pixel (u, v) in the depth map based on its absolute residual distance $|r_{vu}|$ to the fitted model. Our goal is to have a clear distinction between healthy surface points and defect points by the end of the fitting process, such that we can fit tightly to the healthy surface while minimizing the influence of defect points. To achieve this, we propose a sigmoid-based weighting function that strongly reduces the weight of points with large residuals, effectively excluding them from influencing the spline in subsequent iterations. The corresponding weight $w(r)$ for each point is computed using the sigmoid function, visualized in Figure 2.

$$w(r) = 1 - \left(1 + \exp\left(-\frac{|r| - t}{s}\right)\right)^{-1} \tag{4}$$

Here, t is the center of the sigmoid and s controls the width of the transition band. The parameter t serves as a soft defect threshold and is computed using the interquartile range (IQR) of the residuals, which is robust to outliers [32]. Denoting the 25th and 75th percentiles of the residuals by Q_1 and Q_3 respectively, t is defined as $Q_3 + 1.5(Q_3 - Q_1)$.

We let s decrease as the iterations progress: in the first iterations, we are still unsure whether large residuals represent surface defects or oversmoothing. Therefore, it is important to avoid significantly down-weighting data points that are part of the undamaged surface, as this could be impossible to recover from. As the iterations continue, decreasing s sharpens the distinction between healthy surface points and defects. We heuristically let s vary linearly from $Q_3 - Q_1$ to $0.1(Q_3 - Q_1)$.

If the inspected surface is expected to have either only recessing or only protruding defects, residual signs can override $w(r)$, assigning a weight of 1 to points with an unexpected sign.

A notable alternative method for estimating the defect-free surface is Smooth-Sparse Decomposition (SSD) [33], which decomposes the depth map $\mathbf{D}$ into a smooth background and a sparse anomaly map, both modeled using splines. The supplementary material compares our approach with SSD, demonstrating superior performance.

D. Segmentation of Defects

After estimating the reference surface from the observed 3D model and computing point-to-plane distances, defect segmentation can be performed by identifying regions with large residuals. By leveraging the 2D structure of the 3D data – specifically the residual map $\mathbf{R}$ – we can apply well-established and efficient 2D segmentation techniques to isolate defects, resulting in a discrete set of segmented regions that can then be quantified.

Algorithm 1 Seeded Region Growing for Defect Segmentation

```
Require: Residual map $\mathbf{R}$, thresholds $t_g, t_r$
    $\mathbf{R}' \leftarrow$ Smooth($\mathbf{R}$) ▹Reduce noise
    $P_{\text{ext}} \leftarrow \{(u, v) : r'_{vu}$is local extremum$\}$
    $P_{\text{seed}} \leftarrow P_{\text{ext}} \cap \{(u, v) : |r_{vu}| \geq t_g\}$ ▹Identify seed points
    $\mathbf{M} \leftarrow \mathbf{0}_{H \times W}$ ▹Initialize defect mask
    for all $(u, v) \in P_{\text{seed}}$ do
        $\mathbf{M}_{vu} \leftarrow 1$ ▹Mark seed points
    end for
    for all $(u, v) \in P_{\text{seed}}$ do
        Initialize queue $Q \leftarrow [(u, v)]$ ▹Start with seed point
        while $Q$ is not empty do
            $(u', v') \leftarrow Q$.pop() ▹Get next pixel
            for all neighbor $(\hat{u}, \hat{v})$ of $(u', v')$ do
                if $|r_{\hat{v}\hat{u}}| \geq t_r$ and $m_{\hat{v}\hat{u}} = 0$ then
                    $m_{\hat{v}\hat{u}} \leftarrow 1$ ▹Mark as defect
                    $Q$.push($(\hat{u}, \hat{v})$) ▹Expand to neighbor
                end if
            end for
        end while
    end for
    return $\mathbf{M}$ ▹Segmented defect mask
```

The segmentation of $\mathbf{R}$ employs a seeded region growing technique [34]. First, local extrema greater than t_g are identified as defect seed points. The region growing algorithm then expands each seed point by iteratively including neighboring pixels with residual values greater than t_r. The detailed process is outlined in Algorithm 1, which returns a binary mask $\mathbf{M}$ representing the defect regions.

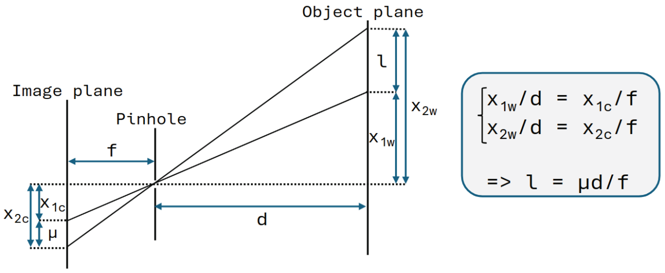


Fig. 3. Side view schematic of a pinhole camera. The law of similar triangles is used to relate the length of a line segment on the image plane, μ, to the length of the corresponding segment on the object plane, l. The surface area of the segment on the object plane corresponding to a pixel with surface area μ^2, denoted A_{pixel}, is given by l^2.

The parameters t_g and t_r are typically chosen manually, but a practical way to estimate suitable values is to apply Otsu's method [35] to the absolute residual map to determine t_g as the threshold that minimizes intra-class variance between defects and the background. The parameter t_r can then be set relative to t_g to adjust the sensitivity of region growing.

E. Quantification of Defects

After segmentation, a connected components analysis [36] is performed to decompose $\mathbf{M}$ into individual binary masks, each representing a single defect. For each defect, we quantify its severity by calculating the surface area of the estimated 3D reference surface impacted by this defect, A, the defect's volume, V, the maximum depth of the defect, r_{max}, and the mean defect depth, r_{mean}.

