Multi-View Stereo with Geometric Encoding for Dense Scene Reconstruction

Guidong Yang, Rui Cao, Junjie Wen, Benyun Zhao, Qingxiang Li,
Yijun Huang, Xi Chen, Alan Lam, Yun-Hui Liu, and Ben M. Chen*

***Abstract*—Multi-view stereo (MVS) implicitly encodes photometric and geometric cues into the cost volume for multi-view correspondence matching, transferring insufficient geometric cues essential to depth estimation and reconstruction. This paper proposes GE-MVS, a novel multi-view stereo network with geometric encoding for more accurate and complete depth estimation and point cloud reconstruction. First, the cross-view adaptive cost volume aggregation module is proposed to strengthen multi-view geometric cues encoding during cost volume construction. Then, the depth consistency optimization is performed in the 3D point space during learning by invoking ground-truth depth cues from adjacent views. Finally, the surface normal geometries are explicitly encoded to refine the sampled depth hypotheses to be consistent in the local neighbor regions. Extensive experiments on the standard MVS benchmarks including DTU, Tanks and Temples, and BlendedMVS demonstrate the state-of-the-art depth estimation and point cloud reconstruction performance of GE-MVS. The GE-MVS is further deployed in real-world experiments for UAV-based large-scale reconstruction, where our method outperforms the prevalent industrial reconstruction solutions concerning reconstruction efficiency and efficacy.**

I. Introduction and Related Work

Multi-view stereo (MVS) reconstructs dense point cloud representation of the scene from an unordered set of multi-view calibrated images by solving multi-view correspondence matching and has been widely adopted in various tasks such as robotic manipulation [1], aerial path planning [2], autonomous driving [3], and underwater exploration [4], etc. Although traditional MVS methods [5]–[9] have achieved decent reconstruction performance based on hand-crafted matching metrics, recent learning-based MVS methods [10]–[12] significantly outperform their traditional counterparts in terms of reconstruction accuracy and completeness on standard MVS benchmarks [13]–[15]. Learning-based MVS methods first adopt deep networks to extract multi-view feature maps. Then, multi-view feature maps and associated camera parameters are implicitly encoded into the cost volume for enhanced multi-view correspondence matching. Afterward, cost volume is regularized to estimate the depth map. Multi-view depth maps are then filtered and fused to the dense point cloud by applying photometric and geometric constraints as a post-processing step. Despite the promising results exhibited by learning-based MVS methods, the following improvements can be made to further improve the reconstruction performance:

* Corresponding Author.

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.

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

1^{st} **Motivation** MVS with varying viewpoints encounters with occlusion, illumination changes, and content variations, where occlusion leads to inaccurate and incomplete depth estimation and illumination changes make the depth estimation and subsequent point cloud reconstruction of non-Lambertian surfaces more challenging. We observe that the source view more adjacent to the reference view tends to have higher feature overlap with respect to the far one and hence offers more reliable photometric and geometric cues for depth estimation and reconstruction [16]–[19]. Based on this observation, we hence propose the cross-view adaptive cost volume aggregation module to strengthen geometric cues encoding by inferring per-view per-pixel visibility weight for pairwise matching cost between the reference and each source view. Experimental results show that the proposed module effectively improves both reconstruction accuracy and completeness by alleviating the above issues.

2^{nd} **Motivation** Most existing methods [17]–[24] perform depth inconsistency check for estimated depth maps by applying photometric and geometric constraints as a post-processing step, where inconsistent pixels are directly discarded during point cloud generation resulting in incomplete reconstruction. The differentiable homography warping exclusively performs implicit geometric modeling during network learning, which delivers insufficient geometric cues essential to depth estimation and subsequent point cloud reconstruction. Different from existing methods, we perform explicit depth inconsistency check during learning by encoding adjacent source-view ground-truth depth cues to geometrically constrain the depth optimization process directly from the 3D point space. Experimental results show that explicit depth consistency optimization endows the network with the ability for more accurate and complete reconstruction.

3^{rd} **Motivation** Most existing methods [17]–[23] adopt coarse-to-fine framework [12] to gradually refine depth hypotheses of each feature level for memory efficiency and high-resolution reconstruction, where coarse-level depth hypotheses are uniformly sampled from predefined depth range and finer-level depth hypotheses are dynamically obtained from coarser-level depth estimation. However, resulting depth hypotheses are less satisfactory as the coarser level suffers from inaccurate and incomplete depth estimation, which imposes learning ambiguity to the cross-entropy loss where ground-truth probability volume is obtained by one-hot encoding the depth hypotheses closest to the ground-truth depths. Inspired by [25]–[28], we propose to refine and constrain the sampled depth hypotheses to be geometrically consistent in local neighbor regions by explicitly encoding surface normal

geometries. Experimental results show that the normal-assisted depth hypotheses refinement effectively enhances overall reconstruction performance.

