Det-Recon-Reg: An Intelligent Framework Toward Automated UAV-Based Large-Scale Infrastructure Inspection

Guidong Yang, *Graduate Student Member, IEEE*, Benyun Zhao, Jihan Zhang, *Member, IEEE*, Junjie Wen, *Graduate Student Member, IEEE*, Qingxiang Li, Lei Lei, *Member, IEEE*, Xi Chen, and Ben M. Chen, *Fellow, IEEE*

***Abstract*—Visual inspection remains essential for inspecting infrastructure surfaces. While there are cornerstones in developing intelligent inspection systems, most existing solutions are limited to small-scale infrastructures and components, making them challenging to scale up for real-world applications. Leveraging deep learning and unmanned aerial vehicles (UAVs), this article proposes *Det-Recon-Reg*, an intelligent framework born for large-scale infrastructure inspection by decomposing it into three complementary stages: *detect* for defect detection, *reconstruct* for infrastructure reconstruction, and *register* for defect registration. In the *detect* stage, we introduce the first high-resolution dataset designed for defect detection on large-scale infrastructure surfaces. State-of-the-art real-time object detectors are evaluated on this dataset, and the CUBIT-Net is proposed to strike a better balance between accuracy and efficiency. In the *reconstruct* stage, we present a scalable multi-view stereo (MVS) network to reconstruct dense point cloud representation of the infrastructure from multi-view images. Extensive experiments on benchmark datasets, including DTU, Tanks and Temples (TNT), and BlendedMVS, demonstrate the superior performance of our method over existing approaches. In the *register* stage, we propose a novel defect registration method that leverages the geographic information system (GIS) to accurately map the detected defects onto the infrastructure model while preserving their geometric and visual properties, thereby enabling global defect localization and more informed maintenance decision-making. The proposed framework can serve as a reference for effective and efficient infrastructure maintenance as consolidated in real-world experiments. Codes, datasets, and pretrained models for each stage will be released at https://github.com/YANG-SOBER/Det-Recon-Reg. The supplementary video is available at: https://youtu.be/MVMp7k9qB84**

***Index Terms*—Infrastructure inspection, intelligent framework, multi-view stereo (MVS), unmanned aerial vehicle (UAV).**

Received 18 October 2024; revised 6 April 2025; accepted 2 May 2025. Date of publication 19 May 2025; date of current version 3 June 2025. This work was supported in part by the Research Grants Council of Hong Kong SAR under Grant 14206821, Grant 14217922, and Grant 14209623; and in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics. The Associate Editor coordinating the review process was Dr. Kunpeng Zhu. *(Corresponding author: Benyun Zhao.)*

The authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong (CUHK), Shatin, NT, Hong Kong (e-mail: gdyang@mae.cuhk.edu.hk; byzhao@mae.cuhk.edu.hk; jhzhang@mae.cuhk.edu.hk; jjwen@mae.cuhk.edu.hk; qingxiang.li@polimi.it; llei43-c@my.cityu.edu.hk; xichen@mae.cuhk.edu.hk; bmchen@mae.cuhk.edu.hk).

Digital Object Identifier 10.1109/TIM.2025.3571118

I. Introduction

CIVIL infrastructure is highly susceptible to performance degradation due to factors such as structural aging, construction deficiencies, design flaws, and environmental impacts [1], posing significant risks to its functional safety, operational efficiency, and long-term cost-effectiveness. Therefore, periodic defect diagnosis and monitoring are critical to preserving structural integrity, ensuring functional safety, and optimizing energy efficiency in infrastructure systems. Among nondestructive testing methods, visual inspection has been the primary method for identifying critical surface defects, such as cracks, spalling, and moisture infiltration. However, traditional manual visual inspection is inherently subjective and prone to errors, while being labor-intensive and time-consuming, often leading to outdated inspection results by the time maintenance is performed. Recent advancements in the integration of unmanned robotic platforms [2], [3], [4], [5], [6] with state-of-the-art learning-based visual inspection techniques [7], [8], [9], [10] have emerged as a promising alternative to traditional manual inspection methods. This convergence has led to the automation of infrastructure inspection, facilitating the development of intelligent inspection systems capable of autonomously detecting and localizing surface defects, thereby providing a reliable reference that supports timely and informed maintenance decisions. For instance, a wall-climbing robot-based inspection system [11] and an unmanned aerial vehicle (UAV)-based inspection framework [12] have been developed to segment surface defects and project them onto 3-D infrastructure models for accurate spatial registration. The wall-climbing system utilizes a truncated signed distance function map combined with Delaunay graph-based mapping for surface reconstruction, while the UAV-based framework employs a patch-based multi-view stereo (MVS) method to reconstruct the point cloud from multiple viewpoints. Similarly, existing inspection frameworks integrate detected surface defects directly into point cloud models [13] generated by PatchMatch MVS or onto surface models [14] reconstructed by OpenMVS, highlighting their increasing potential to improve the accuracy, efficiency, and scalability of infrastructure inspection.

Despite remarkable achievements in existing inspection systems, they are limited to providing only the local positions of defects relative to the reconstructed infrastructure

model. Furthermore, these systems are constrained to small-scale environments, such as walls or bridge piers, due to the scalability limitations of the reconstruction methods employed. Consequently, large-scale infrastructure inspection, particularly the accurate and efficient localization of detected defects in global coordinates, remains a complex and unresolved challenge. To address this issue, we propose the Det-Recon-Reg framework, which automates large-scale infrastructure inspection by leveraging UAVs and learning-based techniques. The framework decomposes the inspection process into three complementary stages: 1) detect for defect detection; 2) reconstruct for infrastructure reconstruction; and 3) register for defect registration.

A. Detect

Unlike existing defect inspection frameworks [11], [12], [13], [14] that adopt defect segmentation for defect inspection, we focus on defect detection to identify and enclose surface defects within bounding boxes, allowing for the efficient determination of the defect center relative to the image center. Such relative positioning is crucial for accurately localizing the defect in global coordinates, facilitating subsequent defect registration, and improving defect management efficiency. However, the scarcity of the publicly available defect dataset has significantly hindered the progress of learning-based defect detection methods [1]. To address this issue, we propose CUBIT-Det,[1] a high-resolution defect dataset designed to facilitate defect detection on large-scale infrastructure surfaces. The dataset contains high-resolution defect images that provide rich semantic context for detecting defects in large-scale scenes, covering three common infrastructure types: buildings, pavements, and bridges, and focusing on three critical defect types: cracks, spalling, and moisture. With this dataset, we benchmark state-of-the-art learning-based defect detection algorithms and propose the CUBIT-Net[1] to strike a better balance between detection accuracy and efficiency.

B. Reconstruct

Existing defect inspection systems [12], [13], [14], [15], [16] utilize traditional MVS methods to reconstruct dense point cloud models of infrastructure from multi-view images, providing a physical foundation for accurate defect registration. Nevertheless, these geometry-based approaches rely on hand-crafted similarity metrics for correspondence matching, often leading to incomplete reconstructions, especially on low-textured, reflective surfaces or those with repetitive patterns under varying illumination. In contrast, learning-based MVS methods [17], [18], [19], [20], [21], [22] implicitly encode camera parameters and global semantic information such as reflective and illumination priors into the network, enabling robust correspondence matching and significantly enhancing reconstruction accuracy, completeness, and efficiency. Moreover, learning-based MVS methods offer superior scalability, allowing inspection frameworks to generalize effectively across multiscale scenes. Despite these advances, their potential for infrastructure inspection remains underexplored. To bridge this gap, we present a scalable MVS network for more accurate, complete, and efficient infrastructure reconstruction. Extensive experiments on benchmark datasets, including DTU [23], TNT [24], and BlendedMVS [25], demonstrate the superior performance of our method over existing state-of-the-art approaches. Real-world deployment further highlights its efficacy and scalability across multiscale scenes.

C. Register

Accurate defect registration in global coordinates is crucial for reliable maintenance decision-making by aligning detected defects with the infrastructure model. Current defect inspection systems [11], [12], [13], [14] typically employ a segment-then-project strategy, projecting segmented defects from image coordinates onto the 3-D infrastructure model. While effective in small-scale scenes, this approach often suffers from reduced accuracy due to incomplete defect segmentation and suboptimal infrastructure reconstruction. Moreover, backprojected defects frequently exhibit geometric distortions due to perspective effects and occlusion, further compromising the reliability of maintenance decisions. Furthermore, this method fails to incorporate geographic information, resulting in local defect positioning relative to the infrastructure model, thereby limiting global defect localization. To address these limitations, we propose a novel method that leverages the geographic information system (GIS) to accurately register defects onto the infrastructure model while preserving their geometric and visual properties, thereby enabling global defect localization and more reliable maintenance planning. Real-world experiments demonstrate the centimeter-level registration accuracy of our method, underscoring its practical applicability for infrastructure maintenance.

To further automate the proposed Det-Recon-Reg framework for large-scale infrastructure inspection, we integrate multi-UAV cooperative coverage path planning [26], [27], [28] for the collaborative capture of close-range facade and multi-view aerial images, enabling surface defect detection and dense infrastructure reconstruction, respectively. All collected images are GPS-tagged and augmented with real-time kinematic data to ensure accurate defect registration. This approach follows an explore-then-exploit framework [28], where UAVs first explore the target infrastructure and its surroundings, dynamically updating a density map to guide the selection of optimal viewpoints. After exploration, UAVs follow optimized trajectories, derived from the traveling salesman problem (TSP) [26], to efficiently capture the required images. This multi-UAV strategy accelerates data acquisition and significantly improves inspection efficiency.

The contributions of this article are as follows.

1) An intelligent framework, Det-Recon-Reg, is proposed to decompose large-scale infrastructure inspection into three complementary stages: defect detection, infrastructure reconstruction, and defect registration.
2) A high-resolution defect dataset, CUBIT-Det, is constructed to address data scarcity, and a defect detection network, CUBIT-Net, is developed to balance accuracy and efficiency for large-scale scenes.

[1]CUBIT stands for CUHK Building Information Technology.

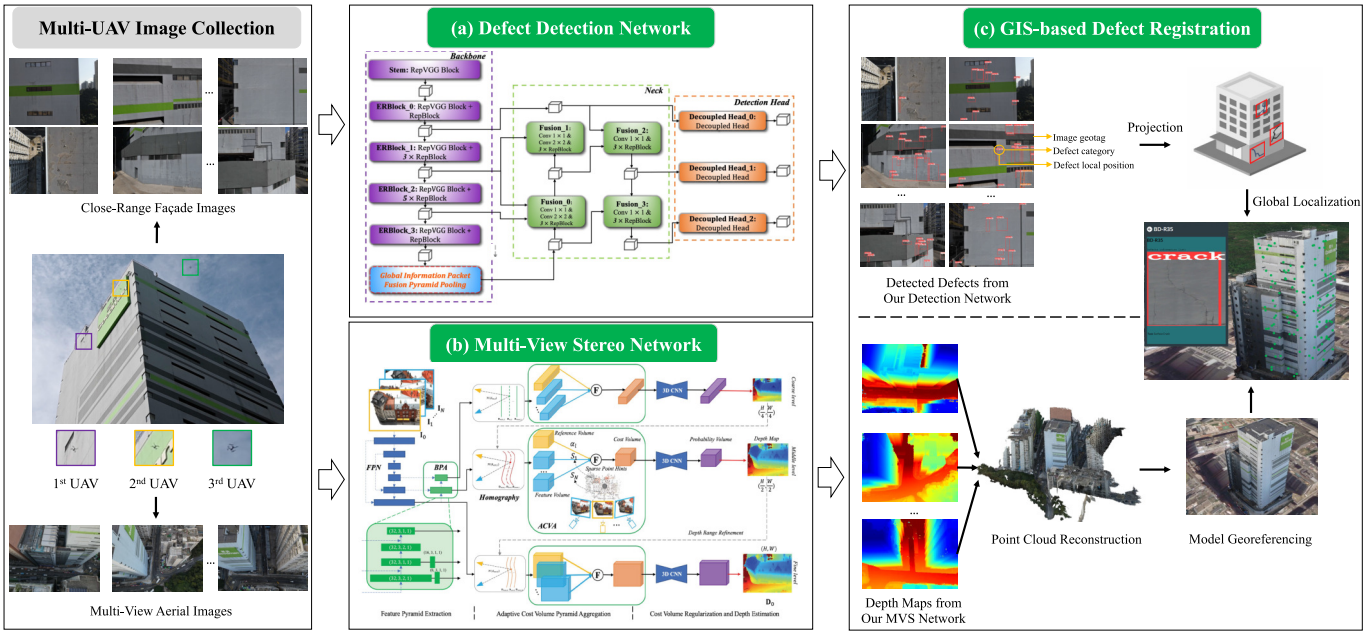


Fig. 1. System architecture of the proposed Det-Recon-Reg framework, enabling large-scale infrastructure inspection through three stages. (a) Detect. (b) Reconstruct, and (c) Register. In real-world application, the framework takes as input close-range facade images (top row) and multi-view aerial images (bottom row) with GPS tags, autonomously collected via multi-UAV cooperative coverage path planning. Surface defects are then identified by CUBIT-Net leveraging the domain-specific CUBIT-Det dataset for training to enable defect detection, while dense infrastructure reconstruction is performed using the proposed MVS network. Detected defects are accurately georeferenced and registered within a GIS framework. Refer to the supplementary video for clarity.

