2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Det-Recon-Reg: An Intelligent Framework Towards Automated Large-Scale Infrastructure Inspection

Guidong Yang[1,†], Jihan Zhang[1,†], Benyun Zhao[1,†], Chuanxiang Gao[1], Yijun Huang[1], Junjie Wen[1], Qingxiang Li[1], Jerry Tang[2], Xi Chen[1], and Ben M. Chen[1]

***Abstract*—Visual inspection plays a predominant role in inspecting infrastructure surface. However, the generalization of existing visual inspection systems to large-scale real-world scenes remains challenging. In this paper, we introduce Det-Recon-Reg, an intelligent framework separating the complex inspection procedure into three stages: *Detect*, *Reconstruct*, and *Register*. (1) For defect detection (*Detect*), we present the first high-resolution defect dataset tailored for large-scale defect detection. Based on the dataset, we evaluate the most effective real-time object detection algorithms and push the boundary by proposing CUBIT-Net for real-world defect inspection. (2) For infrastructure reconstruction (*Reconstruct*), we propose a learning-based multi-view stereo (MVS) network to adapt to large-scale scenes, taking as input the multi-view images and outputting the point cloud reconstruction, where its performance has been validated on the standard MVS datasets, including BlendedMVS, DTU, and Tanks and Temples datasets. (3) For defect localization (*Register*), we propose an effective registration method based on the geographic information system that registers the detected defects onto the reconstructed infrastructure model to establish a global reference for maintenance measures. The real-world experiments further verify the effectiveness and efficiency of our proposed framework. More details about our proposed dataset, code, and appendix are available on our project page: *https://cuhk-usr-group.github.io/large-scale-inspect-framework/*.**

I. Introduction and Related Work

Defect diagnosis and monitoring are essential to ensure the functional safety of the infrastructure, with visual inspection being the primary method [1]–[4] to inspect critical surface defects such as cracks and spalling. The integration of unmanned robotic platforms [5]–[7] and learning-based visual inspection methods [8]–[10] provides an effective and efficient alternative to manual visual inspection. Nevertheless, existing inspection systems [1]–[4] only provide the local position of the detected defects and are limited to small-scale scenes such as a wall. Large-scale infrastructure inspection, the core task of which is to accurately and efficiently localize the global position of detected defects, remains an open and challenging question. To answer this question, we propose ***Det-Recon-Reg***, an intelligent framework for large-scale infrastructure inspection based on the unmanned aerial vehicles (UAVs) and learning-based techniques by decoupling the inspection task into a three-stage procedure: ***Detect*** (defect detection), ***Reconstruct*** (infrastructure reconstruction), and ***Register*** (defect localization).

Detect The performance of the existing defect detection methods is limited by the lack of a publicly available high-resolution defect dataset [11]. We thus boost the community by constructing the first high-resolution defect dataset CUBIT-Det[1] aiming for common defects (crack, spalling, moisture) detection on large-scale infrastructures (building, bridge, pavement). Based on the CUBIT-Det, we conduct extensive experiments to benchmark the state-of-the-art (SOTA) real-time detection algorithms [12]–[19] for defect inspection and propose CUBIT-Net to achieve better trade-off between detection accuracy and efficiency.

Reconstruct Dense point cloud model of the infrastructure is reconstructed from multi-view images and serves as the physical entity for defect localization. Existing defect inspection systems [2], [20], [21] adopt geometry-based multi-view stereo (MVS) methods for infrastructure reconstruction. Recently, learning-based MVS methods [22]–[26] significantly outperform the traditional counterparts regarding reconstruction accuracy and completeness. However, to the best of our knowledge, the potential of the learning-based MVS methods has yet to be investigated and applied in the field of infrastructure inspection. To fill this research gap, we propose our MVS network and extensive experiments on the typical MVS datasets, including BlendedMVS, DTU, and Tanks and Temples datasets, demonstrate the SOTA performance of our proposed method. We further deploy the proposed method into real-world reconstruction to demonstrate its effectiveness, efficiency, and scalability to large-scale scenes.

Register Existing inspection systems [1]–[4] conduct defect segmentation and back-project the segmented defect from the image pixel onto the reconstructed point cloud [1], [3] or triangular surface model [2], [4]. However, the traditional segment-then-project localization method only provides the local position of the inspected defects with qualitative visualization, and subsequent maintenance measures are hard to be taken without global localization. The related research has not provided a robust global understanding for the integration of defects and model in a three-dimensional representation. To address this issue, we propose an effective registration method based on geographic information system (GIS) that accurately registers detected defects onto the reconstructed model with geographic information. To our best knowledge, we are the first to leverage the GIS in defect localization task, paving the

† Equal contribution.

This work was supported by the InnoHK of the Government of the Hong Kong Special Administrative Region via the Hong Kong Center for Logistics Robotics (*Corresponding Author: Jihan Zhang*).

[1]The authors are with the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong (e-mail: {gdyang, jhzhang, byzhao, cxgao, yjhuang, jjwen, xichen, bmchen}@mae.cuhk.edu.hk, qingxiang.li@polimi.it)

[2]The author is with the High Performance Robotics Lab, Department of Mechanical Engineering, University of California Berkeley, USA (e-mail: jerrytang@berkeley.edu)

[1]CUBIT stands for CUHK Building Information Technology.

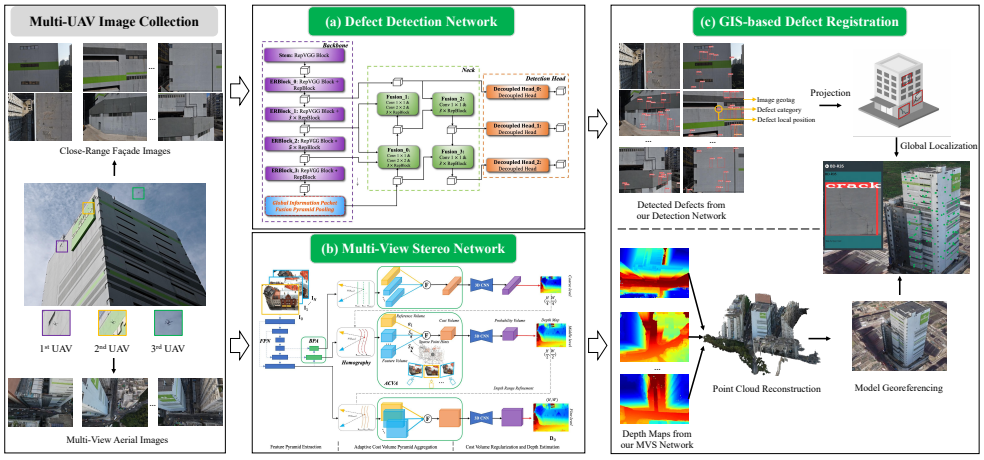


Fig. 1. The structure of the proposed ***Det-Recon-Reg*** framework for large-scale infrastructure inspection. We adopt multi-UAV coverage path planning to collect multi-view images for reconstruction and close-range facade images for surface defect detection. ***Detect***: We deploy the proposed CUBIT-Net trained on our proposed CUBIT-Det dataset to detect surface flaws. ***Reconstruct***: We leverage the proposed MVS network to predict multi-view depth maps and fuse them to reconstruct the infrastructure. ***Register***: We identify the global position of the detected defects based on GIS.