We formulate a coarse-to-fine MVS network with the above geometric encoding strategies to demonstrate the superiority of the proposed modules, termed GE-MVS. Extensive experiments show that our method achieves state-of-the-art depth estimation and point cloud reconstruction performance on standard MVS benchmarks including DTU [13], Tanks and Temples [15], and BlendedMVS [14]. The real-world experiments for large-scale reconstruction based on unmanned aerial vehicle (UAV) are conducted to demonstrate the scalability and generalization ability of GE-MVS, where our method significantly outperforms industrial reconstruction solutions concerning reconstruction efficiency and efficacy.

II. Methodology

The architecture of GE-MVS is shown in Fig. 1. Given N-view images $\{\mathbf{I}_i \in \mathbb{R}^{3\times H\times W}\}_{i=0}^{N-1}$ with their camera intrinsics $\{\mathbf{K}_i \in \mathbb{R}^{3\times 3}\}_{i=0}^{N-1}$ and extrinsics $\{[\mathbf{R}_i \in \mathbb{R}^{3\times 3}; \mathbf{t}_i \in \mathbb{R}^{3\times 1}]\}_{i=0}^{N-1}$, our goal is to estimate the depth map $\mathbf{D}_{0,\text{est}} \in \mathbb{R}^{H\times W}$ for the reference image $\mathbf{I}_0$. First, multi-scale feature pyramids of multi-view input images are extracted through a feature extractor with shared weights among multiple views. The initial 3D cost volume pyramid of the reference view is constructed via differentiable homography warping and cross-view adaptive aggregation to measure multi-view matching similarity (Subsection II-A). Then, the noise-contaminated cost volume pyramid is regularized to acquire the probability volume pyramid for depth map pyramid estimation (Subsection II-B). Afterward, the depth inconsistency of reference-view depth map pyramid is explicitly checked by encoding adjacent source-view ground-truth depth to geometrically constrain the depth optimization from the 3D point space (Subsection II-C). Finally, we refine the sampled depth hypotheses of the coarse-to-fine framework by explicitly encoding surface normal geometries (Subsection II-D). The depth map at the original scale is taken as the final output. The estimated depth maps are filtered and fused [29] to the final point cloud.

A. Cross-View Adaptive Cost Volume Aggregation

Feature Volume Construction Given multi-view images $\{\mathbf{I}_i\}_{i=0}^{N-1}$, the Feature Pyramid Network [30] is adopted to extract L-scale feature pyramids $\{\mathbf{F}_i^l\}_{i=0}^{N-1} \in \mathbb{R}^{C_l\times \frac{H}{2^l}\times \frac{W}{2^l}}$, where $l \in \{0, 1, \ldots, L-1\}$ denotes the feature level and C_l denotes the channel number. For each feature level l, the reference-view depth range $[\mathbf{D}_{\text{min}}^l, \mathbf{D}_{\text{max}}^l]$ is uniformly discretized into M_l discrete depth hypotheses:

$$\mathbf{D}_{\text{ini},m}^l = \mathbf{D}_{\text{min}}^l + m\left(\frac{\mathbf{D}_{\text{max}}^l - \mathbf{D}_{\text{min}}^l}{M_l - 1}\right), \tag{1}$$

where $\{\mathbf{D}_{\text{max}}^l, \mathbf{D}_{\text{min}}^l, \mathbf{D}_{\text{ini},m}^l\} \in \mathbb{R}^{\frac{H}{2^l}\times \frac{W}{2^l}}$ denote the maximum, minimum, and sampled initial depth hypotheses at feature level l, respectively. Here, $m \in \{0, 1, \ldots, M_l - 1\}$ denotes the index of depth hypothesis plane. Note that the coarse-level depth range is predefined and the finer-level depth range is obtained from the coarser-level depth estimation. $\mathbf{D}_{\text{ini},m}^l$ is further refined by surface normal geometries to final depth hypotheses $\mathbf{D}_m^l$ for more accurate and complete depth estimation and reconstruction, as illustrated in Subsection II-D.