To compute these quantities for photogrammetry-based 3D reconstructions, we first need to determine the surface area $dA(\mathbf{p})$ of the inspected object's surface captured by each pixel $\mathbf{p} = (u, v)$. We assume a pinhole camera model given that the preceding 3D reconstruction process has corrected for lens distortion [37]. Under this model, the surface area of a planar segment parallel to the image plane at a distance d from it, captured by a single pixel, is denoted as A_{pixel}. This value can be computed using Equation 5, with f the focal length of the camera and μ the width and height of a pixel on the image sensor. The geometry behind this relationship is illustrated in Figure 3.

$$A_{\text{pixel}} = (\mu d / f)^2 \tag{5}$$

If the surface normal to the planar segment forms an angle θ with the camera axis, the surface area captured by a pixel must be adjusted by a scaling factor of $1/\cos(\theta)$.

To generalize these formulae for non-planar surfaces, we only assume local planarity within the area captured by each individual pixel. Given the small pixel sizes in modern cameras and the fact that inspections are typically conducted at close range, this assumption is very reasonable. Consequently, the depth d and the angle of the normal θ are treated as functions of the pixel location $\mathbf{p}$, and, in turn, so is A_{pixel}.

Let $\mathbf{n}(\mathbf{p})$ be the vector extracted from the normal map $\mathbf{N}$ at pixel coordinate $\mathbf{p}$, and let $\mathbf{z}$ be the unit vector along the positive direction of the z-axis, representing the camera axis. To find the scaling factor for the surface area captured by pixel $\mathbf{p}$, we can use the following equation.

$$\frac{1}{\cos(\theta(\mathbf{p}))} = \frac{\|\mathbf{n}(\mathbf{p})\|_2 \cdot \|\mathbf{z}\|_2}{\mathbf{n}(\mathbf{p}) \cdot \mathbf{z}} = \frac{1}{n_z(\mathbf{p})} \tag{6}$$

Then, the next equation can be used to compute the surface area captured by pixel $\mathbf{p}$.

$$\mathrm{d}A(\mathbf{p}) = A_{\text{pixel}}(\mathbf{p}) \;/\; n_z(\mathbf{p}) \tag{7}$$

To compute the defect metrics of an individual defect, we use Equation 8, where P denotes the set of pixels $\mathbf{p}$ that belong to this defect. These equations are equally applicable to non-photogrammetric 3D reconstructions when the appropriate expression for $\mathrm{d}A(\mathbf{p})$ is substituted.

$$\begin{cases} A = \sum_{\mathbf{p}\in P} \mathrm{d}A(\mathbf{p}) & \text{(8a)} \\ V = \sum_{\mathbf{p}\in P} \mathbf{R}(\mathbf{p}) \cdot \mathrm{d}A(\mathbf{p}) & \text{(8b)} \\ r_{\max} = \max_{\mathbf{p}\in P} \mathbf{R}(\mathbf{p}) & \text{(8c)} \\ r_{\text{mean}} = \dfrac{V}{A} & \text{(8d)} \end{cases}$$

F. Post-Processing

Segmented defects can be refined through post-processing to eliminate outliers, such as those with mean defect depth below a threshold. Additionally, visual information can enhance segmentation when damaged areas exhibit distinct colors or textures. Traditional image processing or machine learning techniques [38], [39] can segment these regions, and fusing 3D- and image-based results improves accuracy. Even when image segmentation is unreliable, visual cues can help eliminate false detections. However, the optimal integration of visual information depends on the application and may range from rule-based heuristics to deep learning methods.

Algorithm 2 Fusion of 3D-Based and Visual Segmentation

```
Require: Defect masks $\mathbf{M}_{\text{RGB}}$, $\mathbf{M}_{\text{3D}}$, threshold $t_o$
    $\mathcal{C} \leftarrow$ ConnectedComponents($\mathbf{M}_{\text{RGB}}$)
    $\mathbf{M}_{\text{final}} \leftarrow \mathbf{0}$
    for all $\mathbf{C} \in \mathcal{C}$ do
        if $\sum(\mathbf{C} \wedge \mathbf{M}_{\text{3D}}) / \sum \mathbf{C} \geq t_o$ then
            $\mathbf{M}_{\text{final}} \leftarrow \mathbf{M}_{\text{final}} \vee \mathbf{C}$
        end if
    end for
    return $\mathbf{M}_{\text{final}}$
```

One simple approach to image-based segmentation for defects with distinct colors from the undamaged surface is to first estimate a defect-free "color profile" by applying a median filter [40] to each color channel, as shown in Figure 7. An RGB residual map is then computed by squaring elementwise differences between the input image and the color profile and summing across channels. This highlights deviations due to defects, dirt, or shadows. The RGB residual map is segmented similarly to the 3D-based residual map, forming an image-based defect mask $\mathbf{M}_{\text{RGB}}$. The final defect mask is obtained by fusing the 3D- and image-based masks: each connected component in the image mask is validated against the 3D mask, with overlap exceeding a threshold confirming a surface defect. This fusion approach is used in the first case study in Section IV-B and formalized in Algorithm 2.

Alternatively, visual information can refine 3D-based segmentation without explicit image-based segmentation. For instance, if defects appear darker, detected 3D defects with high mean intensity can be discarded. This simple yet effective filtering is used in the second case study.

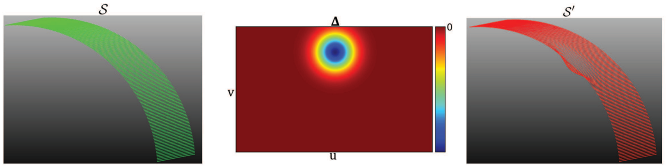


Fig. 4. Generating artificial surface defects on parametric 3D surfaces.

IV. RESULTS

A. Evaluation on Artificial Data

To evaluate our method quantitatively, we can synthetically generate 3D surfaces and apply several types of defects with varying severities. Given a defective surface, we apply our method and evaluate how well we can estimate the original surface, segment, and quantify the defects.