way for future maintenance measures by establishing global defect reference.

To this end, we take a solid step towards automating large-scale infrastructure inspection by presenting a ***Detect-Reconstruct-Register (Det-Recon-Reg)*** inspection framework (as shown in Fig. 1). Extensive experiments on the defect detection dataset and standard MVS datasets validate the effectiveness and efficiency of the proposed defect detection and learning-based MVS method, respectively. We also apply our framework to real-world inspection to verify its efficacy and robustness in defect detection, infrastructure reconstruction, and defect localization. Our main contributions are fourfold as follows:

- We propose a ***Det-Recon-Reg*** inspection framework to automate large-scale infrastructure inspection. The effectiveness and efficiency of this framework have been verified in the real-world inspection task.
- We construct the first high-resolution defect detection dataset, named CUBIT-Det. Based on it, we propose CUBIT-Net for real-world defect detection.
- We propose a learning-based MVS network for point cloud reconstruction and further deploy it into real-world application for large-scale infrastructure reconstruction.
- We present a GIS-based defect registration method to accurately localize the detected defects onto the reconstructed infrastructure model.

II. Methodology

In this section, we successively illustrate the methodology for each stage of the proposed ***Det-Recon-Reg*** inspection framework as shown in Fig. 1, where we rely on multi-UAV coverage path planning [27] to acquire multi-view aerial images for reconstruction and close-range facade images for defect detection.

A. Detect

Dataset Establishment The performance of existing methods on large-scale defect detection is hampered by the lack of a high resolution open source dataset [11]. We therefore established the first high-resolution dataset tailored for large-scale infrastructure defect detection, named CUBIT-Det. The dataset comprises $5,527$ images with a maximum resolution of 8000×6000 captured by the onboard cameras of the UAVs. The data set covers three of the most ubiquitous types of infrastructure, including buildings (65%), pavement (29%), and bridge (6%), and targets for inspecting the three most critical types of surface defect, including crack (82%), spalling (12%), and moisture (6%). The high-resolution images are collected from different viewpoints and distances under various illumination conditions, inherently offering more structural context information and model robustness for real-world inspection.

Detection Method Based on CUBIT-Det dataset, we benchmark SOTA real-time detection methods [13], [14], [16]–[19] to evaluate their performance for large-scale defect detection. Based on the benchmark results, we choose YOLOv6-n [19] as the basis since it achieves the best trade-off between the detection accuracy and latency. To further amplify this advantage, as shown in blue box of Fig. 1(a), we replace the original pyramid pooling module by our proposed *Global Information Packet Fusion Pyramid Pooling* (GIPFPP) module. This module has a fast processing speed while fusing multi-scale information. The structure of the proposed GIPFPP module is detailed in Fig. 2. This module bifurcates the input feature map along the channel dimension into a top and bottom branch. The feature map traverses the top branch through three $Packet\_Conv$, three Maxpooling layers, and two $Packet\_Conv$ successively, and merges with the feature map from the bottom branch that passes through one $Packet\_Conv$. We name this upgraded model as CUBIT-Net.

The $Packet\_Conv$ is similar to usual convolutional structure comprising three operations: convolution, normalization, and non-linear activation. In the convolution phase, we employ a grouped depth-wise convolution technique: the input feature map is evenly divided into 4 groups along the channel

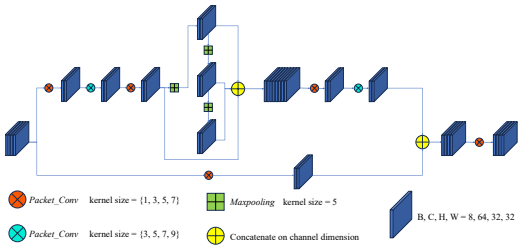
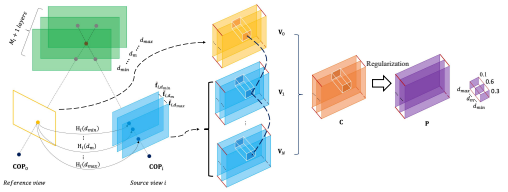


Fig. 2. The structure of proposed GIPFPP module.



(a) Homography warping (b) Cost volume aggregation (c) Regularization and depth estimation

Fig. 3. The schematic demonstration for (a) homography warping, (b) cost volume aggregation, (c) regularization and depth estimation. COP denotes the center of projection.

dimension. The size of the convolution kernel for the i_{th} group is either $2i-1$ ($\{1,3,5,7\}$) (carrot-orange circle) or $2i+1$ ($\{3,5,7,9\}$) (indigo circle). Within each group, depthwise convolution is performed, and all the feature maps are concatenated back to the original dimension. Large kernel excels in extracting global features from large receptive field, while small kernel specializes in extracting local features from small receptive field, which are complementary. Although the introduction of larger convolution kernels imply more parameters, the way of 'packaging the feature map then convolving depthwise' effectively reduces the number of parameters and latency. Besides, we utilize group normalization instead of batch normalization in the $Packet\_Conv$ to reduce sensitivity to batch size changes. GELU (Gaussian error linear unit) is adopted as the activation function as it offers superior smoothness and alleviates vanishing gradient issues compared to most commonly used ReLU. The benchmark and ablation experiments in Section III-A demonstrate the effectiveness of the proposed CUBIT-Net and GIPFPP module. We deploy CUBIT-Net trained on CUBIT-Det to perform the real-world defect inspection task.