The pairwise pixel coordinate mapping between the reference-view feature map $\mathbf{F}_0^l$ and adjacent source-view feature maps $\{\mathbf{F}_i^l\}_{i=1}^{N-1}$ at depth $\mathbf{D}_m^l$ is then established via differentiable homography:

$$\mathbf{p}_i = \mathbf{K}_i\left[\mathbf{R}_{0\to i}\left(\mathbf{K}_0^{-1}\mathbf{p}_0\mathbf{D}_m^l(\mathbf{p}_0)\right) + \mathbf{t}_{0\to i}\right], \tag{2}$$

where $\mathbf{p}_0$ and $\mathbf{p}_i$ denote the reference-view and source-view pixel coordinates, respectively. $\mathbf{R}_{0\to i} = \mathbf{R}_i\mathbf{R}_0^{-1}$ and $\mathbf{t}_{0\to i} = (\mathbf{t}_0 - \mathbf{R}_i\mathbf{R}_0^{-1}\mathbf{t}_i)$ are the relative rotation matrix and translation vector between the reference and source view, respectively. $\mathbf{K}_0$ and $\mathbf{K}_i$ are the scaled camera intrinsics for the reference and source view. Given $\mathbf{p}_i$ from $\mathbf{F}_i^l$, differentiable bilinear interpolation is adopted to interpolate the source-view feature map $\widetilde{\mathbf{F}}_i^l$ aligned to the reference view. The above coordinate mapping and interpolation process are performed for each depth hypothesis $\mathbf{D}_m^l(\mathbf{p}_0)$ to obtain the corresponding feature map $\widetilde{\mathbf{F}}_{i,\mathbf{D}_m^l(\mathbf{p}_0)}^l$, which is consecutively stacked along the depth dimension to construct the source-view feature volume $\{\mathbf{V}_i^l \in \mathbb{R}^{C_l\times M_l\times \frac{H}{2^l}\times \frac{W}{2^l}}\}_{i=1}^{N-1}$. The reference-view feature map $\mathbf{F}_0^l$ is repeated M_l times along the depth dimension to obtain the reference-view feature volume $\mathbf{V}_0^l \in \mathbb{R}^{C_l\times M_l\times \frac{H}{2^l}\times \frac{W}{2^l}}$.

Cost Volume Aggregation Given per-view feature volumes, the next step is to aggregate multi-view feature volumes into the cost volume to measure multi-view feature matching similarity. The cross-view adaptive cost volume aggregation module is proposed to strengthen the geometric cues implicitly encoded by the homography warping and remove multi-view matching ambiguities by considering per-view per-pixel visibility. The schematic plot of the cross-view adaptive aggregation module is shown in the green box of Fig. 1. The initial pairwise matching cost between the reference view and i_{th} source view at pixel $\mathbf{p}$ is measured as:

$$\mathbf{S}_{0\leftrightarrow i}^l(\mathbf{p}) = \frac{1}{C_l}\sum_{i=0}^{C_l-1}\left(\mathbf{V}_0^l(\mathbf{p}) - \mathbf{V}_i^l(\mathbf{p})\right)^2, \tag{3}$$

where we first compute the pairwise feature similarity and then compute mean over the channel dimension for memory efficiency. $\{\mathbf{S}_{0\leftrightarrow i}^l \in \mathbb{R}^{M_l\times \frac{H}{2^l}\times \frac{W}{2^l}}\}_{i=1}^{N-1}$ denotes the initial pairwise cost, which is smoothed via a lightweight network to produce the per-voxel weight $\{\mathbf{W}_{0\leftrightarrow i}^l \in \mathbb{R}^{M_l\times \frac{H}{2^l}\times \frac{W}{2^l}}\}_{i=1}^{N-1}$. Multi-view feature volumes are then adaptively aggregated as:

$$\mathbf{C}^l(\mathbf{p}) = \frac{1}{N-1}\sum_{i=1}^{N-1}\underbrace{\left(1 + \max_d \mathbf{W}_{0\leftrightarrow i}^l(\mathbf{p})\right)}_{\text{Adaptive Visibility}} \odot \mathbf{S}_{0\leftrightarrow i}^l(\mathbf{p}), \tag{4}$$

where $\mathbf{C}^l \in \mathbb{R}^{M_l\times \frac{H}{2^l}\times \frac{W}{2^l}}$ denotes the reference-view cost volume and $\odot$ is the Hadamard product. The per-view per-pixel adaptive visibility is obtained by taking the maximum similarity along the depth dimension and varying spatial saliency along the height and width dimension. The 1 is added to the adaptive visibility to preserve the initial pairwise cost and prevent over-smoothness. The per-pixel adaptive visibility