1) Generating Artificial Data: A parametric 3D surface $\mathcal{S}$ can be expressed as follows, where $\mathbf{g} : \mathbb{R}^2 \to \mathbb{R}^3$:

$$\mathcal{S} = \{\mathbf{g}(u,v) \;:\; (u,v) \in [u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}]\} \tag{9}$$

We can apply defects to this surface by generating a defect map, represented by the function $\Delta : \mathbb{R}^2 \to \mathbb{R}$, which controls the deviation from the original surface along its surface normal. Negative values represent dents, whereas positive values represent protruding defects. The damaged surface $\mathcal{S}'$ is constructed as follows, where $\mathbf{n}(u,v)$ denotes the unit normal vector to $\mathcal{S}$ at location $\mathbf{g}(u,v)$:

$$\mathcal{S}' = \{\mathbf{g}(u,v) + \Delta(u,v)\,\mathbf{n}(u,v) : (u,v) \in [u_{\min}, u_{\max}] \times [v_{\min}, v_{\max}]\} \tag{10}$$

In what follows, we will evaluate the surfaces and defect maps on a discrete (u,v)-grid, with $u \in \{1,\ldots,W\}$ and $v \in \{1,\ldots,H\}$. Consequently, both the 3D model of the original surface and the defective surface will consist of HW points. The defect map will be represented by the $H \times W$ matrix $\boldsymbol{\Delta}$. Figure 4 illustrates the relationship between $\mathcal{S}$, $\boldsymbol{\Delta}$ and $\mathcal{S}'$.

Finally, the defect mask $\mathbf{M}$ is defined as the binary $H \times W$ matrix containing a value of 1 at positions where $|\boldsymbol{\Delta}|$ exceeds a specified threshold and 0 elsewhere.

2) Evaluation Metrics: Our method aims to segment and quantify damaged regions by first estimating the original surface and computing the residuals between the observed and fitted model.

After fitting a spline and computing the residuals for each of the HW points, we reshape the HW residuals into a

TABLE I

COMPARISON OF METRICS FOR VARIOUS EXPERIMENTS (EXP.) AND CONFIGURATIONS (CONF.). CONF. 1: SIGMOID WEIGHTING WITH OUR IMPROVED RESIDUALS. CONF. 2: SIGMOID WEIGHTING WITH NAIVE RESIDUALS. CONF. 3: TUKEY WEIGHTING (USING THE IQR OUTLIER THRESHOLD) WITH OUR IMPROVED RESIDUALS. THE BEST RESULT FOR EACH METRIC IN EACH EXPERIMENTS IS HIGHLIGHTED IN BOLD. TO ENSURE A FAIR COMPARISON, THE SMOOTHING AND SEGMENTATION PARAMETERS ARE KEPT CONSTANT WITHIN EACH EXPERIMENT

Exp.	MAE_d			IoU			ΔV_{rel}		
	Conf. 1	Conf. 2	Conf. 3	Conf. 1	Conf. 2	Conf. 3	Conf. 1	Conf. 2	Conf. 3
1a	**0.00060**	0.00336	0.00116	**0.967**	0.919	0.958	**0.01%**	-10.32%	-3.91%
1b	**0.00050**	0.00053	0.00051	**1.000**	**1.000**	**1.000**	**2.19%**	-3.49%	3.26%
1c	**0.00065**	0.00074	0.00084	**0.896**	0.881	0.849	**-4.53%**	-4.72%	-7.18%
2a	**0.00039**	0.00076	0.00705	0.917	**0.957**	0.363	**-5.30%**	-7.25%	14.22%
2b	**0.00025**	0.00029	0.00247	**0.819**	0.817	0.685	**2.57%**	3.97%	39.78%
2c	**0.00221**	0.00298	0.00516	**0.761**	0.688	0.469	**-29.09%**	-37.76%	-53.68%
3a	**0.00074**	0.00199	0.00148	**0.851**	0.773	0.762	**-10.26%**	-26.86%	-12.91%
3b	**0.00075**	0.00082	0.00087	**0.901**	0.887	0.883	**-9.41%**	-9.54%	-10.77%
3c	**0.00061**	0.00069	0.00169	**0.903**	0.886	0.525	-9.88%	-10.78%	**-3.22%**

$H \times W$ residual map $\mathbf{R}$, similar to the procedure outlined in Section III-D.

To evaluate the estimation of the original surface, we consider the mean absolute error (MAE) of the residual map $\mathbf{R}$ compared to the defect map $\mathbf{\Delta}$. To focus specifically on the defect region(s), we denote the MAE within this region as MAE_d. Conversely, the MAE in undamaged regions is denoted as MAE_h. We analyze these values separately, as the healthy region typically covers a much larger area compared to the damaged region. We use Equation 11 to calculate MAE_d. To compute MAE_h, the same equation can be applied, but substituting $\mathbf{M}$ by its complement.

$$\text{MAE}_d = \frac{1}{\|\mathbf{M}\|_1} \sum_{(u,v)} \mathbf{M}(u,v) \cdot |\mathbf{R}(u,v) - \mathbf{\Delta}(u,v)| \tag{11}$$

Ideally, the MAE approaches 0, indicating that the model accurately captures the deviations from the defective surface to the original surface.

We evaluate the segmentation performance using the intersection-over-union (IoU) of the true defect mask, $\mathbf{M}$, and the estimated defect mask $\hat{\mathbf{M}}$, which is obtained by applying our seeded region growing approach to $\mathbf{R}$. The IoU is defined in Equation 12. It ranges between 0 and 1, where 0 represents the case where there is no overlap and 1 represents binary masks that coincide completely.

$$\text{IoU} = |\mathbf{M} \cap \hat{\mathbf{M}}| \,/\, |\mathbf{M} \cup \hat{\mathbf{M}}| \tag{12}$$

To evaluate the damage quantification, we report the relative difference between the estimated damaged volume ($\hat{V}$) and the actual damaged volume (V).

$$\Delta V_{\text{rel}} = (\hat{V} - V) \,/\, V \tag{13}$$

We can calculate V and $\hat{V}$ using Equations 14a and 14b, which are modified from Equation 8b to include the ground truth information about the original surface. This information allows for an accurate assessment of our method: we compute the residuals relative to this reference surface instead of to the fitted spline.

$$\begin{cases} V = \sum_{(u,v)} \mathbf{M}(u,v) \cdot |\mathbf{\Delta}(u,v)| \cdot \mathrm{d}A(u,v) & \text{(14a)} \\ \hat{V} = \sum_{(u,v)} \hat{\mathbf{M}}(u,v) \cdot \mathbf{R}(u,v) \cdot \mathrm{d}A(u,v) & \text{(14b)} \end{cases}$$