B. Reconstruct

Method Overview Learning-based MVS methods [22]–[26] have significantly outperformed the traditional counterparts in terms of point cloud reconstruction accuracy and completeness by separating the MVS into a two-stage procedure including *learning-based multi-view depth estimation* and *multi-view depth map fusion*. We propose our learning-based MVS network (see Fig. 1(b)) tailored for large-scale infrastructure reconstruction by following the coarse-to-fine depth estimation paradigm [23]. Given an unordered set of multi-view images $\{\mathbf{I}_i\}_{i=0}^{N}$ acquired from $(N+1)$ viewpoints, our network infers the depth map $\mathbf{D}_0$ for the reference image $\mathbf{I}_0$ with N neighboring source images $\{\mathbf{I}_i\}_{i=1}^{N}$. The input images are iteratively treated as the reference image to predict the per-view depth maps $\{\mathbf{D}_i\}_{i=0}^{N}$, which are fused to the final point cloud reconstruction. Our network comprises three modules: 1) Enhanced multi-scale feature pyramid extraction; 2) Sparse point-guided adaptive cost volume aggregation (Sparse ACVA); 3) Cost volume regularization and continuous depth estimation.

Enhanced Multi-Scale Feature Pyramid Extraction Given multi-view images $\{\mathbf{I}_i\}_{i=0}^{N}$ and their camera intrinsics and extrinsics, most recent MVS methods [22]–[26] leverage the feature pyramid network (FPN) to extract multi-scale $(L+1$ scale) features $\{\mathbf{f}_{l,i} \in \mathbb{R}^{F_l \times H/2^l \times W/2^l}\}_{l=0}^{L}$ for each input image $\mathbf{I}_i$, where F_l is the channel number at the l_{th} level ($l = 0, 1, 2$), H and W is the height and width of the input image, respectively. We observe that the depth estimation around the object boundary of the specular and low-textured surfaces confronts with over-smoothing estimation issues (see Fig. 9 of the **Appendix**) due to the lack of low-level spatial features including local edges and textures. We hence introduce a four-layer bottom-up pathway (BPA, refer to Fig. 7 of the **Appendix** for detailed configuration) to augment and smooth the transition of the spatial features between the feature extraction module and the cost volume aggregation module, shown to achieve more accurate and complete depth estimation and point cloud reconstruction through ablation study.

Sparse Point-Guided Adaptive Cost Volume Aggregation For feature level l, given extracted deep features $\{\mathbf{f}_{l,i}\}_{i=0}^{N}$, we first uniformly discretize the depth range $[d_{min,l}, d_{max,l}]$ of the reference-view 3D space into (M_l+1) depth hypotheses $d_{m,l} = d_{min,l} + m \cdot (d_{max,l} - d_{min,l})/M_l,\ m \in \{0, 1, ..., M_l\}$ (green rectangles in Fig. 3(a)). As multi-view stereo is essentially equivalent to solving the pixel correspondences across multi-view images, we adopt the homography to establish the pairwise pixel correspondences between the reference-view feature map $\mathbf{f}_{l,0}$ and the source-view feature map $\mathbf{f}_{l,i}$, where each $d_{m,l}$ determines a homography between the reference-view pixel $\mathbf{p}_{l,0}$ and i_{th} source-view pixel $\mathbf{p}_{l,i}$:

$$\mathbf{p}_{l,i} = \mathbf{K}_i[\mathbf{R}(\mathbf{K}_0^{-1}\mathbf{p}_{l,0}d_{m,l}) + \mathbf{t}] \tag{1}$$

where $\mathbf{K}_0$ and $\mathbf{K}_i$ denote the scaled intrinsic camera parameters for reference and i_{th} source view. $\mathbf{R}$ and $\mathbf{t}$ refer to the relative rotation and translation between the two views. To establish feature correspondence, we adopt differentiable bilinear interpolation to sample pixels from $\mathbf{f}_{l,i}$, where $\mathbf{p}_{l,i}$ specifies pixel location. After interpolation, we derive the warped source-view feature map $\widetilde{\mathbf{f}}_{l,i,d_{m,l}}$ corresponds to reference-view feature map $\mathbf{f}_{l,0}$ under the depth hypothesis $d_{m,l}$.

We repeat the above process for each depth $d_{m,l}$ from (M_l+1) depth hypotheses, i.e., we conduct the homography transformation and feature warping (M_l+1) times to get (M_l+1) warped feature maps $\widetilde{\mathbf{f}}_{l,i,d_{m,l}}$ (blue rectangles in Fig. 3(a)) aligned to the reference feature map $\mathbf{f}_{l,0}$ (yellow rectangle in Fig. 3(a)). After feature warping, we stack the warped feature maps along the depth dimension to get the source-view feature volume $\{\mathbf{V}_{l,i} \in \mathbb{R}^{F_l \times (M_l+1) \times H/2^l \times W/2^l}\}_{i=1}^{N}$ (blue volume in Fig. 3(b)) for the i_{th} source view and stack the reference feature map $\mathbf{f}_{l,0}$ (M_l+1) times along the depth dimension to get the reference-view feature volume $\mathbf{V}_{l,0}$ (yellow volume in Fig. 3(b)).

We then adopt the proposed Sparse ACVA to adaptively aggregate multi-view feature volumes into a single cost volume $\mathbf{C}_l$ (orange volume in Fig. 3(c)) measuring multi-view feature matching similarity. We observe that images from different perspectives have pixel differences due to variations in illumination, occlusions, and image content. In addition, a source image close to the reference view that is devoid of occlusion can provide more precise photometric and geometric information than a distant image with partial occlusion. We hence propose Sparse ACVA to compute the source-view weight based on the sparse point reconstruction given by structure-from-motion (SfM) and learn the reference-view weight from the training data. In this way, we adapt to scene variation compared to heuristic-based aggregation methods [28], [29] and alleviate the computational cost compared to the re-weight network-based methods [30]–[32]. The Sparse ACVA is formulated as follows:

$$\begin{aligned}\mathbf{C}_l &= \mathcal{M}(\mathbf{V}_{l,0}, \cdots, \mathbf{V}_{l,N}) \\ &= \mathcal{M}(\mathbf{B}_{l,0}, \cdots, \mathbf{B}_{l,N}) \\ &= AvgPool(\alpha_l \mathbf{B}_{l,0} \odot \sum_{i=1}^{N} \frac{S_i}{\sum_{i=1}^{N} S_i} \mathbf{B}_{l,i})\end{aligned} \tag{2}$$

where $\mathbf{B}_{l,i} \in \mathbb{R}^{K \times (F_l/K) \times (M_l+1) \times H/2^l \times W/2^l}$ is the batched volume by separating $\mathbf{V}_{l,i}$ into K batches along the channel dimension, where setting K as the small integer reduces memory footprint and speeds up the inference speed. α_l is the learnable parameter to learn reference-view weight and $\{S_i\}_{i=1}^N$ are the scene-dependent feature matching scores between $\{\mathbf{I}_i\}_{i=1}^N$ and $\mathbf{I}_0$ computed based on their common sparse point hints from the SfM. We adopt the normalized feature matching score as the source-view weight. The $\odot$ denotes the Hadamard product measuring the feature matching similarity between the reference view and neighboring source views. We further aggregate them via average pooling along the channel dimension to get the final cost volume $\mathbf{C}_l \in \mathbb{R}^{K \times (M_l+1) \times H/2^l \times W/2^l}$.