3) A scalable, learning-based MVS method is proposed to improve the accuracy, completeness, and efficiency of large-scale infrastructure reconstruction.
4) A GIS-based method is proposed to register defects onto the infrastructure model, preserving their geometric and visual properties.
5) Extensive experiments on benchmark datasets and real-world scenes validate the effectiveness, efficiency, and scalability of the framework for large-scale infrastructure inspection.

The remainder of this article is organized as follows. Section II details the methodology of the proposed inspection framework. Section III presents benchmark experiments and ablation studies to evaluate the effectiveness of each framework component. Section IV demonstrates the effectiveness, efficiency, and scalability of the framework in large-scale real-world scenes. Section V discusses the limitations. Section VI concludes the article and outlines future research directions.

II. Methodology

Framework Overview: Fig. 1 illustrates the system architecture of the proposed Det-Recon-Reg framework, consisting of three complementary stages: 1) detect for defect detection; 2) reconstruct for infrastructure reconstruction; and 3) register for defect registration. In real-world inspection deployments, the framework takes as input close-range facade and multi-view aerial images with GPS tags, collected through multi-UAV cooperative coverage path planning [26], [27], [28], to enable surface defect detection and dense reconstruction of large-scale infrastructure. The defects are then accurately mapped onto the infrastructure model using the proposed GIS-based registration method, providing a reliable global reference for informed maintenance decision-making.

A. Detect

1) Defect Detection Dataset: We leverage defect detection to identify and enclose surface defects within bounding boxes, determining the defect center position relative to the image center. This positioning is crucial for subsequent GIS-based defect registration, ensuring accurate localization of defects in global coordinates. However, the limited availability of publicly available defect datasets has significantly hindered the advancement of learning-based defect detection methods [1]. To address this challenge, we propose CUBIT-Det, a high-resolution defect dataset designed to facilitate defect detection on large-scale infrastructure surfaces. The dataset contains 5527 high-resolution images with resolutions up to $H \times W = 6000 \times 8000$, covering three common infrastructure types: buildings, pavements, and bridges, and focusing on three key defect types: cracks, spalling, and moisture. As shown in Fig. 2, these images are captured from multiple viewpoints using onboard cameras mounted on unmanned vehicles, with variations in shooting angle, surface texture, depth range, and illumination conditions, thereby providing rich semantic context and improving model robustness for real-world large-scale infrastructure inspection. The dataset properties are summarized as follows.

1) *Infrastructure Type:* Fig. 3(a) illustrates the distribution of infrastructure types in the CUBIT-Det dataset, which includes three prevalent categories: buildings, pavements, and bridges, accounting for 65%, 29%, and 6%, respectively. Notably, due to the challenging data

Fig. 2. Images from our dataset. Cracks on building façade (first row), while cracks on road surfaces and bridges (second row). Spalling and moisture damage (third and fourth rows), respectively.

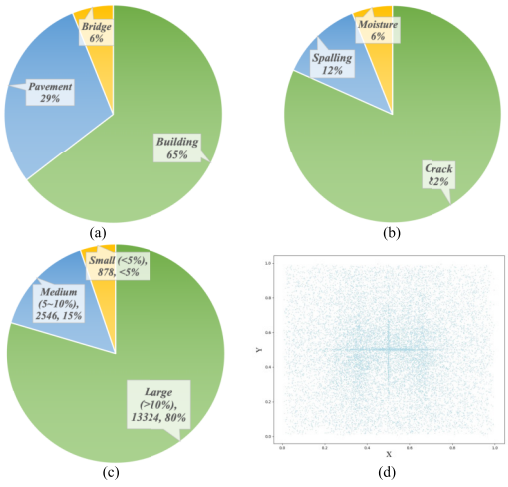


Fig. 3. (a) Infrastructure type. (b) Defect category. (c) Defect dimension. (d) Defect spatial distribution: each scatter represents the spatial position of one target defect relative to the bottom-left corner of the image.

TABLE I
COMPARISON WITH EXISTING DEFECT DETECTION DATASETS

Dataset	Number of Images	Resolution	Data Collection Platform	Defect Type	Structure	Material	Experiments
RDD-2018 [29]	9053	600 × 600	Cameras on ground vehicle	Crack Corrosion	Pavement	Asphalt	- SSD (InceptionV2, MobileNet)
RDD-2019 [30]	13135	600 × 600	Cameras on ground vehicle	Crack Corrosion	Pavement	Asphalt	- SSD (ResNet50, (MobileNet)
RDD-2020 [31]	26336	600 × 600 720 × 720	Cameras on ground vehicle	Crack Pothole	Pavement	Asphalt	- SSD (MobileNet)
RDD-2022 [32]	47420	512 × 512 600 × 600 720 × 720 3650 × 2044	Smartphones Hand-held cameras UAV cameras Google street view	Crack Pothole	Pavement	Asphalt	-
PID [33]	7237	640 × 640	Crawled from Internet	Crack	Pavement	Asphalt	a. YOLOv2 b. Fast R-CNN
Murad [34]	2620	up to 838 × 809	Hand-held phones and UAV	Crack	Pavement	Asphalt	- Faster R-CNN
SUT-Crack [35]	130	4032 × 3024	Cameras on ground vehicle	Crack	Pavement	Asphalt	-
CODEBRIM [36]	1590	up to 6000 × 4000	Hand-held cameras Cameras on UAV	Crack Corrosion	Bridge	Concrete	a. MetaQNN b. Efficient Neural Architecture Search
CUBIT-Det	**5527**	**4624 × 3472 and 8000 × 6000**	**Cameras on Unmanned Vehicles**	**Crack Spallinig Moisture**	**Building (65%) Pavement (29%) Bridge (6%)**	**Concrete Asphalt Stone**	**a. Faster R-CNN [37] (ResNet) b. PP-YOLO [38](ResNet) c. PP-YOLOv2 [39](ResNet) b. PP-YOLOE(s,m) [40] c. PP-YOLOE+(s,m) [40] d. YOLOX(n,t,s,m) [41] e. YOLOv5(n,s,m) [42] f. YOLOv7(t,normal) [43] g. YOLOv6(n,s,m) [44] *i. Real-world experiment with CUBIT-Net***

collection process, no existing defect detection datasets, as shown in Table I, provide annotated defect images specifically for buildings. In contrast, we adopt multi-UAV cooperative coverage path planning [26], [27], [28] to autonomously capture close-range images of building facades, even in GPS-denied environments.

2) *Defect Category:* As shown in Fig. 3(b), the CUBIT-Det dataset covers three primary categories of surface defects: cracks, spalling, and moisture, comprising 82%, 12%, and 6%, respectively. These defect categories were selected following a thorough review of established infrastructure inspection guidelines and standards, including reports from the Hong Kong Housing Authority, guidelines from the Hong Kong Institute of Surveyors, and the BSI Standard Publication on Service Life Planning for Buildings and Constructed Assets.

3) *Defect Dimension:* As depicted in Fig. 3(c), defects in the CUBIT-Det dataset are categorized into large, medium, and small defects, comprising 80%, 15%, and 5% of the total defects, respectively. Thanks to the high-resolution nature of the dataset, even medium-sized defects achieve a resolution of $H \times W = 600 \times 800$, which exceeds the resolution of entire images in most existing defect detection datasets, as shown in Table I. This indicates that our dataset is rich in spatial and semantic context information, which improves model robustness and generalization ability in real-world defect inspection.

4) *Defect Spatial Distribution:* Fig. 3(d) shows the spatial distribution of defects in our dataset. Approximately 20% of defects are concentrated within $[-2.5\%, 2.5\%]$ along the central axes ($x = 0.5$ and $y = 0.5$), forming a cross-shaped pattern. This pattern arises from positioning defects near the image center during close-range imaging to ensure detailed representation. While exhibiting a cross-shaped pattern, defects are distributed across the entire image, covering both central and peripheral regions. This spatial distribution allows models to learn features invariant to spatial variations, which is crucial for real-world applications where defects can appear anywhere in the visual field. By ensuring extensive spatial coverage, our dataset enhances model generalization and spatial awareness, particularly for defect detection in high-resolution images.

5) *Comparison With Existing Defect Datasets:* To highlight the differences between our dataset and existing defect detection datasets, we summarize key properties and present the comparisons in Table I. While CUBIT-Det may not surpass existing datasets in terms of data volume, it offers several distinguishing features, including high-resolution images, diverse infrastructure categories, and a wide variety of defect types, providing rich spatial and semantic context. We validate the effectiveness of CUBIT-Det by evaluating it with approximately 20 distinct models and real-world inspections, ensuring its robustness and distinguishing it from existing datasets.

2) Defect Detection Method: Following the construction of the dataset, approximately 20 state-of-the-art real-time detectors [37], [38], [39], [40], [41], [42], [43], [44] are trained and benchmarked to evaluate the effectiveness of the proposed dataset and identify the model that optimally

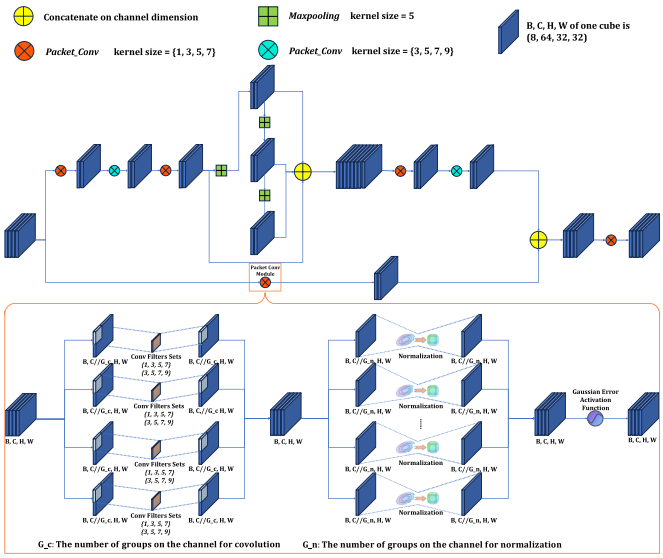


Fig. 4. Detailed architecture of the proposed GIPFPP module, with the structure of the Packet_Conv layer highlighted within the orange box.

balances detection accuracy and efficiency, with YOLOv6-n [44] selected as the baseline due to its superior tradeoff between these metrics. To further enhance this balance, we propose the global information packet fusion pyramid pooling (GIPFPP) module, which replaces the original feature fusion layer to accelerate feature processing and improve multiscale fusion. This enhanced method is referred to as CUBIT-Net, as illustrated in Fig. 1(a). The architecture of the GIPFPP module demonstrated in Fig. 4, divides the input feature map along the channel dimension into two branches: the top branch, which consists of three Packet_Conv layers with varying kernel sizes, followed by three max-pooling layers and two additional Packet_Conv layers, and the bottom branch, which includes a single Packet_Conv layer. The feature maps from both branches are concatenated along the channel dimension and processed by a final Packet_Conv layer to generate the output.

The Packet_Conv layer, depicted in the lower part of Fig. 4, consists of three stages: convolution, normalization, and activation. In the convolution phase, group convolution divides the input feature map into four groups along the channel dimension, with kernel sizes of $2i+1$ $(3,5,7,9)$ for the indigo circle and $2i-1$ $(1,3,5,7)$ for the carrot-orange circle. Depthwise convolution is applied within each group, and the resulting feature maps are concatenated back to their original dimensions. Larger kernels capture global features from a wide receptive field, while smaller kernels focus on local features, providing complementary benefits. Despite the increased parameters of larger kernels, the 'packaging and depthwise convolution' strategy effectively reduces both parameter count and inference latency. Moreover, we employ group normalization in place of batch normalization within Packet_Conv. This approach re-groups feature maps across channels at the same spatial positions, alleviating sensitivity to batch size variations. For nonlinearity, we adopt the Gaussian error linear unit (GELU) activation function, which provides smoother gradients and mitigates vanishing gradient issues, in contrast to the commonly used ReLU. Benchmark results and ablation studies, presented in Section III-A, demonstrate the effectiveness and efficiency of CUBIT-Net. The proposed CUBIT-Net, trained on the CUBIT-Det defect detection dataset, effectively inspects surface defects on the warehouse facade, as shown in Fig. 1(c).

B. Reconstruct

1) Method Overview: Learning-based MVS methods [17], [18], [19], [20], [21], [22] incorporate camera parameters and global semantic information such as reflective and illumination priors to achieve robust multi-view correspondence matching, delivering significant improvements in point cloud reconstruction accuracy, completeness, and efficiency compared to traditional approaches while offering superior scalability for generalizing across multiscale scenes. Despite their advancements, the application of learning-based MVS to infrastructure inspection remains underexplored. To address this limitation, we propose a scalable learning-based MVS method designed for more accurate, complete, and efficient reconstruction of large-scale infrastructure. Our method consists of two sequential stages: 1) multi-view depth map estimation based on the proposed MVS network and 2) multi-view depth map fusion for dense point cloud reconstruction. The network enables high-resolution depth estimation while maintaining computational and memory efficiency in a coarse-to-fine manner. The network takes $(N+1)$ images as input, including a reference-view image $\mathbf{I}_0$ and N source-view images $\{\mathbf{I}_i\}_{i=1}^{N}$, to estimate the depth map $\mathbf{D}_{\text{est},0}$ for $\mathbf{I}_0$. For scene reconstruction, each image is iteratively treated as $\mathbf{I}_0$ to estimate the corresponding depth map. The resulting multi-view depth estimates are then refined with probabilistic and geometric constraints before being fused to reconstruct the dense point cloud $\mathcal{R}$.