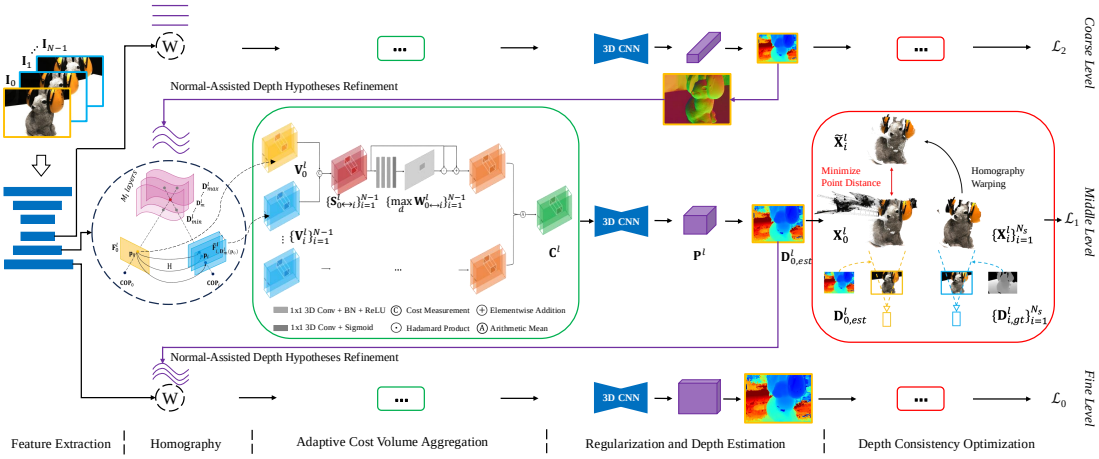


Fig. 1. Network overview of GE-MVS. L is set to 3 to form a three-stage coarse-to-fine network. The green box denotes the proposed cross-view adaptive cost volume aggregation module, the red box represents the proposed depth consistency optimization module, and the violet line indicates the proposed normal-assisted depth hypotheses refinement module. The rest is inherited from our baseline method [12].

is masked over the initial pairwise cost and the resulting pairwise cost is accumulated over all the source views and averaged to obtain the final cost volume. In this way, the pixels with higher feature similarity will have a larger contribution, while pixels that suffer from matching ambiguities will be suppressed during cost volume aggregation.

B. Cost Volume Regularization and Depth Estimation

Following the previous work [12], [17], [18], a multi-scale 3D U-Net is utilized to regularize the initial noise-contaminated cost volume $\mathbf{C}^l \in \mathbb{R}^{M_l \times \frac{H}{2^l} \times \frac{W}{2^l}}$ and transform multi-view matching cost into the probability volume $\mathbf{P}^l \in \mathbb{R}^{M_l \times \frac{H}{2^l} \times \frac{W}{2^l}}$ via softmax operation along the depth dimension. The probability volume represents the probability map corresponding to M_l depth hypotheses. The depth estimation is treated as a pixel-wise depth classification problem where the depth hypothesis corresponds to the maximum probability is taken as the depth estimation result:

$$\mathbf{D}^l_{0,\text{est}}(\mathbf{p}) = \underset{d \in \{\mathbf{D}^l_m(\mathbf{p}_0)\}^{M_l-1}_{m=0}}{\arg\max} \mathbf{P}^l_0(\mathbf{p}), \tag{5}$$

where $\mathbf{D}^l_{0,\text{est}} \in \mathbb{R}^{\frac{H}{2^l} \times \frac{W}{2^l}}$ denotes the reference-view depth estimation for feature level l. Note that the depth estimation at the fine level is taken as the final output.

C. Depth Consistency Optimization

Most existing methods [17]–[24] perform depth inconsistency check after network learning and directly discard the inconsistent pixels during point cloud fusion, leading to incomplete reconstruction. Besides, the cost volume delivers insufficient geometric cues for depth estimation and subsequent reconstruction. To dynamically improve depth consistency and explicitly strengthen the geometric modeling, we perform depth consistency optimization in the 3D point space by encoding ground-truth depth cues from adjacent views to suppress the inconsistent pixels during learning .

2D→3D Backward Projection At each feature level l, we first back-project the pixel coordinates from the reference view and source views to the 3D point space, using the reference-view depth estimation and the source-view ground-truth depth, respectively. The backward projection is formulated as:

$$\mathbf{X}^l_i(\mathbf{p}) = (\mathbf{K}_i\mathbf{R}_i)^{-1}\mathbf{p}\mathbf{D}^l_i(\mathbf{p}) - \mathbf{R}_i^{-1}\mathbf{t}_i, \tag{6}$$

where $\{\mathbf{X}^l_i \in \mathbb{R}^{3\times\frac{H}{2^l}\times\frac{W}{2^l}}\}^{N_s}_{i=0}$ denotes the back-projected point coordinates in the world space. Here, $\mathbf{X}^l_0$ and $\{\mathbf{X}^l_i\}^{N_s}_{i=1}$ represent the point coordinates for the reference view and N_s adjacent source views involved in the depth consistency optimization, respectively. The sets $\{\mathbf{K}_i\}^{N_s}_{i=0}$ and $\{[\mathbf{R}_i|\mathbf{t}_i]\}^{N_s}_{i=0}$ denote the scaled camera intrinsics and extrinsics, respectively. Additionally, $\{\mathbf{D}^l_i(\mathbf{p})\}^{N_s}_{i=0}$ represents the depth at pixel $\mathbf{p}$, where $\mathbf{D}^l_0(\mathbf{p}) = \mathbf{D}^l_{0,\text{est}}(\mathbf{p})$ is the depth estimation of the reference view, and $\{\mathbf{D}^l_i(\mathbf{p}) = \mathbf{D}^l_{i,\text{gt}}(\mathbf{p})\}^{N_s}_{i=1}$ are the ground-truth depths from the N_s adjacent source views.