The Algorithm 1 elaborates the pseudo process for computing the scene-dependent matching score $\{S_i\}_{i=1}^N$ between the source-view image $\{\mathbf{I}_i\}_{i=1}^N$ and reference-view image $\mathbf{I}_0$. $\{\mathbf{p}_{ij} \in \mathbb{R}^{3\times1}, j \in \{0, 1, ..., n_i - 1\}\}_{i=1}^N$ denotes the inhomogeneous coordinates of the sparse points triangulated by reference image $\mathbf{I}_0$ and i_{th} source image $\{\mathbf{I}_i\}_{i=1}^N$ and n_i represents the number of points for the i_{th} source image. $\{\mathbf{c}_0, \mathbf{c}_i\} \in \mathbb{R}^{3\times1}$ stands for the inhomogeneous coordinate of the center of projection for the reference view and source view, respectively. The baseline angle of $\mathbf{p}_{ij}$ is denoted as θ_j. To enable our network to adapt to the input scene variation, we first accumulate the S_i based on a piecewise gaussian function which favors the baseline angle θ_0 and then normalize S_i as $\frac{S_i}{\sum_{i=1}^N S_i}$ to serve as the i_{th} source-view weight.

Cost Volume Regularization and Continuous Depth Estimation Following the common practices [22]–[26], for feature level l, we adopt a multi-scale 3D CNN to regularize the noise-contaminated cost volume $\mathbf{C}_l$ to predict the probability volume $\mathbf{P}_l$ with same dimension (violet volume in Fig. 3(c)). Existing MVS methods conduct depth estimation through regression [22] (for each pixel, its depth estimation is the weighted sum of the depth hypotheses, where the weight is the probability from $\mathbf{P}_l$) or classification [33] (for each pixel, its depth estimation corresponds to the depth hypothesis with the maximum probability). The regression suffers from ambiguity caused by the imbalance between the single optimal depth value and various possible weight combination while the classification leads to discrete depth estimation. To tackle this issue, we refine the discrete depth estimation by adding the depth residual between the target and discrete depth to achieve continuous depth estimation:

Algorithm 1: Matching Score Computation

Input : $\{\mathbf{p}_{ij} \in \mathbb{R}^{3\times1}, j \in \{0, 1, ..., n_i - 1\}\}_{i=1}^N$; Reference-view and source-view camera extrinsics $\mathbf{R}_0 \in \mathbb{R}^{3\times3}$, $\mathbf{t}_0 \in \mathbb{R}^{3\times1}$, $\{\mathbf{R}_i\}_{i=1}^N \in \mathbb{R}^{3\times3}$, $\{\mathbf{t}_i\}_{i=1}^N \in \mathbb{R}^{3\times1}$.

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 $S_i = 0$.

Referece-view camera center $\mathbf{c}_0 = -\mathbf{R}_0^T \mathbf{t}_0$;

for $i = 1$ **to** N **do**

  Source-view camera center $\mathbf{c}_i = -\mathbf{R}_i^T \mathbf{t}_i$;

  for $j = 0$ **to** $n_i - 1$ **do**

    $\theta_j = \frac{180°}{\pi} \arccos \frac{(\mathbf{c}_0 - \mathbf{p}_{ij}) \cdot (\mathbf{c}_i - \mathbf{p}_{ij})}{||\mathbf{c}_0 - \mathbf{p}_{ij}||_2 ||\mathbf{c}_i - \mathbf{p}_{ij}||_2}$

    if $\theta_j \leq \theta_0$ **then**

      $S_i = S_i + \exp(-\frac{(\theta_j - \theta_0^2)}{2\sigma_1^2})$

    else

      $S_i = S_i + \exp(-\frac{(\theta_j - \theta_0^2)}{2\sigma_2^2})$

    end

  end

end

return S_i

$$\mathbf{D}_{l,discrete} = \underbrace{\underset{d_{m,l} \in [d_{min,l}, d_{max,l}]}{\arg\max} \mathbf{P}_l(d_{m,l})}_{\text{discrete depth estimation}} \tag{3}$$

$$\mathbf{D}_{l,residual} = \underbrace{\underbrace{\frac{(d_{max,l} - d_{min,l})}{M_l}}_{\text{depth interval}} \underbrace{\max \mathbf{P}_l(d_{m,l})}_{\text{normalized depth residual}}}_{\text{depth residual}} \tag{4}$$

$$\mathbf{D}_{l,est} = \underbrace{\mathbf{D}_{l,discrete} + \mathbf{D}_{l,residual}}_{\text{continuous depth estimation}} \tag{5}$$

We take the depth estimation at the finest level ($l = 0$) as the final depth map for reference image $\mathbf{I}_0$.