2) Feature Pyramid Extraction: The MVS network takes N-view images $\{\mathbf{I}_i\}_{i=0}^{N}$ with associated camera intrinsics $\{\mathbf{K}_i \in \mathbb{R}^{3\times3}\}_{i=0}^{N}$ and extrinsics $\{[\mathbf{R}_i \in \mathbb{R}^{3\times3}; \mathbf{t}_i \in \mathbb{R}^{3\times1}]\}_{i=0}^{N}$ as input, processed by a feature pyramid network (FPN) [17], [18], [19], [20], [21] to extract feature pyramid $\{\mathbf{f}_{l,i} \in \mathbb{R}^{F_l\times(H/2^l)\times(W/2^l)}\}_{i=0}^{N}$ at $(L+1)$ scales, where $l \in \{0,1,\ldots,L\}$ represents feature level, F_l indicates channel number at feature level l, and H and W denote image height and width, respectively. The FPN parameters are shared across all views to improve learning efficiency. However, despite the effectiveness of the FPN in extracting multiscale features, depth estimation near the boundaries of reflective and textureless surfaces remains challenging. This is primarily due to the absence of low-level spatial features, such as edges and textures, which are essential for accurate depth prediction. As a result, over-smoothing often occurs in these regions, leading to suboptimal depth estimation. To overcome this limitation, we propose a bottom-up pathway (BPA) following the FPN, as shown in Fig. 1(b). This pathway is designed to enhance the transition of spatial features between the feature pyramid extraction module and the ACVA module, improving the robustness of multi-view correspondence matching. Our ablation study demonstrates that the BPA effectively improves both the depth estimation and the reconstruction performance.

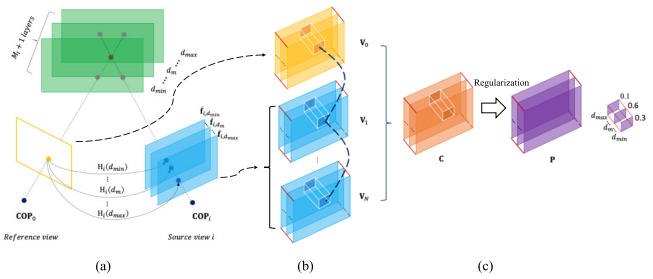


Fig. 5. Illustration of ACVA, including (a) homography warping, (b) cost volume aggregation, and (c) cost volume regularization with depth estimation. COP indicates the center of projection.

3) Adaptive Cost Volume Aggregation: Depth sampling is performed to uniformly discretize the reference-view depth range $[d_{\min,l}, d_{\max,l}]$ into $(M_l + 1)$ depth plane hypotheses, as illustrated by the green rectangles in Fig. 5(a). The depth hypotheses are given by

$$d_{m,l} = d_{\min,l} + m \cdot \frac{(d_{\max,l} - d_{\min,l})}{M_l} \tag{1}$$

where $m \in \{0, 1, \ldots, M_l\}$ and M_l represents the number of depth intervals between these depth plane hypotheses. While the depth range for the coarser level is predefined, the depth range at finer levels is adaptively adjusted based on depth estimates from the coarser levels, as explained in the following coarse-to-fine depth estimation paradigm.

The sampled depth hypothesis $d_{m,l}$ defines the homography between the pixel $\mathbf{p}_{l,0}$ in the reference-view feature map $\mathbf{f}_{l,0}$ and the pixel $\mathbf{p}_{l,i}$ in the ith source-view feature map $\mathbf{f}_{l,i}$, as expressed by the following equation:

$$\mathbf{p}_{l,i} = \mathbf{K}_{l,i}\left(\mathbf{R}_{0\to i}\left(\mathbf{K}_{l,0}^{-1}\mathbf{p}_{l,0}d_{m,l}\right) + \mathbf{t}_{0\to i}\right) \tag{2}$$

where the scaled camera intrinsic matrices at feature level l, denoted as $\mathbf{K}_{l,0}$ and $\mathbf{K}_{l,i}$, correspond to the reference view and ith source view, respectively. The relative rotation matrix $\mathbf{R}_{0\to i}$ and relative translation vector $\mathbf{t}_{0\to i}$ between the reference and ith source view are computed as $\mathbf{R}_i\mathbf{R}_0^{-1}$ and $\mathbf{t}_0 - \mathbf{R}_i\mathbf{R}_0^{-1}\mathbf{t}_i$, respectively. For each depth hypothesis $d_{m,l}$, feature correspondences between the reference and ith source view are established by performing differentiable bilinear interpolation to warp the ith source-view feature map $\widetilde{\mathbf{f}}_{l,i,d_{m,l}}$ [the blue rectangles in Fig. 5(a)] aligned with the reference-view feature map $\mathbf{f}_{l,0}$ [the yellow rectangle in Fig. 5(a)]. The reference-view feature map $\mathbf{f}_{l,0}$ is replicated $(M_l + 1)$ times along the depth dimension to obtain the reference-view feature volume $\mathbf{V}_{l,0} \in \mathbb{R}^{F_l\times(M_l+1)\times(H/2^l)\times(W/2^l)}$, represented by the yellow volume in Fig. 5(b). Meanwhile, the warped source-view feature maps $\widetilde{\mathbf{f}}_{l,i,d_{m,l}}$ at each depth hypothesis $d_{m,l}$ are aggregated along the depth dimension to form the source-view feature volumes $\{\mathbf{V}_{l,i} \in \mathbb{R}^{F_l\times(M_l+1)\times(H/2^l)\times(W/2^l)}\}_{i=1}^{N}$, shown as the blue volumes in Fig. 5(b).

The multi-view feature volumes $\{\mathbf{V}_{l,i}\}_{i=0}^{N}$ are aggregated into the cost volume $\mathbf{C}_l$ [the orange volume of Fig. 5(c)] to evaluate feature matching similarity across multiple views. Heuristic-based methods [45], [46], [47] assign equal significance to all multi-view feature volumes, leading to matching ambiguity.

Algorithm 1 Matching Score Computation

Input : $\{\mathbf{p}_{ij} \in \mathbb{R}^{3\times1}, j \in \{0, 1, \ldots, n_i - 1\}\}_{i=1}^{N}$; $\{\mathbf{R}_i\}_{i=0}^{N} \in \mathbb{R}^{3\times3}$, $\{\mathbf{t}_i\}_{i=0}^{N} \in \mathbb{R}^{3\times1}$.
Output : Matching score $\{S_i\}_{i=1}^{N}$ between $\mathbf{I}_0$ and $\{\mathbf{I}_i\}_{i=1}^{N}$.
Initialization: Favoring baseline angle $\theta_0 = 5°$; Standard deviation of the piecewise gaussian function $\sigma_1 = 1$ and $\sigma_2 = 10$; Matching score $S_i = 0$.
Compute reference-view camera center: $\mathbf{c}_0 = -\mathbf{R}_0^T\mathbf{t}_0$;
for $i = 1$ **to** N **do**
  Compute source-view camera center: $\mathbf{c}_i = -\mathbf{R}_i^T\mathbf{t}_i$;
  for $j = 0$ **to** $n_i - 1$ **do**
    Compute the baseline angle θ_j:
    $\theta_j = \frac{180°}{\pi}\arccos\left(\frac{(\mathbf{c}_0-\mathbf{p}_{ij})\cdot(\mathbf{c}_i-\mathbf{p}_{ij})}{\|\mathbf{c}_0-\mathbf{p}_{ij}\|_2\|\mathbf{c}_i-\mathbf{p}_{ij}\|_2}\right)$;
    if $\theta_j \le \theta_0$ **then**
      $S_i \mathrel{+}= \exp\left(-\frac{(\theta_j-\theta_0)^2}{2\sigma_1^2}\right)$
    else
      $S_i \mathrel{+}= \exp\left(-\frac{(\theta_j-\theta_0)^2}{2\sigma_2^2}\right)$
    end
  end
end
return S_i *for each* i

In contrast, learning-based methods [48], [49], [50], [51], [52] adaptively aggregate the cost volume by learning pixel-wise, patch-wise, or channel-wise significance, but incurring additional computational costs. Moreover, existing methods fail to account for pixel discrepancies arising from viewpoint variations, where a source-view image that is both closer to the reference view and free from occlusions tends to offer more accurate photometric and geometric information than a more distant image affected by partial occlusions [53]. To address these issues, we propose sparse ACVA, an ACVA method guided by sparse point reconstruction from structure-from-motion (SfM). The sparse ACVA is determined as follows:

$$\begin{aligned}\mathbf{C}_l &= \mathcal{M}(\mathbf{V}_{l,0}, \ldots, \mathbf{V}_{l,N}) \\ &= \mathcal{M}(\mathbf{B}_{l,0}, \ldots, \mathbf{B}_{l,N}) \\ &= \text{AvgPool}\left(\alpha_l\mathbf{B}_{l,0} \odot \sum_{i=1}^{N}\frac{S_i}{\sum_{i=1}^{N}S_i}\mathbf{B}_{l,i}\right)\end{aligned} \tag{3}$$

where the mapping function $\mathcal{M}$ transforms the multi-view feature volumes $\{\mathbf{V}_{l,i}\}_{i=0}^{N}$ into the cost volume $\mathbf{C}_l$. To reduce memory footprint and enhance computational efficiency, each feature volume is partitioned into K subfeature volumes $\{\mathbf{B}_{l,i} \in \mathbb{R}^{K\times(F_l/K)\times(M_l+1)\times(H/2^l)\times(W/2^l)}\}_{i=0}^{N}$. During the aggregation process, the significance of the source-view subfeature volumes $\{\mathbf{B}_{l,i}\}_{i=1}^{N}$ is computed as $\left\{S_i/\sum_{i=1}^{N}S_i\right\}_{i=1}^{N}$, where $\{S_i\}_{i=1}^{N}$ representing the scene-dependent matching scores between source-view images $\{\mathbf{I}_i\}_{i=1}^{N}$ and the reference-view image $\mathbf{I}_0$ are computed based on Algorithm 1. The set $\{\mathbf{p}_{ij} \in \mathbb{R}^{3\times1}, j \in \{0, 1, \ldots, n_i - 1\}\}_{i=1}^{N}$ represents the sparse points triangulated by $\{\mathbf{I}_i\}_{i=1}^{N}$ and $\mathbf{I}_0$, where n_i denotes the number of sparse points. The baseline angle corresponding to $\mathbf{p}_{ij}$ is

denoted by θ_j. To adapt to the scene variation, S_i is initially computed through a piecewise Gaussian function that prioritizes the favoring baseline angle θ_0 and then normalized as the aggregation significance for the source view. The reference-view aggregation significance is governed by the learnable parameter α_l. The symbol $\odot$ denotes the Hadamard product, which calculates the feature-matching similarity between the weighted reference-view subfeature volume $\mathbf{B}_{l,0}$ and the normalized source-view subfeature volumes $\{\mathbf{B}_{l,i}\}_{i=1}^{N}$. Finally, the feature similarities are aggregated using average pooling across the channel dimension, yielding the final cost volume $\mathbf{C}_l \in \mathbb{R}^{K\times(M_l+1)\times(H/2^l)\times(W/2^l)}$.

4) Cost Volume Regularization and Depth Estimation: In line with previous studies [17], [18], [19], [20], [21], a multiscale 3-D U-Net is adopted to regularize the noise-affected cost volume $\mathbf{C}_l$, generating the estimated probability volume $\mathbf{P}_{l,\text{est}} \in \mathbb{R}^{(M_l+1)\times(H/2^l)\times(W/2^l)}$ [the violet volume in Fig. 5(c)], corresponding to $(M_l + 1)$ depth hypotheses. Existing methods typically address the depth estimation through either depth regression [17] or classification [54]. In regression, depth is predicted as the probabilistic weighted sum of sampled depth hypotheses, but this approach is susceptible to learning ambiguity as multiple weight combinations can produce identical depth estimations. On the other hand, the classification approach directly assigns the depth estimate to the hypothesis with the highest probability, resulting in a discrete depth prediction. To overcome these limitations, we improve discrete depth estimation by incorporating a depth residual between the discrete and target depths, enabling continuous depth estimation and enhancing the accuracy and completeness of the subsequent point cloud reconstruction. For level l, the continuous depth estimation $\mathbf{D}_{\text{est},l}$ is defined as

$$\mathbf{D}_{\text{est},l} = \mathbf{D}_{l,\text{discrete}} + \mathbf{D}_{l,\text{residual}} \tag{4}$$

where $\mathbf{D}_{l,\text{discrete}}$ and $\mathbf{D}_{l,\text{residual}}$ represent the discrete depth estimation and depth residual at level l, respectively. The discrete depth estimation $\mathbf{D}_{l,\text{discrete}}$ is determined as

$$\mathbf{D}_{l,\text{discrete}} = \underset{d_{m,l}\in[d_{\min,l},d_{\max,l}]}{\arg\max} \mathbf{P}_{l,\text{est}}(d_{m,l}) \tag{5}$$

where the depth hypothesis corresponding to the maximum probability is selected as the discrete depth estimation. The depth residual $\mathbf{D}_{l,\text{residual}}$ is defined as

$$\mathbf{D}_{l,\text{residual}} = \frac{d_{\max,l} - d_{\min,l}}{M_l} \cdot \underbrace{\max \mathbf{P}_{l,\text{est}}(d_{m,l})}_{\text{normalized depth residual}} \tag{6}$$

where the depth residual is computed as the product of the depth interval and the normalized depth residual. The fine-level depth estimation $\mathbf{D}_{0,\text{est}}$ is utilized as the output for $\mathbf{I}_0$.