Point Cloud Alignment As shown in red box of Fig. 1, after backward projection, the reference-view point cloud $\mathbf{X}^l_0$ is partially corrupted with noisy points due to inaccurate and incomplete depth estimation $\mathbf{D}^l_{0,\text{est}}$, while the source-view point cloud $\mathbf{X}^l_i$ is complete and clean benefiting from ground-truth depth map $\mathbf{D}^l_{i,\text{gt}}$. Furthermore, there also exists misalignment between $\mathbf{X}^l_0$ and $\mathbf{X}^l_i$ due to cumulative errors in camera intrinsics and extrinsics. We hence establish pairwise coordinate mapping to align the source-view point cloud to the reference view. The coordinate mapping is formulated as:

$$\mathbf{p}_i = \mathbf{K}_i[\mathbf{R}_{0\to i}(\mathbf{K}_0^{-1}\mathbf{p}_0\mathbf{D}^l_{0,\text{est}}(\mathbf{p}_0)) + \mathbf{t}_{0\to i}], \tag{7}$$

where $\mathbf{D}^l_{0,\text{est}}(\mathbf{p}_0)$ denotes reference-view depth estimation at pixel $\mathbf{p}_0$. Given source-view pixel coordinates $\mathbf{p}_i$ from $\mathbf{X}^l_i$, we regard x, y, z point coordinates as channel features and utilize the differentiable bilinear interpolation to obtain the source-view point cloud $\widetilde{\mathbf{X}}^l_i$ aligned to the reference view.

Depth Inconsistency Check The pointwise distance error is then computed as the Euclidean norm between reference-view points and aligned source-view points:

$$\mathbf{E}^l_{0\leftrightarrow i}(\mathbf{p}) = \|\mathbf{X}^l_0(\mathbf{p}) - \widetilde{\mathbf{X}}^l_i(\mathbf{p})\|_2, \tag{8}$$

where $\mathbf{E}^l_{0\leftrightarrow i} \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}}$ denotes error map, $\|\cdot\|_2$ is the L_2-norm. If the pointwise distance error at pixel $\mathbf{p}$ exceeds a certain threshold, then the reference-view depth estimation $\mathbf{D}^l_{0,\text{est}}(\mathbf{p})$ is considered as inaccurate. For each reference view, the 2D→3D backward projection is performed for N_s source views, and the depth inconsistency of the reference view with respect to all source views is accumulated and averaged as:

$$\mathbf{M}^l_0(\mathbf{p}) = \frac{1}{N_s}\sum_{i=1}^{N_s}[\mathbf{E}^l_{0\leftrightarrow i}(\mathbf{p}) > \epsilon_l], \tag{9}$$

where $\mathbf{M}^l_0 \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}}$ denotes the depth inconsistency mask of the reference view at feature level l. $[\cdot]$ denotes the Iverson bracket and ϵ_l denotes the per-level point distance threshold.

Loss Function As illustrated in Subsection II-B, the depth estimation is treated as a pixel-wise depth classification problem, where the depth hypothesis corresponds to the maximum probability is taken as the estimation result. Therefore, the cross-entropy loss is adopted to supervise on the difference between estimated probability volume $\mathbf{P}^l(\mathbf{p})$ and ground-truth probability volume $\mathbf{P}^l_{\text{gt}}(\mathbf{p})$, where $\mathbf{P}^l_{\text{gt}}(\mathbf{p})$ is obtained by one-hot encoding the depth hypotheses closest to the ground-truth depths. The cross-entropy loss of feature level l is defined as follows:

$$\mathcal{L}^l_{CE} = \sum_{\mathbf{p}\in\{\mathbf{p}_v\}}\sum_{m=0}^{M_l-1} -\mathbf{P}^l_{\text{gt},m}(\mathbf{p})\log(\mathbf{P}^l_m(\mathbf{p})), \tag{10}$$

where $\{\mathbf{P}^l_{\text{gt},m}, \mathbf{P}^l_m\} \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}}$ are ground-truth and estimated probability map of m_{th} depth hypothesis. $\{\mathbf{p}_v\}$ is the set of pixels with valid ground-truth depth.