3) Experiments: We evaluate the performance of our method on various defect maps applied to three parametric surfaces: a semicylinder, denoted as $\mathcal{S}_1$, representing tubular structures like pipes or cylindrical beams commonly used in fluid transport or structural supports; a V-shaped surface, $\mathcal{S}_2$, representing structures such as square-like bars or beams found in frameworks like transmission towers; and a sinusoidal surface, $\mathcal{S}_3$, which models wavy or corrugated surfaces often encountered in flexible materials or undulating terrains. These surfaces, illustrated in Figure 5, are defined in Equation 15, where $\mathbf{r}_x(30°)$ denotes a matrix that applies a 30° rotation around the x-axis. This rotation, applied to the V-shaped surface, ensures that one half of the surface has a surface normal with a large z-component, while the other half has a small z-component. This asymmetry creates challenges when computing the residuals naively, but our method is designed to handle them effectively.

$$\begin{cases} \mathcal{S}_1 = \{(u, \cos(v), \sin(v))^T : (u,v) \in [0,5] \times [0,\pi]\} \\ \mathcal{S}_2 = \left\{\mathbf{r}_x(30°)(u, v, |v|)^T : (u,v) \in [0,5] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]\right\} \\ \mathcal{S}_3 = \{(u, v, 0.3\sin(2\pi v))^T : (u,v) \in [0,5] \times [0,\pi]\} \end{cases} \tag{15}$$

The surfaces and defect maps were evaluated on a (u,v)-grid with dimensions $H = 314$ and $W = 500$. The applied defect maps $\mathbf{\Delta}$, their corresponding defect masks $\mathbf{M}$ and the results of our method ($\mathbf{R}$ and $\hat{\mathbf{M}}$) are visualized in Figure 5. The estimated defect masks are post-processed by removing segmented defects occupying more than $HW/10$ pixels. Although the defect maps are simplistic, they were chosen to demonstrate the robustness of our method, even in cases where defects are small, shallow, or vary in size and severity.

The evaluation metrics for the experiments are presented in Table I. To highlight the effectiveness of the improvements to the fitting method, outlined in Section III-C, the table also includes metrics from the same experiments performed with naively computed residuals as input to the weighting function. Additionally, it provides performance metrics for the experiments utilizing Tukey's biweight loss function – a robust M-estimator [41] recommended in ISO standards for robust spline fitting [14] – instead of our sigmoid-based weighting.

We also showcase that gradually decreasing the smoothing parameter across the fitting iterations can be beneficial for

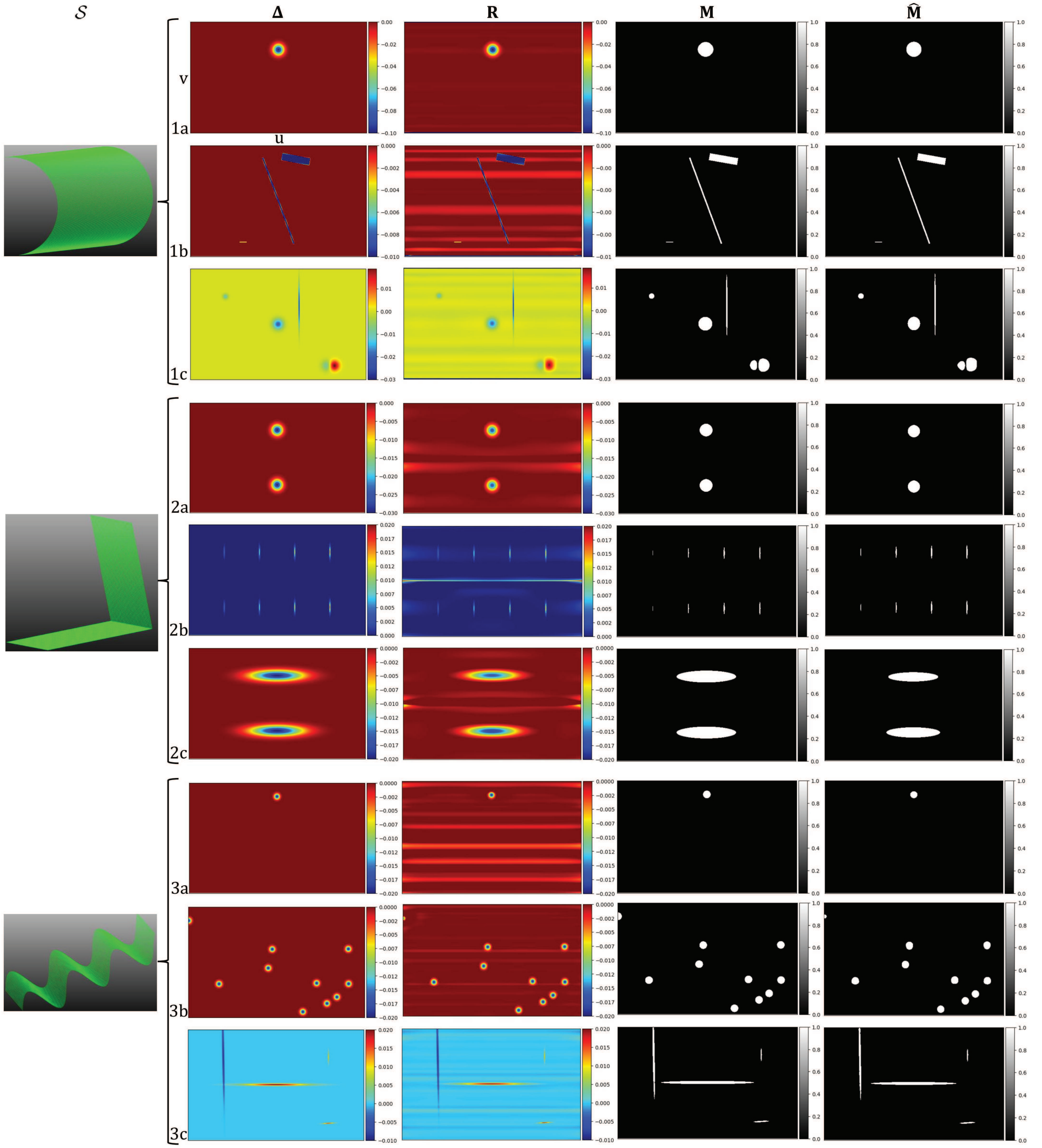
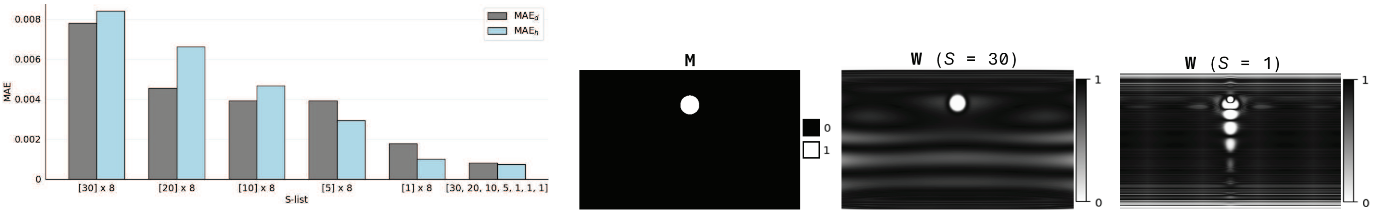