5) Loss Function: The normalized depth residual is obtained by adapting the generalized focal loss [20], [55] to minimize the discrepancy between the ground-truth probability volume $\mathbf{P}_{l,\text{gt}}$ and the estimated probability volume $\mathbf{P}_{l,\text{est}}$, where $\mathbf{P}_{l,\text{gt}}$ is the ground-truth normalized depth residual between the discrete depth hypothesis and the ground-truth depth. The network loss is defined as

$$\mathcal{L} = \sum_{l=0}^{L} \lambda_l \mathcal{L}_l \tag{7}$$

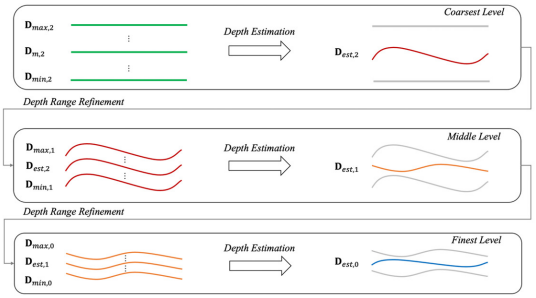


Fig. 6. Coarse-to-fine depth estimation paradigm, demonstrating the hierarchical depth estimation process across multiple resolution levels.

where $\mathcal{L}$ denotes the overall network loss for optimization and $\mathcal{L}_l$ and λ_l represent the loss and loss weight at feature level l, respectively. The per-level loss $\mathcal{L}_l$ is defined as

$$\begin{aligned}\mathcal{L}_l = \sum_{\mathbf{p}\in\{\mathbf{p}_{\text{valid}}\}} & -\beta_l \left|\mathbf{P}_{l,\text{gt}}(\mathbf{p}) - \mathbf{P}_{l,\text{est}}(\mathbf{p})\right|^{\gamma_l} \\ & \cdot \left[(1-\mathbf{P}_{l,gt}(\mathbf{p}))\log(1-\mathbf{P}_{l,\text{est}}(\mathbf{p})) \right. \\ & \left. + \mathbf{P}_{l,gt}(\mathbf{p})\log(\mathbf{P}_{l,\text{est}}(\mathbf{p}))\right]\end{aligned} \tag{8}$$

where $\{\mathbf{p}_{\text{valid}}\}$ represents valid pixel set, while γ_l and β_l denote the focusing and balancing factors, respectively.

6) Coarse-to-Fine Depth Estimation: As shown in Fig. 1(b), our MVS method is a three-stage network with three resolution levels including coarse level ($l = 2$), middle level ($l = 1$), and fine level ($l = 0$). We hence gradually conduct the depth estimation from the coarse level to the fine level to estimate the depth map $\mathbf{D}_0$ for the reference image $\mathbf{I}_0$ by following our previous depth estimation strategy.

For the coarse level $l = 2$, $(M_2 + 1)$ parallel depth planes (the green lines in Fig. 6) are uniformly sampled from the depth range $[d_{\min,2}, d_{\max,2}]$ measured at the reference view

$$\mathbf{D}_{\max,2} = d_{\max,2} \tag{9}$$

$$\mathbf{D}_{\min,2} = d_{\min,2} \tag{10}$$

$$\mathbf{D}_{m,2} = d_{\min,2} + m \cdot \frac{d_{\max,2} - d_{\min,2}}{M_2}, \quad m \in \{0, 1, \ldots, M_2\} \tag{11}$$

where $\mathbf{D}_{\max,2}$, $\mathbf{D}_{\min,2}$, and $\mathbf{D}_{m,2}$ represent the maximum, minimum, and sampled depth plane hypotheses, respectively. Notably, we perform depth hypothesis sampling to discretize the 3-D space into $(M_2 + 1)$ parallel depth planes along the depth direction. This does not imply that we assume the object in the reference image $\mathbf{I}_0$ is at the same depth, as each pixel in $\mathbf{I}_0$ has $(M_2 + 1)$ depth candidates. Specifically, based on the depth plane hypotheses, we adaptively construct the cost volume with sparse ACVA and perform continuous depth estimation to obtain the depth map $\mathbf{D}_{\text{est},2}$ (red curve in Fig. 6), where each pixel is assigned its optimal depth value.

For the middle level $l = 1$, we refine the depth range by utilizing the coarse-level depth estimation to derive refined depth hypotheses. Specifically, we center the depth range of the middle level at $\mathbf{D}_{\text{est},2}$ and concurrently reduce the depth interval I_1 and the number of depth hypotheses $(M_1 + 1)$ at

the middle level $l = 1$. The process can be formulated as follows:

$$\mathbf{D}_{\max,1} = \mathbf{D}_{\text{est},2} + \frac{(M_1+1)}{2} \cdot I_1 \tag{12}$$

$$\mathbf{D}_{\min,1} = \mathbf{D}_{\text{est},2} - \frac{(M_1+1)}{2} \cdot I_1 \tag{13}$$

$$I_1 = r_1 \cdot I_2, \quad r_1 < 1 \tag{14}$$

$$M_1 = \rho_1 \cdot M_2, \quad \rho_1 < 1 \tag{15}$$

$$\mathbf{D}_{m,1} = \mathbf{D}_{\min,1} + m \cdot \frac{\mathbf{D}_{\max,1} - \mathbf{D}_{\min,1}}{M_1}, \quad m \in \{0, 1, \ldots, M_1\} \tag{16}$$

where $\mathbf{D}_{\max,1}$, $\mathbf{D}_{\min,1}$, and $\mathbf{D}_{m,1}$ represent the maximum, minimum, and sampled depth hypotheses, respectively. The reduction factors r_1 and ρ_1 correspond to the depth interval and the number of depth hypotheses, respectively. The refined depth hypotheses $\mathbf{D}_{m,1}$ at level $l = 1$ are depicted using red curves. Next, we construct the cost volume with sparse ACVA and perform continuous depth estimation to obtain the depth map $\mathbf{D}_{\text{est},1}$ (orange curve). For the fine level $l = 0$, the same process is repeated to generate the final depth map estimation $\mathbf{D}_{\text{est},0}$ (blue curve) for the reference image $\mathbf{I}_0$.

7) Depth Map Fusion: Given the multi-view depth estimates, the probabilistic constraint τ and geometric constraint N_c are applied to reject depth outliers and promote multi-view depth consistency. The refined depth maps are subsequently backprojected and fused into the dense point cloud reconstruction $\mathcal{R}$. As depicted in Fig. 1(c), the proposed MVS method is deployed to real-world applications for dense infrastructure reconstruction, providing a reliable physical foundation for accurate defect registration in global coordinates.

C. Register

1) Method Overview: The key objective of large-scale infrastructure inspection is the accurate and efficient identification of global defect locations. The segment-then-project strategy of existing defect inspection systems [11], [12], [13], [14] only localizes the defects relative to the infrastructure model and often suffers from reduced accuracy due to incomplete defect segmentation and suboptimal infrastructure reconstruction. Moreover, backprojected defects frequently exhibit geometric distortions due to perspective effects and occlusion, further compromising the reliability of maintenance decisions. To overcome these challenges, we propose a novel approach that leverages the GIS for accurate defect registration. Our method aligns defects with the infrastructure model while preserving both their geometric and visual properties, enabling global defect localization and enhancing the reliability of maintenance decision-making and planning. The GIS-based defect registration method consists of two stages.

2) Model Georeferencing: As illustrated in Fig. 1(c), the reconstructed infrastructure model is georeferenced to align with its geographic footprint within the GIS framework. The GIS is implemented using the scalable and robust WebGIS platform, Cesium [56], designed for handling 3-D geospatial data efficiently. The georeferenced infrastructure model serves as a critical foundation for enabling accurate global defect localization and supports the integration of geographic and structural information to enhance maintenance planning.

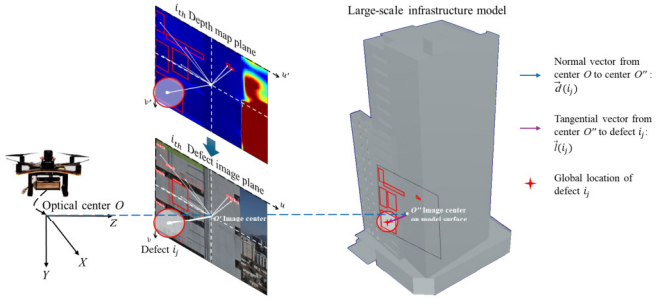


Fig. 7. Geographic projection paradigm for global defect registration.

3) Defect Localization: The projection of surface defects from image coordinates to the georeferenced infrastructure model follows the geographic projection paradigm, as shown in Fig. 7. The geographic coordinates of the image center, O, for the ith image, are first obtained via the RTK positioning system for centimeter-level accuracy. Next, O is translated to O'' along the blue normal vector, using the estimated depth at the image center, to align with the model surface. Finally, the global position of the jth defect, i_j (denoted by the red star), is calculated by translating O'' to the defect's bounding box center along the violet vector, with the metric distance derived from the pixel distance. Moreover, duplicate detections of the same defect are filtered out through geographic comparison. Our method enables the automatic identification of global positions for all detected defects and preserves their geometric and visual properties for reliable maintenance decision-making. Real-world experiments validate the centimeter-level registration accuracy of our method, highlighting its practical applicability for infrastructure maintenance.

III. Benchmark and Ablation Experiments

This section demonstrates the effectiveness and efficiency of the proposed defect detection and infrastructure reconstruction method through extensive experiments on the established defect detection dataset and standard MVS benchmark datasets, while the efficacy of the GIS-based defect registration method is validated in large-scale real-world infrastructure inspections.

A. Detect

1) Dataset and Evaluation Metrics: The proposed defect detection dataset is partitioned into three subsets: 72% of images for training, 8% for validation, and 20% for testing. The mean average precision (mAP) metric is adopted to evaluate defect detection accuracy. Specifically, $\text{mAP}_{0.5}$ (%) measures accuracy with an intersection over union (IoU) threshold of 0.5, indicating how well predicted bounding boxes overlap with ground-truth boxes. In contrast, $\text{mAP}_{0.5:0.95}$ (%) averages detection accuracy across IoU thresholds from 0.5 to 0.95 in 0.05 increments, offering a more comprehensive assessment of model performance across varying precision requirements.

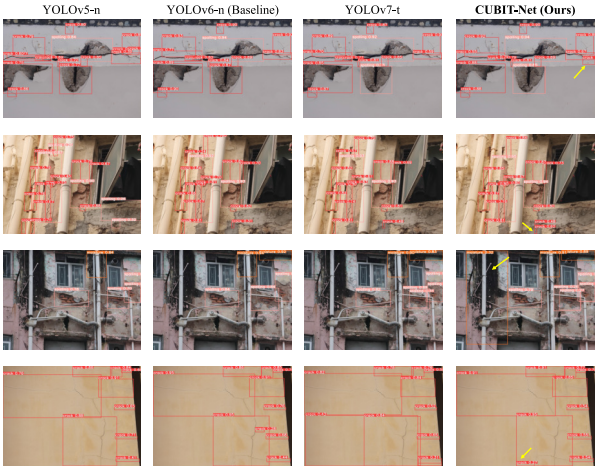


Fig. 8. Detection results on the test set of the CUBIT-Det dataset. Compared to state-of-the-art methods [42], [43], [44], the CUBIT-Net accurately identifies tiny defects (see yellow arrows) in complex and low-texture scenarios.

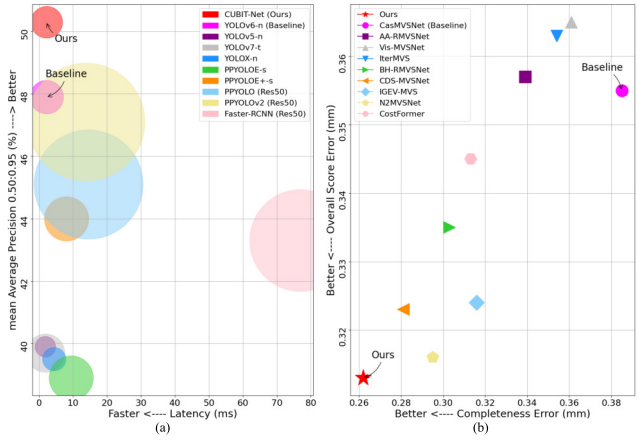


Fig. 9. Benchmark performance comparison of our learning-based (a) defect detection method and (b) reconstruction method.