Loss Function To obtain normalized depth residual, we adapt the generalized focal loss [25], [34] to supervise the difference between the predicted probability volume $\mathbf{P}_l$ and ground-truth probability volume $\mathbf{P}_{l,gt}$. We compute the $\mathbf{P}_{l,gt}$ as the normalized depth residual between the ground-truth

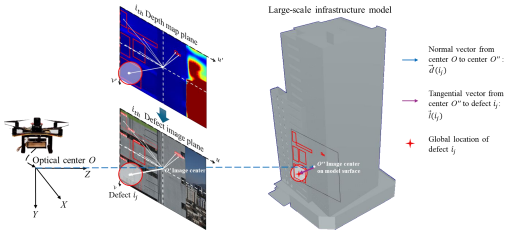


Fig. 4. Defect registration method for globally locating the individual defect.

depth and the discrete depth hypothesis. The total loss of the network $\mathcal{L}$ is the weighted sum of the per-level loss $\mathcal{L}_l$.

$$\mathcal{L} = \sum_{l=0}^{L} \lambda_l \mathcal{L}_l \tag{6}$$

and

$$\mathcal{L}_l = \sum_{\mathbf{x}\in\{\mathbf{x}_{valid}\}} -\beta_l |\mathbf{P}_{l,gt}(\mathbf{x}) - \mathbf{P}_l(\mathbf{x})|^{\gamma_l} \cdot ((1-\mathbf{P}_{l,gt}(\mathbf{x}))\log(1-\mathbf{P}_l(\mathbf{x})) + \mathbf{P}_{l,gt}(\mathbf{x})\log(\mathbf{P}_l(\mathbf{x}))) \tag{7}$$

where $\{\mathbf{x}_{valid}\}$ denotes the valid pixel set, β_l is the balancing factor, and γ_l is the focusing factor.

Depth Map Fusion With multi-view depth maps $\{\mathbf{D}_i\}_{i=0}^{N}$, we conduct depth map filtering by applying the probability threshold τ to remove depth outliers and enforcing the number of consistent views N_c to ensure depth consistency. We then fuse the filtered multi-view depth maps into the final point cloud by following the SOTA MVS methods for a fair comparison [24]–[26]. We follow the proposed MVS method above for infrastructure reconstruction from multi-view aerial images as shown in Fig. 1(c).

C. Register

Method Overview The core task of large-scale infrastructure inspection is to globally locate the detected defects. To achieve this goal, we present a GIS-based defect registration method to register detected defects onto the reconstructed model with high accuracy and efficiency.

Model Georeferencing Georeferencing refers to assigning real-world coordinates to an image (or model) based on its geographic location. As shown in Fig. 1(c), we first develop our WebGIS platform based on Cesium [35], a robust, scalable, and secure GIS platform for 3D geospatial data. We then import the reconstructed model and conduct model georeferencing to align the model with its real-world topology based on the geographic information. The aligned model serves as the physical entity for the localization of the detected defects.

Defect Localization We follow the geographic transformation paradigm shown in Fig. 4 to project the detected defects from the image coordinate onto the geo-referenced model. For j_{th} defect in the i_{th} image (denoted as defect i_j), we first capture the geographic coordinate of the image center O through real-time kinematic (RTK) positioning system. We then shift O to O'' along the depth dimension (blue vector $\vec{d}(i_j)$) to align with the infrastructure facade. Then, we transform the relative distance between O' and bounding box center of the defect i_j into the metric distance (i.e., violet vector $\vec{l}(i_j)$ from O'' to defect i_j). Finally, we obtain the global location of defect i_j (red star) by shifting geographic coordinate of point O'' along the tangential vector $\vec{l}(i_j)$. Following this paradigm, we automatically identify the global position of each detected defect. The defect category and appearance are also recorded to facilitate the maintenance measures. Especially, the defects that are repeatedly detected have been removed by geographic comparison.

TABLE I
QUANTITATIVE BENCHMARK RESULTS ON CUBIT-DET

Model	Params. (M) ↓	GFLOPs ↓	$mAP^{val}_{0.5}$ (%) ↑ / $mAP^{val}_{0.5:0.95}$ (%) ↑	Latency (ms) ↓
Faster R-CNN (Res50 [36]) [12]	42.62	477.24	71.5 / 43.3	76.9
PP-YOLO (Res50 [36]) [13]	48.99	136.43	76.4 / 45.1	14.5
PP-YOLOv2 (Res50 [36]) [14]	56.91	146.50	77.3 / 47.1	13.8
PP-YOLOE-s [15]	8.02	20.73	64.6 / 38.9	9.4
PP-YOLOE-m [15]	24.63	62.93	74.2 / 44.8	11.2
PP-YOLOE+-s [15]	8.02	20.73	70.6 / 44.0	8.1
PP-YOLOE+-m [15]	24.63	62.93	78.8 / 50.9	8.9
YOLOv5-n [17]	**1.76**	**4.10**	73.4 / 39.9	1.8
YOLOv5-s [17]	7.18	15.80	78.5 / 47.2	3.3
YOLOv5-m [17]	20.86	47.90	80.4 / 51.3	7.1
YOLOv7-t [18]	6.01	13.01	71.1 / 39.7	**1.9**
YOLOv7 [18]	36.49	61.94	77.5 / 47.8	8.4
YOLOX-n [16]	2.24	17.75	73.0 / 39.5	4.4
YOLOX-t [16]	5.03	39.00	75.3 / 49.2	5.8
YOLOX-s [16]	8.94	68.51	77.9 / 49.4	7.6
YOLOX-m [16]	25.30	73.80	78.2 / 52.2	13.7
YOLOv6-n (Baseline) [19]	4.63	29.03	76.3 / 47.9	2.2
YOLOv6-s [19]	18.50	115.64	79.0 / 48.2	5.3
YOLOv6-m [19]	37.90	225.55	**80.4 / 54.1**	9.8
CUBIT-Det (Ours)	4.14 (-0.49)	28.02 (-1.01)	77.5 (+1.2%) / 50.3 (+3.1%)	2.2

TABLE II
ABLATION EXPERIMENTS FOR PROPOSED CUBIT-NET

Method	Non-linear Act.	Conv.	Norm.	$mAP^{val}_{0.5}$ (%) ↑/ $mAP^{val}_{0.5:0.95}$ (%) ↑	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**

III. BENCHMARK AND ABLATION EXPERIMENTS

In this section, we demonstrate the efficacy of the proposed defect detection and MVS method on the corresponding datasets. **Please refer to the Appendix for more details of benchmark visualization, dataset, evaluation metric, implementation, and ablation study**. Due to the lack of public dataset, we validate the effectiveness of our registration strategy through real-world experiments.

A. Detect

Dataset and Evaluation Metrics We spilt proposed CUBIT-Det dataset into three parts: 3980 of images for training (72%), 442 for validation (8%), and the remaining 1105 (20%) for testing. Nine SOTA real-time object detection methods [12]–[19] have been trained and evaluated. For evaluation metrics, we adopt $\text{mAP}_{0.5}$ (%) from Pascal VOC and $\text{mAP}_{0.5:0.95}$ (%) from MS COCO to demonstrate the detection accuracy, and latency (ms) to show the detection speed.