Fig. 5. Results of applying our method on artificial data. Each row represents a different experiment. We evaluate our method on three surface types, each with three distinct defect maps applied. The residual maps in this figure represent signed point-to-plane distances, facilitating comparisons with the applied defect maps.

the estimation of the original surface, and, in turn, also for the subsequent segmentation and quantification of defects. To illustrate this, we apply our fitting algorithm to the defective surface from experiment 1a, utilizing several S-lists as input. The results are presented in Figure 6a. Notably, despite consisting of a combination of the smoothing parameters found in the constant S-lists, the S-list composed of varying elements demonstrates superior performance.

B. *Feasibility for Real-World Data*

To demonstrate the feasibility of our method on real data, we apply it to a photogrammetric 3D reconstruction of a wind



(a) MAE$_d$ and MAE$_h$ for various S-lists. We use the shorthand notation $[X] \times n$ to denote a list with n repetitions of X. We confirmed that $n = 8$ is sufficient for the fitting process to converge.

(b) Left to right: ground truth defect mask $\mathbf{M}$, fitting weight maps $\mathbf{W}$ after one iteration of spline fitting with $S = 30$ and with $S = 1$, respectively.

Fig. 6. Benefits of allowing the smoothing parameter to vary across fitting iterations.

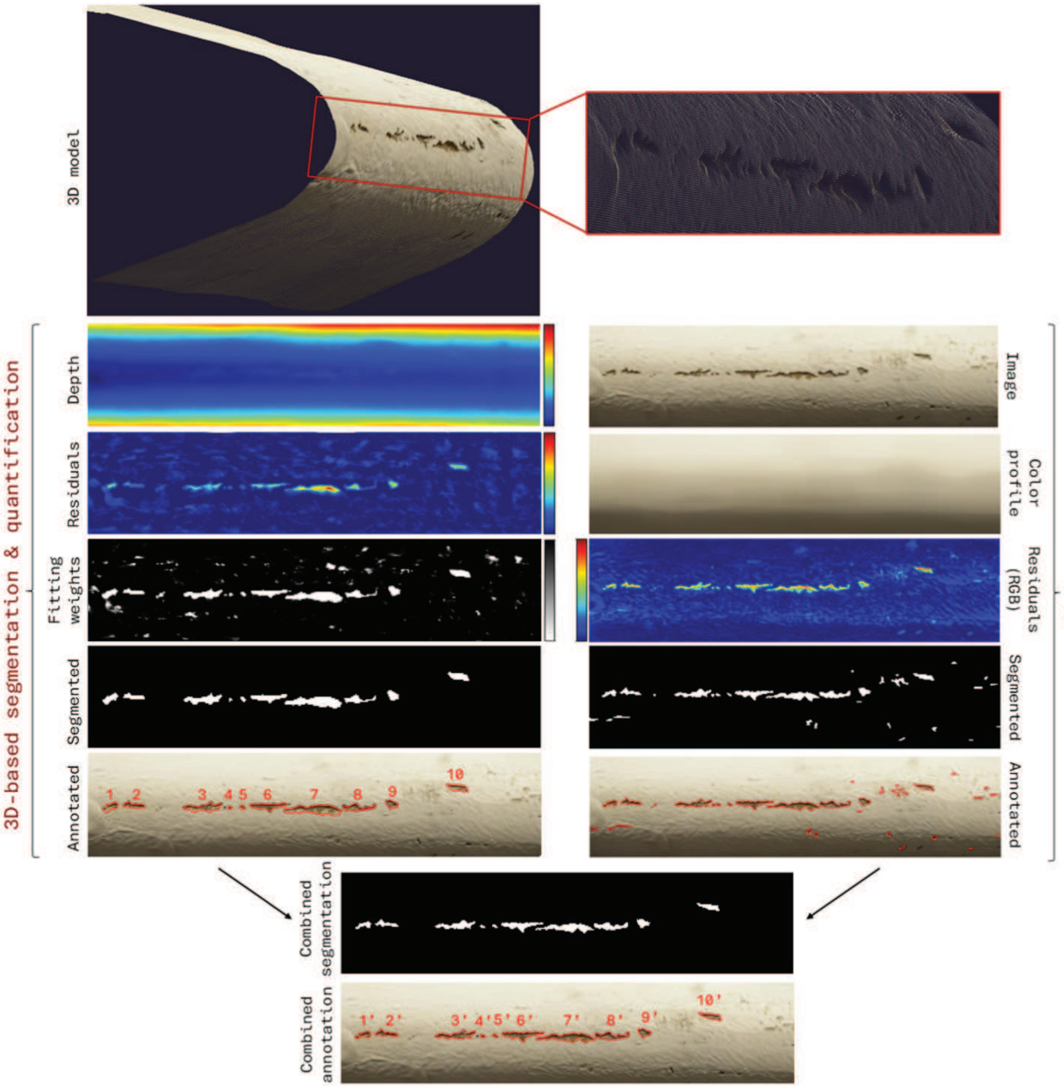


Fig. 7. Results of applying our method on a photogrammetry-based 3D reconstruction of a wind turbine blade. Segmentation and quantification of defects using the 3D information in a dense depth map are shown on the left. The segmentation of defects using visual information in an image is visualized on the right.

turbine blade segment, and to a 3D reconstruction of a gear, obtained through optical profilometry.