2) Implementation Details: A total of 19 state-of-the-art real-time object detectors [37], [38], [39], [40], [41], [42], [43], [44], along with the proposed CUBIT-Net, are trained and evaluated on the CUBIT-Det dataset. To enable real-time defect detection in autonomous unmanned systems, we focus on compact and medium-sized networks, including the nano, tiny, small, and medium variants of the YOLO series [38], [39], [40], [41], [42], [43], [44], as well as the ResNet50-based Faster R-CNN [37]. For a fair comparison, all networks are trained and evaluated with input images of size $H \times W = 1024 \times 1024$. The networks are implemented in PyTorch and optimized using stochastic gradient descent (SGD) for 400 epochs, with a batch size of 8, on an NVIDIA RTX 3090Ti GPU. The cosine learning rate scheduler is utilized, with an initial learning rate of 0.02.

3) Benchmark Results: The quantitative benchmark results on the CUBIT-Det test set are summarized in Table II. Compared to state-of-the-art methods [37], [38], [39], [40], [41], [42], [43], [44], CUBIT-Net demonstrates superior $mAP_{0.5:0.95}$ accuracy while maintaining competitive detection speed and lightweight parameters, making it suitable for UAV

TABLE II

QUANTITATIVE BENCHMARK RESULTS ON THE CUBIT-DET TEST SET

Model Type	Methods	#Param. (M) ↓	GFLOPs ↓	$mAP^{test}_{0.5}$ (%) ↑ / $mAP^{test}_{0.5:0.95}$ (%) ↑	Latency (ms) ↓
Medium-Size	Faster R-CNN (Res50) [37]	42.62	477.24	71.5 / 43.3	76.9
	PP-YOLO (Res50) [38]	48.99	136.43	76.4 / 45.1	14.5
	PP-YOLOv2 (Res50) [39]	56.91	146.50	77.3 / 47.1	13.8
	PP-YOLOE-m [40]	24.63	62.93	74.2 / 44.8	11.2
	PP-YOLOE+-m [40]	24.63	62.93	78.8 / 50.9	8.9
	YOLOv5-m [42]	20.86	47.90	80.4 / 51.3	7.1
	YOLOv7 [43]	36.49	61.94	77.5 / 47.8	8.4
	YOLOX-m [41]	25.30	73.80	78.2 / 52.2	13.7
	YOLOv6-m [44]	37.90	225.55	80.4 / 54.1	9.8
	YOLOv6-s [44]	18.50	115.64	79.0 / 48.2	5.3
Compact-Size	PP-YOLOE-s [40]	8.02	20.73	64.6 / 38.9	9.4
	PP-YOLOE+-s [40]	8.02	20.73	70.6 / 44.0	8.1
	YOLOv5-n [42]	**1.76**	**4.10**	73.4 / 39.9	**1.8**
	YOLOv5-s [42]	7.18	15.80	**78.5** / 47.2	3.3
	YOLOv7-t [43]	6.01	13.01	71.1 / 39.7	1.9
	YOLOX-n [41]	2.24	17.75	73.0 / 39.5	4.4
	YOLOX-t [41]	5.03	39.00	75.3 / 49.2	5.8
	YOLOX-s [41]	8.94	68.51	77.9 / 49.4	7.6
	YOLOv6-n (Baseline) [44]	4.63	29.03	76.3 / 47.9	2.2
	CUBIT-Net (Ours)	4.14 **(-0.49)**	28.02 **(-1.01)**	77.5 **(+1.2%)** / **50.3 (+3.1%)**	2.2

TABLE III

ABLATION EXPERIMENTS ON MODULES OF THE PROPOSED CUBIT-NET

Method	Non-linear Act.	Conv.	Norm.	$mAP^{test}_{0.5}$ (%) ↑ / $mAP^{test}_{0.5:0.95}$ (%) ↑	Latency (ms) ↓	#Param. (M) ↓	GFLOPs ↓
Baseline (YOLOv6-n)				76.3% / 47.9%	2.23	4.63	29.03
Baseline + GIPFPP (GELU)	✓			77.4% / 49.0%	2.26	4.63	29.03
Baseline + GIPFPP (GELU + Packet_Conv)	✓	✓		77.2% / 49.2%	**2.17**	4.14	28.02
CUBIT-Net (GELU + Packet_Conv + Group Norm.)	✓	✓	✓	**77.5% / 50.3%**	2.22	**4.14**	**28.02**

deployment. As shown in Fig. 8, CUBIT-Net outperforms state-of-the-art models [42], [43], [44] in complex residential walls (first to third rows) and low-texture facades (fourth row), achieving a higher recall rate and detecting finer defects, as indicated by the yellow arrows. To intuitively assess model performance, we visualize latency (X-axis), $mAP_{0.5:0.95}$ (Y-axis), and model parameters (circle size) in Fig. 9(a). Models closer to the top-left corner indicate superior detection speed and higher accuracy. A comparison with the smallest models from various methods [37], [38], [39], [40], [41], [42], [43], [44] demonstrates that our approach strikes a superior balance between detection speed, accuracy, and model compactness.

4) Ablation Study: Table III provides a comprehensive summary of the ablation experiments conducted to evaluate the effectiveness and efficiency of the proposed CUBIT-Net. Incorporating the GIPFPP module with the GELU activation function improves defect detection accuracy, measured by $mAP_{0.5}$ and $mAP_{0.5:0.95}$, with a negligible impact on detection speed. The introduction of Packet_Conv layers further reduces the model parameters by 10.6% while maintaining detection accuracy. Moreover, group normalization enhances the $mAP_{0.5}$ and $mAP_{0.5:0.95}$ scores by alleviating the model sensitivity to batch size. Overall, replacing the original feature fusion layer with the GIPFPP module achieves a 2.4% increase in $mAP_{0.5:0.95}$ and a 10.6% reduction in model parameters.

Further ablation experiments evaluating the effectiveness of the proposed GIPFPP module in the CUBIT-Net are presented in Tables IV–VI, focusing on nonlinear activation functions, convolutional structures, and group normalization settings, respectively. Table IV compares defect detection performance with different activation functions, including ReLU, GELU, SiLU, HardSwish, and Mish. Among these, GELU achieves the highest accuracy, attributed to its smooth and continuous output that stabilizes gradient propagation and enhances fea-

TABLE IV
ABLATION EXPERIMENTS ON THE NONLINEAR ACTIVATION FUNCTION

Non-Linear Activation	$mAP^{feat}_{0.5}\uparrow$	$mAP^{feat}_{0.5:0.95}\uparrow$	Crack		Spalling		Moisture		Latency (ms)↓
			$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	
ReLU (Baseline)	76.3%	47.9%	80.6%	49.2%	88.8%	60.9%	59.5%	33.6%	2.23
GELU	77.4%	49.0%	80.4%	49.6%	88.9%	61.2%	63.0%	36.3%	2.26
SiLU	76.7%	47.1%	77.7%	46.5%	83.3%	56.8%	69.0%	38.0%	2.26
HardSwish	76.8%	47.9%	78.4%	46.9%	85.2%	58.4%	66.7%	38.4%	2.23
Mish	77.4%	48.8%	81.4%	49.6%	88.1%	60.8%	62.8%	36.0%	2.34

TABLE V
ABLATION EXPERIMENTS ON THE CONVOLUTIONAL STRUCTURE

Group Number	$mAP^{feat}_{0.5}\uparrow$	$mAP^{feat}_{0.5:0.95}\uparrow$	Crack		Spalling		Moisture		Latency (ms) ↓	#Param. (M) ↓	GFLOPs ↓
			$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$			
1 (Initial Baseline)	77.4%	49.0%	80.4%	49.6%	88.9%	61.2%	63.0%	36.3%	2.26	4.63	29.03
4 (New Baseline)	77.4%	49.4%	80.1%	48.6%	87.8%	60.2%	64.5%	39.4%	2.24	4.25	28.25
8	77.0%	49.2%	80.0%	49.5%	87.5%	60.0%	63.5%	38.3%	2.19	4.19	28.12
16	76.6%	48.4%	80.1%	49.4%	87.5%	58.8%	62.2%	37.0%	2.14	4.15	28.04
32	76.5%	48.1%	81.5%	47.6%	89.1%	60.2%	58.7%	36.5%	2.12	4.13	27.99
4 (*Packet_Conv*)	77.2%	49.2%	80.1%	49.5%	86.8%	60.0%	64.6%	38.3%	2.17	4.14	28.02

TABLE VI
ABLATION EXPERIMENTS ON GROUP NORMALIZATION

Group Number	$mAP^{feat}_{0.5}\uparrow$	$mAP^{feat}_{0.5:0.95}\uparrow$	Crack		Spalling		Moisture		Latency (ms)↓
			$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	$AP^{feat}_{0.5}\uparrow$	$AP^{feat}_{0.5:0.95}\uparrow$	
1 (Baseline)	77.2%	49.2%	80.1%	49.5%	86.8%	60.0%	64.6%	38.3%	2.17
4	77.3%	49.4%	80.3%	49.5%	88.8%	61.3%	62.9%	37.3%	2.34
8	77.6%	49.8%	80.2%	49.7%	88.9%	60.6%	63.7%	39.0%	2.27
16	77.4%	49.4%	80.1%	48.5%	88.5%	62.1%	63.5%	37.5%	2.25
32	77.5%	50.3%	80.2%	49.9%	89.6%	61.6%	62.7%	39.5%	2.24

ture learning. Consequently, GELU is selected for the GIPFPP module.

Table V summarizes the impact of varying group numbers in the Packet_Conv module to reduce model size while maintaining detection accuracy. Increasing the group number yields diminishing returns, with less significant parameter reductions and notable accuracy degradation. The setting of four groups achieves the optimal balance, effectively reducing model parameters and GFLOPs while improving detection accuracy. Moreover, incorporating depthwise convolutions within each group further reduces model parameters and GFLOPs, with less compromise to detection accuracy.

To overcome the limitations of batch normalization with varying batch sizes and enhance detection accuracy, batch normalization is replaced with group normalization, integrating feature information across spatial locations by reorganizing and normalizing feature maps. As shown in Table VI, the model achieves optimal $mAP_{0.5:0.95}$ when the group number is set to 32, consistent with the findings in the original study [57].

B. Reconstruct

1) Datasets: The DTU dataset [23] comprises 119 object-centric scenes, with multi-view images captured from fixed camera positions under seven distinct illumination conditions. Ground-truth point clouds are provided for quantitative evaluation of point cloud reconstruction performance. In line with standard practice, the dataset is partitioned into 79 scenes for training, 22 for validation, and 18 for testing. The BlendedMVS dataset [25] features 113 multiscale indoor and outdoor scenes, encompassing over 17000 images paired with corresponding depth maps, designed for fine-tuning MVS models to generalize depth estimation and point cloud reconstruction in real-world applications. The dataset is split into 106 scenes for fine-tuning and seven scenes for evaluating depth estimation performance. The Tanks and Temples (TNT) [24] dataset serves as a public benchmark for assessing point cloud reconstruction performance. It includes an intermediate set of eight scenes and an advanced set of six scenes, offering variations in depth range, illumination, surface texture, and reflectivity.

2) Evaluation Metrics: The DTU dataset evaluates point cloud reconstruction performance using reconstruction accuracy and completeness, quantified by the mean error distance (in mm, where lower values indicate better performance). For a given target scene, the reconstruction accuracy is defined as

$$e_{\mathbf{r}\to\mathcal{G}} = \min_{\mathbf{g}\in\mathcal{G}} \|\mathbf{r}-\mathbf{g}\|_2, \tag{17}$$

$$\text{Acc} = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}} [e_{\mathbf{r}\to\mathcal{G}} < d]\cdot e_{\mathbf{r}\to\mathcal{G}} \tag{18}$$

where $\mathbf{r}$ represents the point from the reconstructed point cloud $\mathcal{R}$ and $\mathbf{g}$ denotes the point from the ground-truth point cloud $\mathcal{G}$. $\|\cdot\|_2$ indicates the Euclidean distance and $e_{\mathbf{r}\to\mathcal{G}}$ represents the minimum Euclidean distance between the point $\mathbf{r}$ and the nearest point in $\mathcal{G}$. The threshold d defines the maximum acceptable error for a point to be considered accurately reconstructed. The Iverson bracket is denoted as $[\cdot]$ and $|\cdot|$ indicates the cardinality of the set of points. Similarly, the reconstruction completeness is determined as

$$e_{\mathbf{g}\to\mathcal{R}} = \min_{\mathbf{r}\in\mathcal{R}} \|\mathbf{g}-\mathbf{r}\|_2 \tag{19}$$

$$\text{Comp} = \frac{1}{|\mathcal{G}|}\sum_{\mathbf{g}\in\mathcal{G}} [e_{\mathbf{g}\to\mathcal{R}} < t]\cdot e_{\mathbf{g}\to\mathcal{R}} \tag{20}$$

where $e_{\mathbf{g}\to\mathcal{R}}$ denotes the minimum Euclidean distance between the point $\mathbf{g}$ and the nearest point in $\mathcal{R}$ while t defines the maximum acceptable error for a point to be considered completely reconstructed. A tradeoff exists between reconstruction accuracy and completeness: accuracy is maximized by sparse but precisely localized point clouds, while completeness is maximized by dense point clouds that cover the entire space. To provide a summary measure of reconstruction performance across multiple scenes, the overall score is computed as

$$\text{Overall} = \frac{1}{2N_t}\sum_{i=0}^{N_t-1}(\text{Acc}_i + \text{Comp}_i) \tag{21}$$

where N_t is the number of scenes.