Benchmark Results The benchmark results are shown in Table I. In comparison to SOTA methods [12]–[19], our method demonstrates higher detection accuracy in terms of $\text{mAP}_{0.5:0.95}$ and concurrently possesses competitive detection speed and lightweight parameters, suitable for onboard deployment of the UAV for real-time defect inspection. See benchmark visualization in Fig. 2(a) and Fig. 3 of the **Appendix**.

Ablation Study Table II shows the ablation experiments for CUBIT-Net. The introduction of the GIPFPP with GELU significantly improve detection accuracy with a minor loss of detection speed. Then, $Packet\_Conv$ reduces the model parameters by about 10% while maintaining the detection capability. Lastly, group normalization mitigates the sensitivity to batch size, further improving detection accuracy. After switching from the original module to GIPFPP module, the

TABLE III
QUANTITATIVE BENCHMARK RESULTS ON BLENDEDMVS VALIDATION SET FOR EVALUATING DEPTH ESTIMATION PERFORMANCE

Methods	EPE ↓	e_1 (%) ↓	e_3 (%) ↓
MVSNet [22]	1.49	21.98	8.32
CVP-MVSNet [37]	1.90	19.73	10.24
CDS-MVSNet [38]	1.80	22.88	9.28
Vis-MVSNet [31]	1.56	21.68	8.36
EPP-MVSNet [39]	1.17	12.66	6.20
UniMVSNet [25]	1.17	11.27	4.96
TransMVSNet [24]	1.05	13.74	5.47
CasMVSNet (Baseline) [23]	1.43	19.73	10.24
Ours	**1.02** (-0.41)	**10.15** (-9.58%)	**4.54** (-5.7%)

TABLE IV
QUANTITATIVE BENCHMARK RESULTS ON STANDARD MVS DATASETS FOR EVALUATING 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 [40]	2016	0.400	0.664	0.535	42.14	27.24
R-MVSNet [33]	2019	0.385	0.459	0.422	48.40	24.91
AttMVS [30]	2021	0.383	0.329	0.356	60.05	31.93
PatchMatchNet [41]	2021	0.427	0.277	0.352	53.15	32.31
AA-RMVSNet [32]	2021	0.376	0.339	0.357	61.51	33.53
EPP-MVSNet [39]	2021	0.413	0.296	0.355	61.68	35.72
CDS-MVSNet [38]	2022	0.365	0.281	0.323	61.58	-
NP-CVP-MVSNet [42]	2022	0.356	0.275	0.315	59.64	-
Vis-MVSNet [31]	2022	0.369	0.361	0.365	60.03	33.78
IterMVS [43]	2022	0.373	0.354	0.363	56.94	34.17
BH-RMVSNet [44]	2022	0.368	0.303	0.335	61.96	34.81
IS-MVSNet [45]	2022	0.351	0.359	0.355	62.82	34.87
TransMVSNet [24]	2022	0.360	0.271	0.316	**63.52**	37.00
DCS-MVSNet [46]	2023	0.316	0.372	0.344	53.48	-
IGEV-MVS [47]	2023	0.331	0.316	0.324	-	-
N2MVSNet [26]	2023	0.336	0.295	0.316	62.14	-
CostFormer [48]	2023	0.378	0.313	0.345	57.10	34.31
DispMVS [49]	2023	0.354	0.324	0.339	59.07	34.90
CasMVSNet [23] (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.

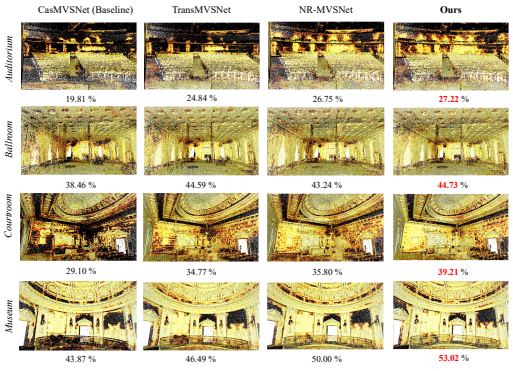


Fig. 5. The visualization of the reconstruction error (darker color indicates larger error) on complex indoor scenes of TNT dataset with varying illumination condition, surface texture, and depth range. In comparison to the SOTA methods [23], [24], [50], our method demonstrates higher reconstruction accuracy and completeness (number denotes F-score).

TABLE V
ABLATION EXPERIMENTS FOR PROPOSED MVS METHOD ($N = 5, W \times H = 1152 \times 864, \tau = 0.3$, 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**

$\text{mAP}_{0.5:0.95}$ of baseline model [19] is improved by 3%, while the number of parameters is reduced by 10%.

B. Reconstruct

Datasets and Evaluation Metrics DTU dataset consists of 124 objects where each object is captured from 49 viewpoints under 7 illumination conditions, with ground-truth point clouds for evaluation. Tanks and Temples (TNT) dataset contains realistic and challenging indoor and outdoor scenes with varying depth ranges and illumination conditions. BlendedMVS dataset comprises 113 scenes with over 17,000 high-resolution images with ground-truth depth maps. DTU and TNT datasets adopt the Overall Score (mm) and Mean F-score (%) to compute the summary measure of the reconstruction accuracy and completeness, respectively. BlendedMVS dataset adopts the end point error (EPE), 1-threshold error (e_1) and 3-threshold error (e_3) to evaluate depth estimation quality.

Benchmark Results For reconstruction, we train our MVS network on the DTU training set and evaluate it on the DTU evaluation set (22 scenes) by following the standard evaluation protocol. We further finetune the trained model on the BlendedMVS training set and benchmark the large-scale reconstruction performance on TNT dataset (14 scenes). The benchmark results on the DTU and TNT dataset in Table IV demonstrate the SOTA reconstruction performance of our method. The Fig. 5 presents the rendered reconstruction error on complex scenes of TNT dataset with varying depth ranges, our method produces less error in comparison to SOTA methods. For depth estimation, we benchmark our method on BlendedMVS validation set in Table III, where our method predicts more accurate and complete depth maps.

Ablation Study We conduct ablation study on DTU evaluation set to validate the effectiveness of each component of our proposed MVS method. We choose CasMVSNet [23] as our baseline method consisting of FPN for feature extaction, heuristic-based method for cost volume aggregation and regression-based depth estimation. The ablation experiments in Table V demonstrate that the BPA, Sparse ACVA, and continuous depth estimation strategy successively improve the overall reconstruction performance to the state of the art. We analyze the efficiency of the Sparse ACVA (33.73% memory footprint reduction and 15.65% runtime improvement) in the Table VIII of the **Appendix**.