1) Leading Edge Erosion on a Wind Turbine Blade: We demonstrate the results of our method applied on a photogrammetry-based 3D reconstruction of a wind turbine blade segment, where the surface shows signs of erosion and grime. The 90 mm-long segment was captured indoors by a high-resolution camera from seven viewpoints at an approximate distance of 0.2 m. Figure 7 presents the 3D model, which was reconstructed by the custom 3D reconstruction pipeline detailed in [42]. This pipeline starts by identifying keypoint track across multiple images by leveraging dense matches

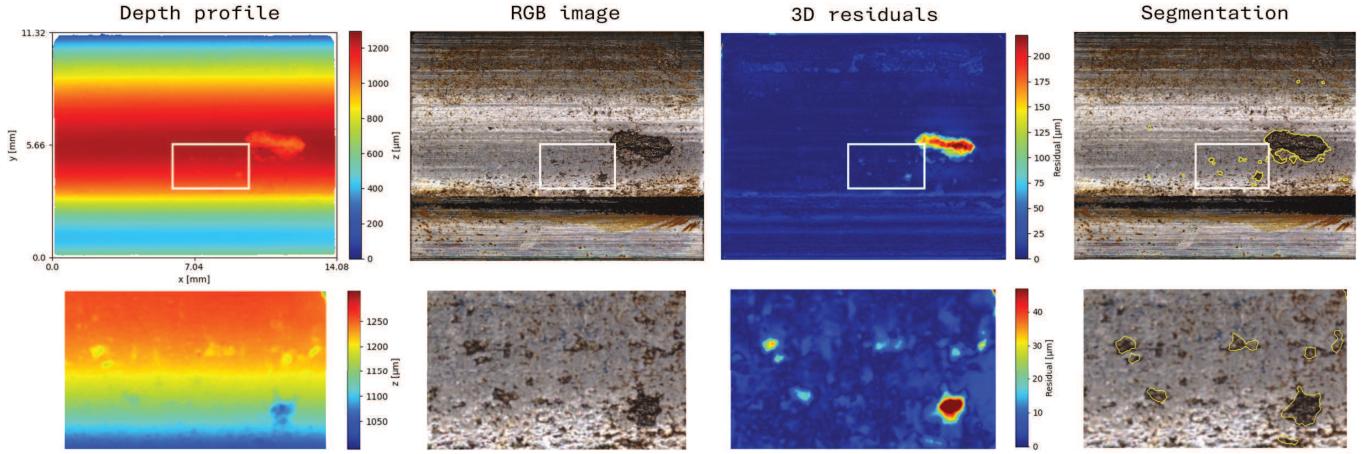


Fig. 8. Results of applying our method on an optical profilometer-based 3D reconstruction of gear. The images in the bottom row show the area outlined by the white box in the top row.

produced by RoMa [43], and passing these keypoint tracks to COLMAP [44] for the incremental reconstruction of the sparse point cloud. The sparse reconstruction is followed by dense reconstruction using 2D Gaussian Splatting [45].

Focusing on erosion, data points with negative signed residuals are assigned a fitting weight of 1 during surface estimation, preventing false positives from protrusions like accumulated paint.

The left half of Figure 7 visualizes the inputs and outputs of the 3D-based defect segmentation and quantification pipeline. The image-based defect segmentation is illustrated on the right. In this and the following experiment, the smoothing and segmentation parameters were fine-tuned through a graphical interface with real-time feedback via interactive sliders.

Fusion of visual and 3D information follows the color profile method in Section III-F. This approach benefits from the sharp edges of RGB-based segmentation while eliminating false positives through 3D-based segmentation.

For defect quantification, we can address the inherent scale ambiguity in photogrammetry outputs if the inspected scene includes markings or objects with known dimensions or if the distance from the camera to the object is known. Here, the camera was 0.2 m from the object's closest point. Depth values are scaled accordingly. We used a Nikon D5300 camera with a sensor pixel area of 15.13 μm^2 and a focal length of 26 mm. With these parameters, we combine Equations 7 and 8 to compute the defect metrics. The largest detected defect (labeled 7′ in Figure 7) measures 9.384 mm^2 in area, 3.301 mm^3 in volume, with a mean depth of 0.351 mm and a maximum defect depth of 0.743 mm. The smallest (labeled 4′) has a surface area of 0.156 mm^2, volume of 0.025 mm^3, mean depth of 0.127 mm and maximum depth of 0.163 mm.

2) Spalling and Pitting on a Gear: Our final experiment involves a 3D reconstruction of a gear with a pitted, spalled surface, scanned using a Keyence VR-5200 optical profilometer. Details on the sample and setup are available in Poletto et al. [46]. Figure 8 shows the input data and our results.

Selecting a segmentation threshold is challenging due to varied pit sizes and shallow pits. To capture all significant defects, we set a low t_g and filter out false positives by discarding defects with a mean image intensity below a specified threshold, as damaged regions appear darker.

For defect quantification, we use d$A(\mathbf{p})$ as defined in Equation 16. There is no depth dependency, as the profilometer captures the surface orthographically. With a spatial resolution of 7.409 μm, A_{pixel} is 54.893 μm^2.

$$\mathrm{d}A(\mathbf{p}) = \frac{A_{\text{pixel}}}{n_z(\mathbf{p})} \tag{16}$$

The largest defect has a surface area of 3.174 mm^2, a volume of 0.332 mm^3 and a maximum depth of 0.256 mm; the smallest one has an area of 0.009 mm^2, a volume of $9 \cdot 10^{-5}$ mm^3 and a maximum depth of 0.014 mm.