Similar to the DTU dataset, the TNT dataset uses precision and recall in terms of mean error percentage (in %, where higher values indicate better performance) to quantify point cloud reconstruction accuracy and completeness. The harmonic mean of precision and recall is defined as the F-score and the mean F-score across multiple scenes serves as the overall summary measure of reconstruction performance. In contrast to the DTU and TNT datasets, the BlendedMVS dataset evaluates depth estimation performance using the 1-threshold error (e_1), 3-threshold error (e_3), and endpoint error (EPE). The e_1 and e_3 metrics compute the pixel percentages

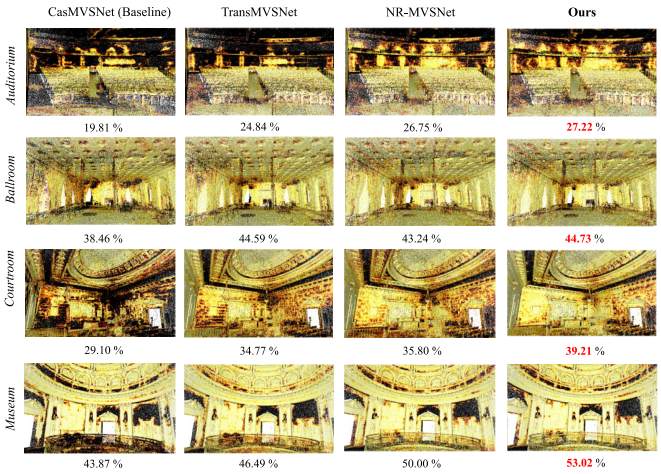


Fig. 10. Reconstruction errors rendering for multiple scenes, exhibiting variations in depth ranges, surface textures, and illumination conditions from the advanced set of the TNT benchmark. Darker regions indicate higher reconstruction errors and the number represents F-score.

where the absolute depth errors, scaled by the depth interval, exceed thresholds of 1 and 3, respectively, while EPE represents the mean absolute scaled depth error.

3) Implementation Details: The proposed MVS network is trained on the DTU training set. Screened Poisson surface reconstruction and depth rendering are applied to acquire ground-truth depth maps [17], [18], [19] to enable end-to-end training. Following standard practice, the depth range at the coarse level, $[d_{\min,2}, d_{\max,2}]$, is set as [425 mm, 935 mm]. The number of depth intervals M_2, M_1, and M_0 are set to 47, 31, and 7, respectively, while the depth intervals I_2, I_1, and I_0 are set to 4, 2, and 1 times the value of I_2. For a fair comparison with state-of-the-art methods, the number of views N is set to 5, and the image resolution is specified as $H \times W = 512 \times 640$. The network is implemented in PyTorch, with the Adam optimizer configured using $\beta_1 = 0.9$ and $\beta_2 = 0.999$. A batch size of 2 is used, and training is performed for 60 epochs on two NVIDIA RTX 3090Ti GPUs. The cosine learning rate scheduler is applied, with an initial learning rate of 0.001.

4) Benchmark Results: To evaluate reconstruction performance, the proposed method is first benchmarked on the DTU evaluation set (22 scenes) following the standard evaluation protocol. With an image resolution of $H \times W = 864 \times 1152$, the number of views N is set to 7 for depth map estimation. The probabilistic constraint ($\tau = 0.1$) is then applied, and the geometric constraint ($N_c = 3$) is enforced to refine and fuse the depth estimates into the point cloud reconstruction. To enhance the generalization ability for large-scale scenes with diverse camera trajectories, the trained model is fine-tuned on the BlendedMVS training set with an image resolution of $H \times W = 576 \times 768$ and $N = 7$. The fine-tuned model is subsequently benchmarked on the TNT intermediate set and the advanced set, with N set to 11. Benchmark results on the DTU and TNT datasets, as shown in Table VII, demonstrate that the proposed method outperforms nearly two dozen state-of-the-art methods in terms of reconstruction performance. The superiority of the method is illustrated in Fig. 9(b), where it (bottom-left corner) achieves higher reconstruction completeness and overall score. Fig. 10 displays the rendered reconstruction error on complex scenes from the TNT dataset with varying depth ranges, where the proposed method produces less error compared to state-of-the-art approaches. To evaluate depth estimation performance, the proposed method is benchmarked on the BlendedMVS validation set with an image resolution of $H \times W = 576 \times 768$ and $N = 5$ for a fair comparison. The results presented in Table VIII demonstrate that the method achieves more accurate and complete depth estimates.

5) Ablation Study: Table IX summarizes the results of ablation experiments evaluating the effectiveness of individual modules in the proposed MVS network. In comparison to the baseline method [18], integrating the proposed BPA significantly enhances point cloud reconstruction completeness

TABLE VII

QUANTITATIVE BENCHMARKING RESULTS ON THE DTU AND TNT DATASETS FOR EVALUATING POINT CLOUD RECONSTRUCTION PERFORMANCE

Methods	Year	*DTU* Evaluation Set (22 Small-Scale Scenes)			*Tanks and Temples* (14 Large-Scale Scenes)	
		Accuracy (mm) ↓	Completeness (mm) ↓	**Overall Score** (mm) ↓	**Mean F-score** (%) ↑ (Intermediate Set)	**Mean F-score** (%) ↑ (Advanced Set)
Colmap [58]	2016	0.400	0.664	0.535	42.14	27.24
R-MVSNet [54]	2019	0.385	0.459	0.422	48.40	24.91
AttMVS [49]	2021	0.383	0.329	0.356	60.05	31.93
PatchMatchNet [59]	2021	0.427	0.277	0.352	53.15	32.31
AA-RMVSNet [51]	2021	0.376	0.339	0.357	61.51	33.53
EPP-MVSNet [60]	2021	0.413	0.296	0.355	61.68	35.72
CDS-MVSNet [61]	2022	0.365	0.281	0.323	61.58	-
NP-CVP-MVSNet [62]	2022	0.356	0.275	0.315	59.64	-
IterMVS [63]	2022	0.373	0.354	0.363	56.94	34.17
BH-RMVSNet [64]	2022	0.368	0.303	0.335	61.96	34.81
IS-MVSNet [65]	2022	0.351	0.359	0.355	62.82	34.87
TransMVSNet [19]	2022	0.360	0.271	0.316	**63.52**	37.00
Vis-MVSNet [50]	2023	0.369	0.361	0.365	60.03	33.78
DCS-MVSNet [66]	2023	0.316	0.372	0.344	53.48	-
IGEV-MVS [67]	2023	0.331	0.316	0.324	-	-
N2MVSNet [21]	2023	0.336	0.295	0.316	62.14	-
CostFormer [68]	2023	0.378	0.313	0.345	57.10	34.31
DispMVS [69]	2023	0.354	0.324	0.339	59.07	34.90
CasMVSNet [18] (Baseline)	2020	0.325	0.385	0.355	56.84	31.12
Ours ($N = 5, N_c = 5$)		**0.285** (-0.04mm)	0.427	0.356		
Ours ($N = 7, N_c = 3$)		0.364	**0.262** (-0.123mm)	**0.313** (-0.042mm)	63.33 (+6.49%)	**38.54** (+7.42%)
Ours ($N = 7, N_c = 4$)		0.317	0.323	0.320		

We first sort the table in chronological order and then sort by Mean F-score of the Advanced Set.
The – denotes that the method does not report the MVS performance on the benchmark.

TABLE VIII

QUANTITATIVE BENCHMARKING RESULTS ON THE BLENDEDMVS VALIDATION SET FOR EVALUATING DEPTH ESTIMATION PERFORMANCE

Methods	EPE ↓	e_1 (%) ↓	e_3 (%) ↓
MVSNet [17]	1.49	21.98	8.32
CVP-MVSNet [70]	1.90	19.73	10.24
CDS-MVSNet [61]	1.80	22.88	9.28
Vis-MVSNet [50]	1.56	21.68	8.36
EPP-MVSNet [60]	1.17	12.66	6.20
UniMVSNet [20]	1.17	11.27	4.96
TransMVSNet [19]	1.05	13.74	5.47
CasMVSNet (Baseline) [18]	1.43	19.73	10.24
Ours	**1.02** (-0.41)	**10.15** (-9.58%)	**4.54** (-5.7%)

TABLE IX

ABLATION EXPERIMENTS ON MODULES OF THE MVS NETWORK ($N = 5$, $W \times H = 1152 \times 864$, $\tau = 0.3$, AND $N_c = 3$)

Models	Feature Extraction		Cost Volume Aggregation		Depth Estimation		Mean Error Distance (mm) ↓		
	FPN	FPN+BPA	Heuristic	Sparse ACVA	Regression	Continuous	Acc.	Comp.	Overall
Baseline (CasMVSNet)	✓		✓		✓		0.364	0.370	0.367
Baseline+BPA		✓	✓		✓		0.364	0.344	0.354
Baseline+BPA+ACVA		✓		✓	✓		**0.348**	0.331	0.340
Baseline+BPA+ACVA+Continuous		✓		✓		✓	0.362	**0.274**	**0.318**

TABLE X
ABLATION EXPERIMENTS ON RUNTIME AND MEMORY

Models	Memory (MB) ↓	Runtime (s) ↓	Mean F-score (%) ↑
Baseline	10427	0.901	44.90
+ABN	9115	0.834	44.90
+ABN+AA	8119	0.944	40.86
+ABN+Sparse ACVA (**Ours**)	**6910**	**0.760**	**45.20**

Within our MVS network, we adopt in-place activated batch normalization (ABN) instead of batch normalization to reduce computational complexity and memory footprint.

by facilitating improved spatial feature transition between the feature pyramid extraction and cost volume aggregation modules, thereby strengthening multi-view correspondence matching. Replacing the heuristic cost volume construction with sparse ACVA further refines reconstruction accuracy and completeness by dynamically assigning scene-adaptive significance to individual viewpoints. Finally, the continuous depth estimation strategy elevates overall performance to state-of-the-art levels by mitigating learning ambiguities in depth estimation. Table X presents the ablation experiments on runtime and memory. Incorporating ABN reduces memory usage by 12.58%, while sparse ACVA achieves an additional 24.19% reduction. Regarding runtime, the ABN and sparse ACVA accelerate inference by 15.65%. Compared to per-pixel adaptive aggregation (AA) modules [20], [51], sparse ACVA demonstrates superior efficiency, significantly reducing memory usage and runtime while achieving a higher mean F-score, underscoring its superior generalization performance.

C. Register

1) Evaluation Metrics: The defect registration accuracy is evaluated using the following metrics: 1) the interquartile range (IQR), which is defined as the difference between the first and third quartiles; 2) the root-mean-square error (RMSE); and 3) the mean absolute error (MAE).

2) Localization Accuracy: First, the GIS environment is established on the scalable WebGIS platform, Cesium [56]. The reconstructed infrastructure model is then georeferenced to align with its geographic footprint. Detected defects, highlighted within the red bounding boxes in Fig. 11(a), are projected onto the geo-referenced model and globally registered within the GIS virtual space, as indicated by the green symbols in Fig. 11(b). Each camera viewpoint is restored within the GIS environment, matching virtual and real viewpoints to compute the defect registration error [as indicated in Fig. 11(c)], defined as the metric distance between the center of the detected bounding box and the localized defect position. Table XI demonstrates the centimeter-level defect registration accuracy of the proposed GIS-based method in real-world inspections.

IV. REAL-WORLD EXPERIMENTS

We deploy our proposed inspection framework on various large-scale scenarios to verify its effectiveness and efficiency. Here, we take a large-scale high-rise warehouse ($L\times$ W $\times$ H = 36 m $\times$ 27 m $\times$ 100 m) as a representative instance.

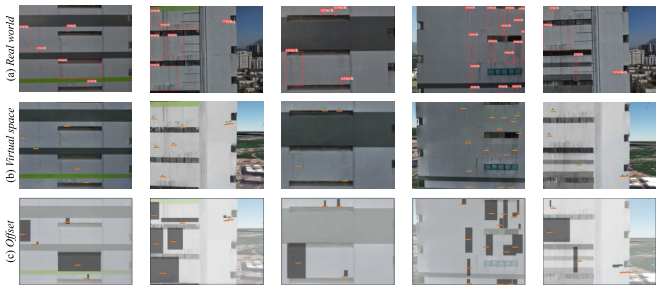


Fig. 11. Defect registration visualizations. (a) Real-world defects identified within the red bounding boxes. (b) Registered virtual defects, represented by green symbols, in the GIS-based virtual space, with orange lines highlighting their positions. (c) Metric offset between the defects in (a) and (b).