C. Register

We establish the GIS-based virtual environment [35] to demonstrate global localization of defects. The reconstructed infrastructure model is imported into this virtual space and geo-referenced, simultaneously. The detected defects, represented as red bounding boxes in Fig. 6(a), are then respectively registered onto the reconstructed model and marked as green symbols in Fig. 6(b).

Localization Accuracy We compute the defect localization error as the offset between the registered defect location and the center of the bounding box of the detected defect. To achieve this goal, we restore each viewpoint of onboard camera within the established virtual environment with same geographical and optical parameters (Fig. 6(b)). We then mask the defect image over the virtual one to compute the defect localization error and transform the error in pixel to physical metric in cm. We compute mean absolute error (MAE), root mean square error (RMSE), and interquartile range (IQR) which is the difference between the 1st quartile (Q1) and the 3rd quartile (Q3), as shown in Table VI, verifying the effectiveness of our registration method.

TABLE VI
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

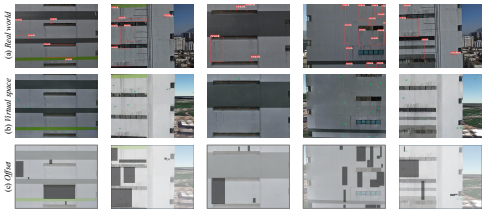


Fig. 6. The visualization of the defect registration results. (a) demonstrates accurate defect detection results (red rectangle boxes) from our proposed CUBIT-Net for real-world warehouse inspection. (b) shows the corresponding registered defects (green symbols) in GIS-based virtual environment. (c) represents the offset between the results in (a) and (b).

IV. REAL-WORLD EXPERIMENTS

We deploy our framework on various large-scale scenarios to verify its effectiveness and efficiency. Here, we take a large-scale high-rise warehouse ($36m \times 27m \times 100m$) as a representative instance. **Refer to Appendix for more details.**

Effectiveness (1) ***Detect***: We collect 923 close-range facade images (1152×832) for defect (crack, spalling, moisture) detection and our CUBIT-Net achieves 82% $\text{mAP}_{0.5}$ detection accuracy. As shown in Fig. 6(a), our method can accurately detect multi-scale cracks. (2) ***Reconstruct***: We collect 826 multi-view aerial images (1152×832) for infrastructure reconstruction and deploy our proposed MVS method to achieve dense and complete infrastructure reconstruction. Our method has significantly outperformed (see Table I of the **Appendix**) the industrial reconstruction solutions used in the existing inspection systems [2], [20], [21]. We conduct more real-world experiments to demonstrate the superiority of our proposed MVS method as shown in Fig. 7. For scene *Warehouse*, *Tulou*, *Campus* and *Library* with different scene scale, depth range, surface texture, and brightness level, our method outperforms the SOTA methods [23], [24], [43] by achieving more accurate and complete depth estimation, especially in scene *Library* with specular reflection and low-textured surface. (3) ***Register***: We adopt proposed GIS-based registration method for defect global localization. The localization accuracy guaranteed by RTK (horizontal and vertical positioning error is $\pm 2cm$ and $\pm 4cm$, respectively) and our effective depth estimation strategy has reached cm-level as in Table VI.

Efficiency and Scalability (1) ***Detect***: Our CUBIT-Net achieves a detection speed of 22.7 FPS on the edge detector NVIDIA Jetson Orin NX, which is well-suited for real-time detection tasks. We first convert our model to the ONNX format, then generate a TensorRT inference engine, and finally utilize Triton for deploying the model on the edge detector. (2) ***Reconstruct***: Our MVS method (44.928 mins: 24.378 mins for SfM, 6.569 mins for view selection, and 13.981 mins for MVS) is 8.88 times faster than industrial solution (DJI Terra, 399 mins) on the 3090Ti GPU. (3) ***Register***: The computation of the global defect locations takes 16.426 ms on the i9-10920X CPU (excluding the uploading time of the reconstruction model and defect images to the GIS environment). Besides, the proposed framework can extend from small-scale to large-scale scenes benefiting from our MVS network (see Fig. 4 and Table II of the **Appendix**).

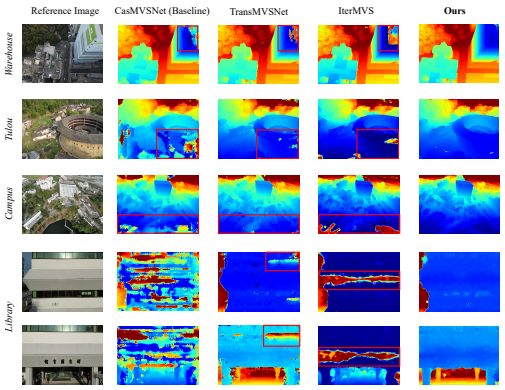


Fig. 7. The depth map estimation on the complex outdoor scenes with varying surface texture, illumination condition, and depth range. Our method outperforms the SOTA methods [23], [24], [43] from depth estimation quality, especially in scene *Library* with specular reflection and low-textured surface.

V. CONCLUSION

We have presented a ***Detect-Reconstruct-Register*** framework towards automated large-scale infrastructure inspection. We have constructed a large-scale high-resolution defect dataset and proposed a tailored effective defect detection algorithm. We have also presented a learning-based MVS network for large-scale infrastructure reconstruction and a GIS-based registration method for defect global localization. Extensive experiments on benchmark datasets and real-world scenarios cross-validate the effectiveness, efficiency, and scalability of the proposed framework.

REFERENCES

[1] L. Yang, B. Li, G. Yang, Y. Chang, Z. Liu, B. Jiang, and J. Xiaol, "Deep neural network based visual inspection with 3d metric measurement of concrete defects using wall-climbing robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2849–2854.

[2] 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 Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7512–7524, 2022.

[3] L. Deng, T. Sun, L. Yang, and R. Cao, "Binocular video-based 3d reconstruction and length quantification of cracks in concrete structures," *Automation in Construction*, vol. 148, p. 104743, 2023.

[4] Y.-F. Liu, X. Nie, J.-S. Fan, and X.-G. Liu, "Image-based crack assessment of bridge piers using unmanned aerial vehicles and three-dimensional scene reconstruction," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 5, pp. 511–529, 2020.