V. DISCUSSION

The experiments on artificial data have shown that our method performs well on a variety of surfaces and defect severities. Table I shows the effectiveness of the custom weighting function and the use of point-to-plane residuals. In one experiment, our method does not yield the highest IoU, and in another, it does not achieve the lowest $|\Delta V_{\text{rel}}|$. However, in the first case, the competing method has a significantly higher $|\Delta V_{\text{rel}}|$, and in the second, it has a much lower IoU. In both cases, MAE$_d$ is also substantially greater. The second case may arise when a defect appears smoothed in the residual map, leading to a larger estimated defect area but a reduced maximum depth. This can result in an estimated defect volume that matches the true volume. However, if MAE$_d$ is high, IoU cannot be high while $|\Delta V\text{rel}|$ is low, nor can $|\Delta V_{\text{rel}}|$ be low while IoU is high.

Figure 6 shows the benefits of adaptive smoothing. With $S = 1$, the spline closely approximates the surface but overfits to defects. This produces low residuals everywhere, rendering the fitting weights ineffective as shown in Figure 6b. Using $S = 30$ causes slight oversmoothing but helps quickly identify damaged regions in early iterations. Once these regions are identified and the fitting weights adjusted, reducing the smoothing parameter in later iterations improves surface estimation without overfitting.

We applied our method to a real dataset consisting of an image and a dense, photogrammetry-based depth map of a wind turbine blade segment affected by leading edge erosion. The method successfully segmented all major defects and generated a meaningful residual map. Additionally, we tested it on a profilometer-derived 3D reconstruction of a gear, detecting most pits with significant depth. While no ground truth defect metrics were available, manual inspection of the depth maps shows low residual values in undamaged areas and higher values at apparent defects. This provides qualitative validation, indicating that our model accurately fits the undamaged surface without overfitting to defects. Our experiments demonstrate the method's feasibility and versatility across real-world applications and measurement devices. For quantitative validation on real-world data, future work could involve reconstructing a 3D model of a surface, introducing controlled damage, and then comparing our method's estimated original surface with the actual original surface.

Compared to 3D anomaly detection, our approach avoids generalization issues by using traditional surface fitting and segmentation while enabling precise defect quantification for better maintenance and tracking. Unlike visual detection, our method differentiates defects from discoloration and detects subtle anomalies. However, we demonstrated that combining 3D and visual information outperforms using either modality alone, with even simple image-based methods proving effective in this fusion.

One limitation of our method is its difficulty in handling very shallow defects, or defects spanning a large surface area remains a challenge, as they make it difficult to distinguish between the natural curvature of the surface and the curvature induced by the damage. This limitation may be overcome when additional prior knowledge is available, such as geometric constraints on the shape of the original surface.

Finally, the performance of defect segmentation is influenced by the quality of the reconstructed 3D model and the validity of the smoothness assumption. Artifacts or noise in the reconstruction process can lead to false detections or missed defects, while inappropriate smoothness assumptions may introduce errors in residual computation. Simple rule-based outlier filtering, such as defect size constraints, can help mitigate false positives but requires some domain knowledge.

VI. CONCLUSION AND FUTURE WORK

We introduced a method for effectively identifying, segmenting, and quantifying surface defects in 3D reconstructions, achieving robust performance across various surface types and defect profiles.

Applied to real-world data, our approach successfully segments defects and produced detailed residual maps, particularly on a wind turbine blade segment and a gear. By integrating 3D data with image-based techniques, we enhance defect detection and quantification, enabling better maintenance decisions and a greater understanding of defect progression, crucial for predictive modeling.

A current limitation is the inability to handle objects with complex shapes. Future work could integrate spatially adaptive smoothing splines to adjust smoothing parameters dynamically, improving surface modeling accuracy. A key challenge is preventing over-adjustment in damaged areas.

Another avenue is tracking defect progression across inspections by aligning 3D reconstructions, identifying corresponding defects, and handling cases where segmented defects merge over time.

Furthermore, conducting a sensitivity analysis on inspection conditions – such as the number of images, image blur, and other factors affecting 3D model quality – could provide valuable insights into the robustness of our approach in real-world settings. Additionally, integrating 3D residual maps into defect classification through early fusion with RGB images could significantly improve performance and open up new research opportunities.

REFERENCES

[1] M. R. Jahanshahi, J. S. Kelly, S. F. Masri, and G. S. Sukhatme, "A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures," *Struct. Infrastruct. Eng.*, vol. 5, no. 6, pp. 455–486, Dec. 2009.

[2] M. Fox, S. Goodhew, and P. De Wilde, "Building defect detection: External versus internal thermography," *Building Environ.*, vol. 105, pp. 317–331, Aug. 2016.

[3] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Informat.*, vol. 29, no. 2, pp. 196–210, Apr. 2015.

[4] Z. Geng, C. Shi, and Y. Han, "Intelligent small sample defect detection of water walls in power plants using novel deep learning integrating deep convolutional GAN," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7489–7497, Jun. 2023.

[5] Y. Du, S. Zhou, X. Jing, Y. Peng, H. Wu, and N. Kwok, "Damage detection techniques for wind turbine blades: A review," *Mech. Syst. Signal Process.*, vol. 141, Jul. 2020, Art. no. 106445.

[6] M. Sánchez-Silva, D. M. Frangopol, J. Padgett, and M. Soliman, "Maintenance and operation of infrastructure systems," *J. Struct. Eng.*, vol. 142, no. 9, 2016, Art. no. 4016004.

[7] E. I. Basri, I. H. A. Razak, H. Ab-Samat, and S. Kamaruddin, "Preventive maintenance (PM) planning: A review," *J. Quality Maintenance Eng.*, vol. 23, no. 2, pp. 114–143, May 2017.

[8] Y. Cha, R. Ali, J. S. Lewis, and O. Büyüköztürk, "Deep learning-based structural health monitoring," *Autom. Construction*, vol. 161, Jan. 2024, Art. no. 105328.

[9] S. Jordan et al., "State-of-the-art technologies for UAV inspections," *IET Radar, Sonar Navigat.*, vol. 12, no. 2, pp. 151–164, Feb. 2018.

[10] T. Schenk, "Introduction to photogrammetry," *Ohio State Univ., Columbus*, vol. 106, no. 1, p. 1, 2005.

[11] P. Bergmann and D. Sattlegger, "Anomaly detection in 3D point clouds using deep geometric descriptors," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 2612–2622.