TABLE XI
DEFECT REGISTRATION ERROR FOR LARGE-SCALE INFRASTRUCTURE (COMPUTED OVER 923 CLOSE-RANGE FACADE IMAGES)

Registration Error (cm)	Mean ↓	MAE ↓	RMSE ↓	IQR ↓
Horizontal	0.490	2.350	4.746	0
Vertical	0.592	1.037	2.385	0
Diagonal	1.360	4.056	7.149	3.747

A. Multi-UAV Cooperative Coverage Path Planning

As illustrated in Fig. 1, we employ a multi-UAV coverage path-planning strategy to efficiently collect close-range facade images for defect detection and multi-view aerial images for warehouse reconstruction. This approach is based on an explore-then-exploit framework [28], which enables multiple UAVs to safely explore both the target warehouse and its surrounding environment in partially unknown settings while operating within the limitations of sensor capabilities. The process begins with the dynamic updating of a density map as effective viewpoints are acquired. This map subsequently guides each UAV to its optimal target locations. A spatially balanced deployment strategy is employed to ensure optimal sensor coverage across the designated warehouse area. Upon completing the exploration phase, the UAVs gather critical surface information about the target warehouse. For each UAV, specific viewpoints within its operational area are determined. The efficient trajectory to cover all these viewpoints is computed by solving the TSP [26], transforming the complex inspection task into a series of manageable TSP instances, each involving a limited set of viewpoints for computational efficiency. Each UAV then follows the resulting paths to capture the required images for both defect detection and reconstruction. In our real-world experiments, we employ three DJI Mavic 2 Enterprise UAVs, each equipped with a visual camera offering a maximum resolution of H $\times$ W = 3000 $\times$ 4000 pixels and a field of view of 82.6°. The UAVs operate autonomously at a speed of 2 m/s, maintaining an altitude of 30 m above the warehouse roof for multi-view aerial image collection, with the camera oriented in a nadir view (vertically downward). For close-range facade image collection, the UAVs maintain a distance of 10 m from the facade, with the camera positioned perpendicular to the facade surface. This multi-UAV image collection process accelerates data acquisition by more than three times.

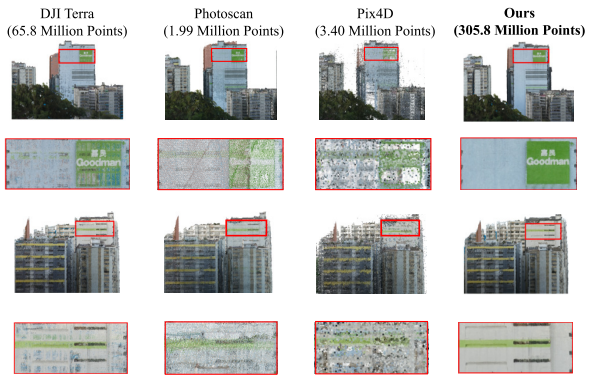


Fig. 12. Point cloud comparison with prevalent industrial reconstruction solutions including DJI Terra, Photoscan, Pix4D, and COLMAP. COLMAP fails to reconstruct due to memory overflow. With the same input images, our learning-based MVS method achieves significantly denser and more complete reconstruction with fine-grained surface texture preserved.

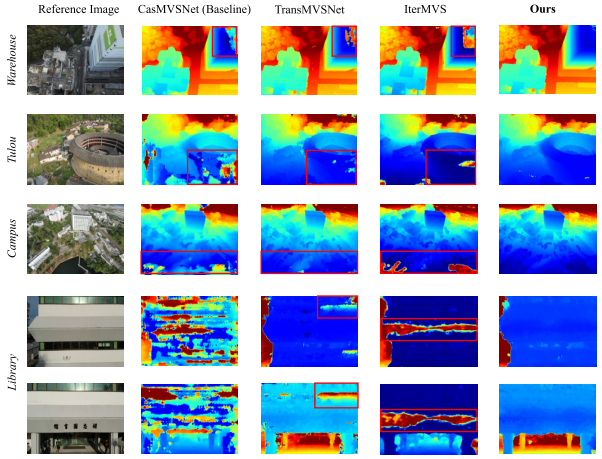


Fig. 13. Depth estimates for infrastructures with varying depth ranges and surface textures. Our method delivers more accurate and complete depth reconstructions compared to existing approaches [18], [19], [63].

TABLE XII

COMPARISON WITH INDUSTRIAL 3-D RECONSTRUCTION SOLUTIONS ON THE ADVANCED SET OF THE TNT BENCHMARK

Methods	F-score (%) ↑						
	Mean	*Auditorium*	*Ballroom*	*Courtroom*	*Museum*	*Palace*	*Temple*
VisualSfM + OpenMVS	12.70	7.94	15.21	21.21	19.78	9.10	2.99
MVE	18.28	4.11	12.63	27.93	34.67	13.58	16.79
OpenMVG + OpenMVS	21.85	9.79	22.49	26.54	36.89	14.64	20.76
OpenMVG + MVE	22.93	14.70	26.36	32.48	37.57	3.65	22.84
COLMAP	27.24	16.02	25.23	34.70	41.51	18.05	27.94
Pix4D	25.07	10.83	18.53	33.21	47.37	14.47	26.01
Ours	**38.54**	**27.22**	**44.73**	**39.21**	**53.02**	**32.73**	**34.33**

B. Effectiveness

With multi-UAV coverage path planning, 923 close-range images of the warehouse ($H \times W = 832 \times 1152$) are efficiently captured. The proposed CUBIT-Net, trained on the CUBIT-Det defect detection dataset, is employed to detect surface defects on the warehouse facade. As shown in the top part of Fig. 1(c), the method effectively identifies cracks, spalling, and moisture, achieving 82% $mAP_{0.5}$ detection accuracy. Meanwhile, 826 multi-view aerial images ($H \times W = 832 \times 1152$) are captured and processed by our MVS method for dense infrastructure reconstruction, as shown in the lower part of Fig. 1(c). Compared to industrial solutions, our learning-based method produces a significantly denser and more complete reconstruction, as demonstrated in Fig. 12, and achieves superior reconstruction accuracy and completeness on the TNT benchmark, as summarized in Table XII. Furthermore, additional experiments in large-scale scenes (Fig. 13) validate that our method delivers more accurate and complete depth estimates than state-of-the-art methods [18], [19], [63], particularly in challenging scenes, such as *Library*, with texture-less and specular surfaces. For global defect registration, our GIS-based method achieves centimeter-level accuracy, as detailed in Section III-C.

TABLE XIII

RECONSTRUCTION RUNTIME ANALYSIS ON REAL-WORLD SCENARIOS FROM SMALL TO LARGE SCALE (TOP TO BOTTOM ROW)

Scene	Image Amount	Resolution	Runtime (mins)			
			SfM	Depth Estimation	Depth Map Fusion	Total
Library	51	1152 × 832	0.555	1.737	0.029	**2.321**
Tulou	248	1152 × 640	4.266	2.163	0.124	**6.553**
Campus	543	1152 × 832	17.316	6.137	0.357	**23.810**
Warehouse	826	1152 × 832	30.947	9.820	4.161	**44.928**

C. Efficiency

The CUBIT-Net model is initially converted to the ONNX format, followed by the creation of a TensorRT inference engine, and deployed on the NVIDIA Jetson Orin NX edge device, achieving a detection speed of 22.7 FPS. The proposed MVS network reconstructs the large-scale warehouse in 44.928 min on a 3090Ti GPU, with 24.378 min for SfM, 6.569 min for view selection, and 13.981 min s for depth estimation and point cloud reconstruction, making it 8.88 times more efficient than DJI Terra, which requires 399 min for the same task. As shown in Table XIII, the efficiency of the MVS network enables the framework to scale effectively across multiscale scenes. The GIS-based defect registration method is completed in 16.426 ms on an i9-10920X CPU, excluding the time required for uploading defect images and the infrastructure model.

D. Scalability

For clarity, scene scales are classified into three categories: 1) small-scale scenes, such as individual statues; 2) medium-scale scenes, such as standalone architectural structures; and 3) large-scale scenes, encompassing groups of architectural structures. As shown in Fig. 14, robust scalability is demonstrated by the proposed MVS method, which adapts effectively to scenes of all scales while maintaining competitive reconstruction performance. This scalability ensures that the proposed inspection framework is suitable for multiscale applications, with the MVS method leveraged to achieve high-quality reconstructions regardless of scene complexity or size. Additionally, the integrated GIS-based defect registration method allows for accurate localization and systematic registration of defects

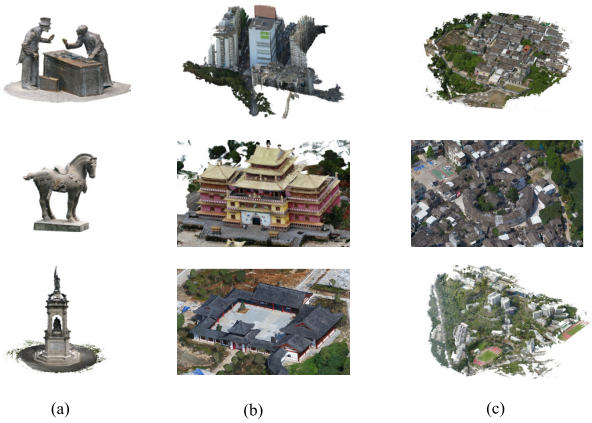


Fig. 14. Point cloud reconstruction on multiscale scenes. (a) Small scale. (b) Medium scale. (c) Large scale.

across varying scales. By combining these capabilities, a robust solution is provided by the proposed framework for the efficient inspection and management of diverse infrastructures, such as buildings, bridges, pavements, and energy sectors, ultimately enhancing safety, efficiency, and cost-effectiveness.

V. Limitations

The limitations of our method are threefold, as described below. First, the proposed defect dataset covers only the three most common surface defect types and does not encompass all possible defect categories. Second, similar to existing approaches, the performance of the proposed learning-based MVS method is sensitive to hyperparameters such as the number of input views, the geometric constraint, and the probabilistic constraint. Finally, the defect localization accuracy of the GIS-based registration method is limited in GPS-denied environments such as indoor scenes and underground facilities.

VI. Conclusion

In this article, we propose Det-Recon-Reg, an intelligent framework designed for automated large-scale infrastructure inspection, addressing the limitations of existing systems that only provide local defect positions and are constrained to small-scale scenes. The proposed framework decomposes the complex inspection process into three stages: defect detection, infrastructure reconstruction, and defect registration. For defect detection, we constructed a large-scale, high-resolution defect dataset to alleviate the challenges of data scarcity in large-scale deployments. We also developed an effective detection method that strikes an excellent balance between accuracy and computational efficiency, making it suitable for UAV onboard deployment. For infrastructure reconstruction, we presented a learning-based MVS network that enables more accurate and complete point cloud reconstruction, serving as the physical foundation for defect localization. For defect registration, we introduced a GIS-based defect registration method, which accurately registers detected defects onto the reconstructed infrastructure model.

Extensive experiments on benchmark datasets and real-world scenarios validate the effectiveness, efficiency, and scalability of the proposed framework. Future work will focus on three main directions: 1) extending the framework to larger-scale areas, such as city blocks; 2) developing an unsupervised learning-based inspection framework to enhance data efficiency; and 3) advancing underwater image enhancement, detection, and reconstruction techniques to adapt the framework for underwater structural inspections.

References

[1] G. Yang et al., "Datasets and processing methods for boosting visual inspection of civil infrastructure: A comprehensive review and algorithm comparison for crack classification, segmentation, and detection," *Construct. Building Mater.*, vol. 356, Nov. 2022, Art. no. 129226.

[2] C. Tang et al., "An inspection robot-based health monitoring method for monorail crane tracks in underground coal mines," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.

[3] R.-J. Yan, E. Kayacan, I.-M. Chen, L. K. Tiong, and J. Wu, "QuicaBot: Quality inspection and assessment robot," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 2, pp. 506–517, Apr. 2019.

[4] K.-W. Tse, R. Pi, W. Yang, X. Yu, and C.-Y. Wen, "Advancing UAV-based inspection system: The USSA-net segmentation approach to crack quantification," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–14, 2024.

[5] H. Xu, J. Cao, Z. Cheng, Z. Liang, and J. Chen, "Design and development of a deformable in-pipe inspection robot for various diameter pipes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 2439–2446.

[6] F. Wang et al., "Internal defect detection of overhead aluminum conductor composite core transmission lines with an inspection robot and computer vision," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–16, 2023.

[7] W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, "Rethinking federated learning with domain shift: A prototype view," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 16312–16322.

[8] P. Luo, B. Wang, H. Wang, F. Ma, H. Ma, and L. Wang, "An ultrasmall bolt defect detection method for transmission line inspection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[9] W. Zhou and J. Hong, "FHENet: Lightweight feature hierarchical exploration network for real-time rail surface defect inspection in RGB-D images," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–8, 2023.

[10] Y. Li, X. Wu, P. Li, and Y. Liu, "Ferrite beads surface defect detection based on spatial attention under weakly supervised learning," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[11] L. Yang et al., "Deep neural network based visual inspection with 3D metric measurement of concrete defects using wall-climbing robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2849–2854.