The per-level cross-entropy loss $\mathcal{L}^l_{CE}$ is further weighted by the depth inconsistency mask $\mathbf{M}^l_0(\mathbf{p})$ to geometrically optimize depth consistency:

$$\mathcal{L}_l = \mathcal{L}^l_{CE} + \sum_{\mathbf{p}\in\{\mathbf{p}_v\}}\sum_{m=0}^{M_l-1} -\mathbf{M}^l_0(\mathbf{p})\left(\mathbf{P}^l_{\text{gt},m}(\mathbf{p})\log(\mathbf{P}^l_m(\mathbf{p}))\right), \tag{11}$$

where $\mathcal{L}_l$ denotes the per-level loss for depth optimization, $\mathbf{M}^l_0(\mathbf{p})$ provides the per-pixel penalty for reference-view depth inconsistent to N_s source views, the original cross-entropy loss is retained to prevent excessive depth consistency correction. The total loss for optimization is the weighted sum of the per-level loss:

$$\mathcal{L} = \sum_{l=0}^{L-1}\lambda_l\mathcal{L}_l, \tag{12}$$

where $\mathcal{L}$ represents the total loss for depth optimization, L denotes the total number of feature levels, and λ_l is the loss weight of level l.

D. Normal-Assisted Depth Hypotheses Refinement

Coarse-to-fine depth hypotheses sampling imposes learning ambiguity on network learning. Besides, sampled depth hypotheses lead to inconsistent depth estimation in the local neighbor regions under challenging multi-view matching conditions. To alleviate these issues, we explicitly encode surface normal geometries [25]–[28] to refine the depth hypotheses to be geometrically smooth and consistent in the local neighbor regions. The normal map is generated by the monocular normal estimation network Omnidata [31] to complement multi-view matching ambiguities under challenging conditions.

Given reference-view normal map $\mathbf{N} \in \mathbb{R}^{3\times H\times W}$ and m_{th} initial depth hypotheses $\mathbf{D}^l_{\text{ini},m} \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}}$ from the coarse-to-fine framework, we interpolate the $\mathbf{D}^l_{\text{ini},m}$ to the original resolution and perform back-projection to project the reference-view pixel coordinates to the camera coordinates:

$$\mathbf{X}(\mathbf{p}) = \mathbf{K}^{-1}\mathbf{p}\mathbf{D}^l_{\text{ini},m}(\mathbf{p}), \tag{13}$$

where $\mathbf{X}(\mathbf{p}) \in \mathbb{R}^{3\times 1}$ denotes back-projected camera coordinates at pixel $\mathbf{p}$, $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is the scaled camera intrinsics of the reference view. For each pixel $\mathbf{p}$, we then search its n square neighboring pixels $\mathbf{p}_i, i \in \{0, 1, ..., n-1\}$ centered at pixel $\mathbf{p}$ and perform the same back-projection process to obtain corresponding camera coordinates $\mathbf{X}(\mathbf{p}_i) \in \mathbb{R}^{3\times 1}$. With local planar priors, the normal constraints are further imposed as:

$$\mathbf{N}(\mathbf{p}_i)\cdot(\mathbf{X}(\mathbf{p}_i) - \mathbf{X}(\mathbf{p})) = 0, \tag{14}$$

where $\mathbf{N}(\mathbf{p}_i) \in \mathbb{R}^{3\times 1}$ denotes the normal vector at pixel $\mathbf{p}_i$, $i \in \{0, 1, ..., n-1\}$. We then refine neighboring depth hypotheses by encoding normal geometries according to Eq. 14:

$$\mathbf{D}^l_m(\mathbf{p}_i) = \frac{\mathbf{N}(\mathbf{p}_i)\cdot(\mathbf{K}^{-1}\mathbf{p})}{\mathbf{N}(\mathbf{p}_i)\cdot(\mathbf{K}^{-1}\mathbf{p}_i)}\mathbf{D}^l_{\text{ini},m}(\mathbf{p}), \tag{15}$$

where $\mathbf{D}^l_m(\mathbf{p}_i)$ denotes the refined depth hypothesis at pixel $\mathbf{p}_i$. The above refinement is performed for all M_l depth hypotheses to obtain the refined depth hypotheses $\mathbf{D}^l_m \in \mathbb{R}^{H\times W}, m \in \{0, 1, ..., M_l - 1\}$, which is interpolated to corresponding feature level $\mathbf{D}^l_m \in \mathbb{R}^{\frac{H}{2^l}\times\frac{W}{2^l}}$ for adaptive cost volume aggregation, depth optimization and estimation.