[12] J. Liu et al., "Real3D-AD: A dataset of point cloud anomaly detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 30402–30415.

[13] Y. Cao, X. Xu, and W. Shen, "Complementary pseudo multimodal feature for point cloud anomaly detection," *Pattern Recognit.*, vol. 156, Dec. 2024, Art. no. 110761.

[14] Geometrical Product Specifications (GPS) Filtration Part 32: Robust Profile Filters: Spline Filters, Standard ISO 16610-32, International Organization for Standardization, Geneva, Switzerland, Oct. 2023.

[15] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *J. Comput. Civil Eng.*, vol. 17, no. 4, pp. 255–263, Oct. 2003.

[16] Y.-J. Cha, K. You, and W. Choi, "Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Autom. Construction*, vol. 71, pp. 181–188, Nov. 2016.

[17] S. Patsias and W. J. Staszewskiy, "Damage detection using optical measurements and wavelets," *Structural Health Monitor.*, vol. 1, no. 1, pp. 5–22, Jul. 2002.

[18] Y. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, 2017.

[19] S. Qi, J. Yang, and Z. Zhong, "A review on industrial surface defect detection based on deep learning technology," in *Proc. 3rd Int. Conf. Mach. Learn. Mach. Intell.*, Sep. 2020, pp. 24–30.

[20] E. Horwitz and Y. Hoshen, "Back to the feature: Classical 3D features are (Almost) all you need for 3D anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 2968–2977.

[21] G. H. Beckman, D. Polyzois, and Y.-J. Cha, "Deep learning-based automatic volumetric damage quantification using depth camera," *Autom. Construction*, vol. 99, pp. 114–124, Mar. 2019.

[22] M. G. Guerra and R. A. Galantucci, "Standard quantification and measurement of damages through features characterization of surface imperfections on 3D models: An application on architectural heritages," *Proc. CIRP*, vol. 88, pp. 515–520, Jan. 2020.

[23] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vis. Comput.*, vol. 23, no. 3, pp. 299–309, Mar. 2005.

[24] J. Zhang, Y. Yao, and B. Deng, "Fast and robust iterative closest point," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3450–3466, Jul. 2022.

[25] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[26] F. Pérez-Arribas and I. Trejo-Vargas, "Computer-aided design of horizontal axis turbine blades," *Renew. Energy*, vol. 44, pp. 252–260, Aug. 2012.

[27] F. Diara and F. Rinaudo, "Building archaeology documentation and analysis through open source HBIM solutions via NURBS modelling," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 1381–1388, Aug. 2020.

[28] X. Zhang et al., "NURBS modeling and isogeometric shell analysis for complex tubular engineering structures," *Comput. Appl. Math.*, vol. 36, no. 4, pp. 1659–1679, Dec. 2017.

[29] G. Micula and S. Micula, *Handbook of Splines*, vol. 462. Cham, Switzerland: Springer, 2012.

[30] P. Dierckx, *Curve and Surface Fitting With Splines*. London, U.K.: Oxford Univ. Press, 1995.

[31] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a 'Kneedle' in a haystack: Detecting knee points in system behavior," in *Proc. 31st Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2011, pp. 166–171.

[32] D. L. Whaley III, "The interquartile range: theory and estimation," M.S. thesis, Dept. Math., East Tennessee State Univ., Johnson City, TN, USA, 2005.

[33] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, no. 1, pp. 102–114, Jan. 2017.

[34] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, Jun. 1994.

[35] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[36] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, "Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling," *IEEE Trans. Image Process.*, vol. 29, pp. 1999–2012, 2020.

[37] M. Pollefeys, R. Koch, and L. V. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," *Int. J. Comput. Vis.*, vol. 32, no. 1, pp. 7–25, 1999.

[38] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern Recognit. Lett.*, vol. 27, no. 14, pp. 1644–1649, Oct. 2006.

[39] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, Mar. 2020.

[40] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*, vol. 84. Cham, Switzerland: Springer, 2013.

[41] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Int. J. Comput. Vis.*, vol. 19, no. 1, pp. 57–91, Jul. 1996.

[42] J. Sterckx, M. Vlaminck, K. De Bauw, and H. Luong, "Accurate and robust 3D reconstruction of wind turbine blade leading edges from high-resolution images," *Autom. Construct.*, vol. 175, Jul. 2025, Art. no. 106153.

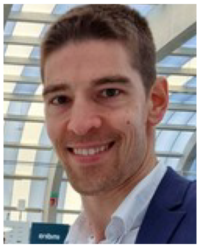
[43] J. Edstedt, Q. Sun, G. Bökman, M. Wadenbäck, and M. Felsberg, "RoMa: Robust dense feature matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 19790–19800.

[44] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4104–4113.

[45] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian splatting for geometrically accurate radiance fields," in *Proc. Special Interest Group Comput. Graph. Interact. Techn. Conf. Conf. Papers*, Jul. 2024, pp. 1–11.

[46] J. C. Poletto et al., "Identification of gear wear damage using topography analysis," *Wear*, vol. 522, Jun. 2023, Art. no. 204837.

Jonathan Sterckx received the master's degree in electrical engineering from Ghent University, Belgium, in 2023, where he is currently pursuing the Ph.D. degree in computer science engineering with the Image Processing and Interpretation Research Group and the UAV Research Center. His research interests include 3D reconstruction and surface analysis techniques for automated inspection processes.



Michiel Vlaminck received the Ph.D. degree in computer science engineering from Ghent University, Belgium, in 2020. Since then, he has been a Post-Doctoral Researcher at the UAV Research Center, Ghent University, focusing on 3D scene reconstruction using active depth sensors. His research interests include applications in augmented reality, autonomous robotics, and UAV.

Hiep Luong received the Ph.D. degree in computer science engineering from Ghent University, Belgium, in 2009. He has been a Post-Doctoral Researcher and the Project Manager of the Image Processing and Interpretation Group (IPI), Ghent University. He is currently a Professor at IPI and leads the UAV Research Center, Ghent University. His research interests include image and real-time video processing across fields, such as high dynamic range imaging, multi-view processing, and multi-sensor fusion for UAV and augmented reality applications.