[12] C. Zhang, M. Jamshidi, C.-C. Chang, X. Liang, Z. Chen, and W. Gui, "Concrete crack quantification using voxel-based reconstruction and Bayesian data fusion," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7512–7524, Nov. 2022.

[13] L. Deng, T. Sun, L. Yang, and R. Cao, "Binocular video-based 3D reconstruction and length quantification of cracks in concrete structures," *Autom. Construction*, vol. 148, Apr. 2023, Art. no. 104743.

[14] Y. Liu, X. Nie, J. Fan, and X. Liu, "Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 5, pp. 511–529, May 2020.

[15] G. Winkelmaier, R. Battulwar, M. Khoshdeli, J. Valencia, J. Sattarvand, and B. Parvin, "Topographically guided UAV for identifying tension cracks using image-based analytics in open-pit mines," *IEEE Trans. Ind. Electron.*, vol. 68, no. 6, pp. 5415–5424, Jun. 2021.

[16] W. Huang et al., "Federated learning for generalization, robustness, fairness: A survey and benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9387–9406, Dec. 2024.

[17] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: Depth inference for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 767–783.

[18] X. Gu, Z. Fan, S. Zhu, Z. Dai, F. Tan, and P. Tan, "Cascade cost volume for high-resolution multi-view stereo and stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2495–2504.

[19] Y. Ding et al., "TransMVSNet: Global context-aware multi-view stereo network with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8585–8594.

[20] R. Peng, R. Wang, Z. Wang, Y. Lai, and R. Wang, "Rethinking depth estimation for multi-view stereo: A unified representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8645–8654.

[21] Z. Zhang, H. Gao, Y. Hu, and R. Wang, "N2MVSNet: Non-local neighbors aware multi-view stereo network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[22] Q. Li et al., "Autonomous design framework for deploying building integrated photovoltaics," *Appl. Energy*, vol. 377, Jan. 2025, Art. no. 124760.

[23] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, vol. 10, pp. 1–16, Jan. 2016.

[24] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.

[25] Y. Yao et al., "BlendedMVS: A large-scale dataset for generalized multi-view stereo networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1787–1796.

[26] C. Gao, W. Ding, Z. Zhao, and B. M. Chen, "Energy-optimal trajectory-based traveling salesman problem for multi-rotor unmanned aerial vehicles," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Dec. 2023, pp. 6110–6115.

[27] Y. Chen, S. Lai, J. Cui, B. Wang, and B. M. Chen, "GPU-accelerated incremental Euclidean distance transform for online motion planning of mobile robots," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6894–6901, Jul. 2022.

[28] C. Gao et al., "A UAV-based explore-then-exploit system for autonomous indoor facility inspection and scene reconstruction," *Autom. Construct.*, vol. 148, Apr. 2023, Art. no. 104753.

[29] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection using deep neural networks with images captured through a smartphone," 2018, *arXiv:1801.09454*.

[30] H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto, and H. Omata, "Generative adversarial network for road damage detection," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 1, pp. 47–60, Jan. 2021.

[31] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2020: An annotated image dataset for automatic road damage detection using deep learning," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107133.

[32] D. Arya, H. Maeda, S. Kumar Ghosh, D. Toshniwal, and Y. Sekimoto, "RDD2022: A multi-national image dataset for automatic road damage detection," 2022, *arXiv:2209.08538*.

[33] H. Majidifard, P. Jin, Y. Adu-Gyamfi, and W. Buttlar, "Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses," *Transp. Res. Rec.*, vol. 2674, no. 2, pp. 328–339, 2020.

[34] M. Al Qurishee, W. Wu, B. Atolagbe, J. Owino, I. Fomunung, and M. Onyango, "Creating a dataset to boost civil engineering deep learning research and application," *Engineering*, vol. 12, no. 3, pp. 151–165, 2020.

[35] M. Sabouri and A. Sepidbar, "SUT-crack: A comprehensive dataset for pavement crack detection across all methods," *Data Brief*, vol. 51, Dec. 2023, Art. no. 109642.

[36] M. Mundt, S. Majumder, S. Murali, P. Panetsos, and V. Ramesh, "Meta-learning convolutional neural architectures for multi-target concrete defect classification with the COncrete DEfect BRidge IMage dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11196–11205.

[37] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[38] X. Long et al., "PP-YOLO: An effective and efficient implementation of object detector," 2020, *arXiv:2007.12099*.

[39] X. Huang et al., "PP-YOLOv2: A practical object detector," 2021, *arXiv:2104.10419*.

[40] S. Xu et al., "PP-YOLOE: An evolved version of YOLO," 2022, *arXiv:2203.16250*.

[41] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[42] G. Jocher et al. (2020). *Ultralytics YOLOv5*. [Online]. Available: https://github.com/ultralytics/yolov5

[43] C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, *arXiv:2207.02696*.

[44] C. Li et al., "YOLOv6 V3.0: A full-scale reloading," 2023, *arXiv:2301.05586*.

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

[46] Q. Xu and W. Tao, "Learning inverse depth regression for multi-view stereo with correlation cost volume," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 12508–12515.

[47] T. Tang, G. Yang, D. Zhang, L. Lei, B. Li, and L. Gao, "A hydrodynamic prediction model of throttle orifice plate using space filling and adaptive sampling method," *Struct. Multidisciplinary Optim.*, vol. 62, no. 3, pp. 1563–1578, Sep. 2020.

[48] K. Luo, T. Guan, L. Ju, H. Huang, and Y. Luo, "P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10451–10460.

[49] K. Luo, T. Guan, L. Ju, Y. Wang, Z. Chen, and Y. Luo, "Attention-aware multi-view stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1587–1596.

[50] J. Zhang, S. Li, Z. Luo, T. Fang, and Y. Yao, "Vis-MVSNet: Visibility-aware multi-view stereo network," *Int. J. Comput. Vis.*, vol. 131, no. 1, pp. 199–214, Jan. 2023.

[51] Z. Wei, Q. Zhu, C. Min, Y. Chen, and G. Wang, "AA-RMVSNet: Adaptive aggregation recurrent multi-view stereo network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6187–6196.

[52] W. Huang, M. Ye, Z. Shi, and B. Du, "Generalizable heterogeneous federated cross-correlation and instance similarity learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 2, pp. 712–728, Feb. 2024.

[53] W. Huang, M. Ye, and B. Du, "Learn from others and be yourself in heterogeneous federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10143–10153.

[54] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent MVSNet for high-resolution multi-view stereo depth inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5520–5529.

[55] X. Li et al., "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 21002–21012.

[56] I. Cesium GS. *Cesium, the Platform for 3D Geospatial*. [Online]. Available: https://www.cesium.com/

[57] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[58] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 501–518.

[59] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "PatchmatchNet: Learned multi-view patchmatch stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14194–14203.

[60] X. Ma, Y. Gong, Q. Wang, J. Huang, L. Chen, and F. Yu, "EPP-MVSNet: Epipolar-assembling based depth prediction for multi-view stereo," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5732–5740.

[61] K. T. Giang, S. Song, and S. Jo, "Curvature-guided dynamic scale networks for multi-view stereo," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–19.

[62] J. Yang, J. M. Alvarez, and M. Liu, "Non-parametric depth distribution modelling based depth inference for multi-view stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8626–8634.

[63] F. Wang, S. Galliani, C. Vogel, and M. Pollefeys, "IterMVS: Iterative probability estimation for efficient multi-view stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8606–8615.

[64] Z. Wei, Q. Zhu, C. Min, Y. Chen, and G. Wang, "Bidirectional hybrid LSTM based recurrent neural network for multi-view stereo," *IEEE Trans. Vis. Comput. Graphics*, vol. 30, no. 7, pp. 3062–3073, Jul. 2024.

[65] L. Wang, Y. Gong, X. Ma, Q. Wang, K. Zhou, and L. Chen, "IS-MVSNet: Importance sampling-based MVSNet," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 668–683.

[66] S. Zhang et al., "DSC-MVSNet: Attention aware cost volume regularization based on depthwise separable convolution for multi-view stereo," *Complex Intell. Syst.*, vol. 9, no. 6, pp. 6953–6969, Dec. 2023.

[67] G. Xu, X. Wang, X. Ding, and X. Yang, "Iterative geometry encoding volume for stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 21919–21928.

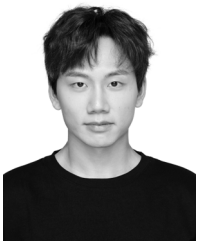
[68] W. Chen et al., "CostFormer: Cost transformer for cost aggregation in multi-view stereo," in *Proc. Int. Joint Conf. Artif. Intell.*, 2023, pp. 1–10.

[69] Q. Yan, Q. Wang, K. Zhao, B. Li, X. Chu, and F. Deng, "Rethinking disparity: A depth range free multi-view stereo based on disparity," in *Proc. AAAI Conf. Artif. Intell.*, 2023, vol. 37, no. 3, pp. 3091–3099.

[70] J. Yang, W. Mao, J. M. Alvarez, and M. Liu, "Cost volume pyramid based depth inference for multi-view stereo," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4877–4886.

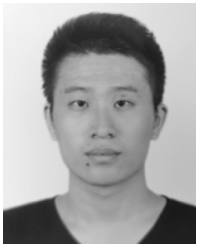
Guidong Yang (Graduate Student Member, IEEE) received the B.Eng. degree in mechanical and automation engineering and the M.Eng. degree in vehicle engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, in 2018, and 2021, respectively and the M.Sc. degree in mechanical engineering from Politecnico di Milano, Milan, Italy, in 2021. He is currently pursuing the Ph.D. degree in mechanical and automation engineering with the Chinese University of Hong Kong, Hong Kong, China.

His current research interests include multi-view stereo and object detection.



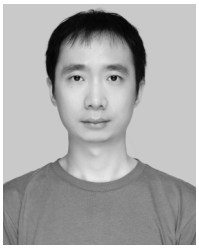
Benyun Zhao received the M.Sc. degree in mechanical and automation engineering from the Chinese University of Hong Kong (CUHK), Hong Kong, China, in 2021, where he is currently pursuing the Ph.D. degree.

He was a Research Assistant with CUHK and Hong Kong Center of Logistics Robotics (HKCLR), Hong Kong, from 2021 to 2022. His current research interests include object detection, semantic segmentation, and 3-D scene understanding.



Jihan Zhang (Member, IEEE) received the B.Eng. degree in automation engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2019, and the M.Sc. degree in mechanical and automation engineering from the Chinese University of Hong Kong (CUHK), Hong Kong, China, in 2020, where he is currently pursuing the Ph.D. degree.

His current research interests include digital twin modeling and simulation.



Junjie Wen (Graduate Student Member, IEEE) received the B.Sc. degree in automotive engineering from Dalian University of Technology, Dalian, China, in 2013, and the M.Sc. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree in mechanical and automation engineering with the Chinese University of Hong Kong, Hong Kong, China.

His research interests include image restoration and novel view synthesis.

Qingxiang Li received the B.Sc. degree in civil engineering and the M.Sc. degree in architecture from Tianjin University (TJU), Tianjin, China, in 2016 and 2019, respectively, and the Ph.D. degree in architectural engineering from Politecnico di Milano, Milan, Italy, in 2023.

He is currently a Post-Doctoral Fellow of mechanical and automation engineering with the Chinese University of Hong Kong, Hong Kong, China. His research interests include multi-view stereo and building automation.

Lei Lei (Member, IEEE) received the B.E. degree in naval architecture and ocean engineering from Harbin Engineering University, Harbin, China, in 2016, the M.E. degree in mechanical engineering from Huazhong University of Science and Engineering, Wuhan, China, in 2019, and the Ph.D. degree in systems engineering from the City University of Hong Kong, Hong Kong, in 2024.

He is a Post-Doctoral Fellow with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include underwater robots, ocean big data, and robotics learning and control.

Xi Chen is currently a Research Assistant Professor of mechanical and automation engineering with the Chinese University of Hong Kong (CUHK), Hong Kong. He has over ten years of experience in sustainable building technology related to urban energy systems, renewable applications in buildings, and built environment modeling and has led or managed multiple research projects including ARC, MOST, RGC, and consultancy projects with the local government and industry. He has published over 40 papers in peer-reviewed international journals and co-authored a book on green building and renewable application areas.

Ben M. Chen (Fellow, IEEE) was an Assistant Professor with the Department of Electrical Engineering, State University of New York at Stony Brook, Stony Brook, NY, USA, from 1992 to 1993. He was a Provost's Chair Professor with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, before joining the Chinese University of Hong Kong (CUHK), Hong Kong, in 2018. He is currently a Professor of mechanical and automation engineering with the CUHK. He has authored/co-authored hundreds of journal and conference articles and a dozen research monographs in control theory and applications, unmanned systems, and financial market modeling. His current research interests are in unmanned systems and their applications.

Dr. Chen is a fellow of the Academy of Engineering, Singapore. He had served on the editorial boards of a dozen international journals including *Automatica* and IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He is currently serving as an Editor-in-Chief of *Unmanned Systems* and Editor of *International Journal of Robust and Nonlinear Control*.