III. Benchmark and Ablation Experiments

In this section, we describe the implementation details and evaluate our method on standard MVS benchmarks [13]–[15]. We also perform real-world experiments to demonstrate the generalization capability of our approach to large-scale scenarios. Benchmark visualizations, along with details on datasets and evaluation metrics, are provided in the appendix.

A. Implementation Details

The proposed network is implemented by PyTorch and trained on the DTU training set. The original DTU dataset [13] only contains ground-truth point clouds scanned by the structured light scanner. Therefore, following the common practices [10], [12], [22], we generate the per-view ground-truth depth map by screened Poisson surface reconstruction and per-view depth map rendering for end-to-end training. The feature level L is set to 3 to form a three-stage coarse-to-fine network. The number of input views N is 5, the number of source views N_s and the point distance threshold ϵ_l for depth inconsistency check are set to 8 and 0.2, respectively.

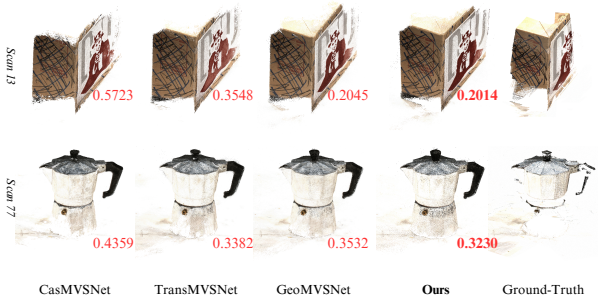


Fig. 2. Reconstruction comparison of *scan 13* and *scan 77* on the DTU evaluation set, where our method achieves more complete dense reconstruction for low-textured and non-Lambertian surfaces under bright light. The bottom-right number denotes the completeness error in mm (**lower the better**).

TABLE I
QUANTITATIVE BENCHMARKING RESULTS ON DTU EVALUATION SET FOR EVALUATING POINT CLOUD RECONSTRUCTION PERFORMANCE ($N = 5$, $H \times W = 864 \times 1152$, **LOWER THE BETTER**)

Type	Methods	Year	Mean Error Distance on 22 scenes		
			ACC. ↓ (mm)	Comp. ↓ (mm)	**Overall*** ↓ (mm)
Traditional	Camp [5]	2008	0.835	0.554	0.695
	Furu [6]	2010	0.613	0.941	0.777
	Tola [7]	2012	0.342	1.190	0.766
	Gipuma [8]	2015	**0.283**	0.873	0.578
	Colmap [9]	2016	0.400	0.664	0.532
Learning-based	SurfaceNet [32]	2017	0.450	1.040	0.745
	MVSNet [10]	2018	0.396	0.527	0.462
	R-MVSNet [11]	2019	0.385	0.459	0.422
	P-MVSNet [33]	2019	0.406	0.434	0.420
	Point-MVSNet [34]	2019	0.342	0.411	0.376
	D^2HC-RMVSNet [35]	2020	0.395	0.378	0.386
	AttMVS [36]	2020	0.383	0.329	0.356
	CVP-MVSNet [37]	2020	0.296	0.406	0.351
	UCS-Net [38]	2020	0.338	0.349	0.344
	AA-RMVSNet [16]	2021	0.376	0.339	0.357
	EPP-MVSNet [39]	2021	0.413	0.296	0.355
	PatchMatchNet [40]	2021	0.427	0.277	0.352
	IterMVS [41]	2022	0.373	0.354	0.363
	CDS-MVSNet† [20]	2022	0.365	0.281	0.323
	UniMVSNet† [17]	2022	0.364	0.279	0.321
	TransMVSNet† [18]	2022	0.360	0.271	0.316
	Vis-MVSNet [21]	2023	0.369	0.361	0.365
	GeoMVSNet† [22]	2023	0.370	0.275	0.323
	ET-MVSNet† [23]	2023	0.359	0.265	0.312
	BH-RMVSNet [24]	2024	0.368	0.303	0.335
	LCM-MVSNet [19]	2024	0.358	0.275	0.317
	CasMVSNet (Baseline) [12]	2020	0.325	0.385	0.355
	Ours		0.354	**0.246**	**0.300**

* **Note that the overall score is the summary measure of the overall reconstruction performance**.
↓ The ↓ means that the smaller value indicates the better MVS performance.
† Re-evaluated by utilizing the released optimal checkpoints with the same depth map filtering & fusion method and in the same platform as ours.