[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 *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 2439–2446.

[6] T. Dias and M. Basiri, "Bogiecopter: A multi-modal aerial-ground vehicle for long-endurance inspection applications," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3303–3309.

[7] H. Ahmed, S. T. Nguyen, D. La, C. P. Le, and H. M. La, "Multi-directional bicycle robot for bridge inspection with steel defect detection system," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 4617–4624.

[8] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "Simplenet: A simple network for image anomaly detection and localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20 402–20 411.

[9] A. Agarwal, A. Ajith, C. Wen, V. Stryzheus, B. Miller, M. Chen, M. K. Johnson, J. L. Susa Rincon, J. Rosca, and W. Yuan, "Robotic defect inspection with visual and tactile perception for large-scale components," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 110–10 116.

[10] J. Yu, H. Oh, S. Fichera, P. Paoletti, and S. Luo, "Multi-source domain adaptation for unsupervised road defect segmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5638–5644.

[11] G. Yang, K. Liu, J. Zhang, B. Zhao, Z. Zhao, X. Chen, and B. M. Chen, "Datasets and processing methods for boosting visual inspection of civil infrastructure: A comprehensive review and algorithm comparison for crack classification, segmentation, and detection," *Construction and Building Materials*, vol. 356, p. 129226, 2022.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[13] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding *et al.*, "Pp-yolo: An effective and efficient implementation of object detector," *arXiv preprint arXiv:2007.12099*, 2020.

[14] X. Huang, X. Wang, W. Lv, X. Bai, X. Long, K. Deng, Q. Dang, S. Han, Q. Liu, X. Hu *et al.*, "Pp-yolov2: A practical object detector," *arXiv preprint arXiv:2104.10419*, 2021.

[15] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du *et al.*, "Pp-yoloe: An evolved version of yolo," *arXiv preprint arXiv:2203.16250*, 2022.

[16] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021.

[17] G. Jocher *et al.*, "Ultralytics yolov5," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[18] C. Wang, A. Bochkovskiy, and H. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[19] C. Li, L. Li, Y. Geng, H. Jiang, M. Cheng, B. Zhang, Z. Ke, X. Xu, and X. Chu, "Yolov6 v3. 0: A full-scale reloading," *arXiv preprint arXiv:2301.05586*, 2023.

[20] 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 Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5415–5424, 2021.

[21] K. Liu and B. M. Chen, "Industrial uav-based unsupervised domain adaptive crack recognitions: From database towards real-site infrastructural inspections," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9410–9420, 2023.

[22] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "Mvsnet: Depth inference for unstructured multi-view stereo," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[23] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2495–2504.

[24] Y. Ding, W. Yuan, Q. Zhu, H. Zhang, X. Liu, Y. Wang, and X. Liu, "Transmvsnet: Global context-aware multi-view stereo network with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8585–8594.

[25] R. Peng, R. Wang, Z. Wang, Y. Lai, and R. Wang, "Rethinking depth estimation for multi-view stereo: A unified representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8645–8654.

[26] Z. Zhang, H. Gao, Y. Hu, and R. Wang, "N2mvsnet: Non-local neighbors aware multi-view stereo network," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[27] C. Gao, W. Ding, Z. Zhao, and B. M. Chen, "Energy-optimal trajectory-based traveling salesman problem for multi-rotors unmanned aerial vehicle," *The 62nd IEEE Conference on Decision and Control*, 2023.

[28] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[29] Q. Xu and W. Tao, "Learning inverse depth regression for multi-view stereo with correlation cost volume," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, 2020, pp. 12 508–12 515.

[30] K. Luo, T. Guan, L. Ju, Y. Wang, Z. Chen, and Y. Luo, "Attention-aware multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] J. Zhang, S. Li, Z. Luo, T. Fang, and Y. Yao, "Vis-mvsnet: Visibility-aware multi-view stereo network," *International Journal of Computer Vision*, pp. 1–16, 2022.

[32] Z. Wei, Q. Zhu, C. Min, Y. Chen, and G. Wang, "Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 6187–6196.

[33] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang, and L. Quan, "Recurrent mvsnet for high-resolution multi-view stereo depth inference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[34] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 002–21 012, 2020.

[35] I. Cesium GS, "Cesium, the platform for 3d geospatial," https://www.cesium.com/.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37] J. Yang, W. Mao, J. M. Alvarez, and M. Liu, "Cost volume pyramid based depth inference for multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4877–4886.

[38] K. T. Giang, S. Song, and S. Jo, "Curvature-guided dynamic scale networks for multi-view stereo," in *International Conference on Learning Representations*, 2022.

[39] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 5732–5740.

[40] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[41] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "Patchmatchnet: Learned multi-view patchmatch stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 194–14 203.

[42] J. Yang, J. M. Alvarez, and M. Liu, "Non-parametric depth distribution modelling based depth inference for multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8626–8634.

[43] F. Wang, S. Galliani, C. Vogel, and M. Pollefeys, "Itermvs: Iterative probability estimation for efficient multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8606–8615.

[44] Z. Wei, Q. Zhu, C. Min, Y. Chen, and G. Wang, "Bidirectional hybrid lstm based recurrent neural network for multi-view stereo," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2022.

[45] L. Wang, Y. Gong, X. Ma, Q. Wang, K. Zhou, and L. Chen, "Is-mvsnet: importance sampling-based mvsnet," in *European Conference on Computer Vision*. Springer, 2022, pp. 668–683.

[46] S. Zhang, Z. Wei, W. Xu, L. Zhang, Y. Wang, X. Zhou, and J. Liu, "Dsc-mvsnet: attention aware cost volume regularization based on depthwise separable convolution for multi-view stereo," *Complex & Intelligent Systems*, pp. 1–17, 2023.

[47] G. Xu, X. Wang, X. Ding, and X. Yang, "Iterative geometry encoding volume for stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 919–21 928.

[48] W. Chen, H. Xu, Z. Zhou, Y. Liu, B. Sun, W. Kang, and X. Xie, "Costformer: Cost transformer for cost aggregation in multi-view stereo," *International Joint Conferences on Artificial Intelligence*, 2023.

[49] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 3, 2023, pp. 3091–3099.

[50] J. Li, Z. Lu, Y. Wang, J. Xiao, and Y. Wang, "Nr-mvsnet: Learning multi-view stereo based on normal consistency and depth refinement," *IEEE Transactions on Image Processing*, 2023.