The image and ground-truth depth map resolutions are resized to $H \times W = 512 \times 640$. The depth range is set to $425mm$ to $905mm$ to uniformly sample depth hypotheses, where the number of depth hypotheses planes M_l is defined as $48, 32, 8$ from coarse to fine level, respectively. The corresponding depth interval is set to $4, 2, 1$ times of the coarse-level depth interval and the loss weight λ_l is $1, 1, 2$ from coarse to fine level. The number of square neighboring pixels n is set to 8 to form a 3×3 local neighbor region for depth hypotheses refinement. The network is optimized by the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.999$) for 30 epochs on two NVIDIA RTX 3090Ti GPUs with a batch size of 2 on each GPU. The initial learning rate and weight decay of the optimizer are 0.001 and 0.0001, respectively. The multi-step learning rate scheduler is adopted to decay the initial learning rate by a factor of 0.5 at epochs $8, 12, 16$, and 20, respectively.

B. Benchmarking Performance

Benchmarking on DTU We benchmark our method on the DTU evaluation set by comparing it with several traditional

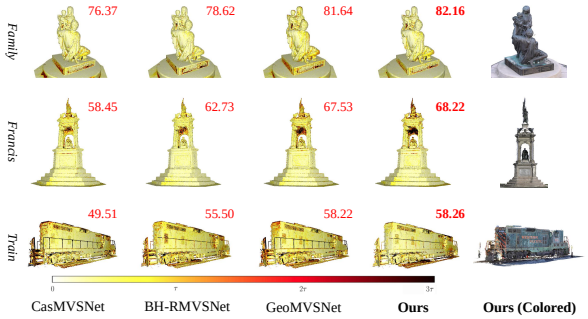


Fig. 3. Reconstruction error rendering of scene *Family*, *Francis*, and *Train* on the Tanks and Temples benchmark. The darker color indicates a larger reconstruction error and τ denotes the per-scene point distance threshold. The top-right number denotes the F-score in % (**higher the better**).

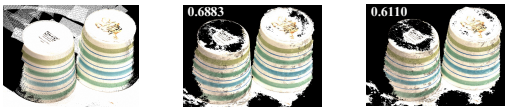
TABLE II
QUANTITATIVE BENCHMARKING RESULTS ON TANKS AND TEMPLES FOR EVALUATING POINT CLOUD RECONSTRUCTION PERFORMANCE ($N = 11$, $H \times W = 1080 \times 1920$, **HIGHER THE BETTER**)

Type	Methods	Year	Mean Error Percentage on 8 scenes		
			Precision ↑ (%)	Recall ↑ (%)	**F-score*** ↑ (%)
Traditional	MVE [42]	2015	19.67	40.16	25.37
	OpenMVG [43] + MVE [42]	2016	27.96	61.97	38.00
	Colmap [9]	2016	43.16	44.48	42.14
	Pix4D [44]	2016	46.85	41.58	43.24
	VisualSfM [45] + OpenMVS [46]	2020	21.76	30.39	24.45
	OpenMVG [43] + OpenMVS [46]	2020	35.25	55.16	41.71
Learning-based	MVSNet [10]	2018	40.23	49.70	43.48
	Point-MVSNet [34]	2019	41.27	60.13	48.27
	R-MVSNet [11]	2019	43.74	57.60	48.40
	P-MVSNet [33]	2019	49.93	63.82	55.62
	CVP-MVSNet [37]	2020	51.41	60.19	54.03
	UCSNet [38]	2020	46.66	70.34	54.83
	D^2HC-RMVSNet [35]	2020	49.88	74.08	59.20
	AttMVS [36]	2020	61.89	58.93	60.05
	PatchMatchNet [40]	2021	43.64	69.37	53.15
	AA-RMVSNet [16]	2021	52.68	75.69	61.51
	EPP-MVSNet [39]	2021	53.03	75.58	61.68
	IterMVS [41]	2022	46.82	73.50	56.22
	CDS-MVSNet [20]	2022	53.23	74.39	61.58
	TransMVSNet [18]	2022	55.14	76.73	63.52
	UniMVSNet [17]	2022	57.54	73.82	64.36
	Vis-MVSNet [21]	2023	54.44	70.48	60.03
	ET-MVSNet [23]	2023	58.52	75.45	65.49
	GeoMVSNet [22]	2023	**59.75**	74.28	65.89
	BH-RMVSNet [24]	2024	53.24	75.82	61.96
	LCM-MVSNet [19]	2024	59.25	68.56	63.33
	CasMVSNet [12] (Baseline)	2020	47.62	74.01	56.84
	Ours		57.17	**79.12**	**65.90**

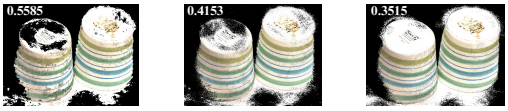
* **Note that the F-score is the summary measure of the overall reconstruction performance**.
↑ The ↑ means that the larger value indicates the better MVS performance.

and dozens of learning-based MVS methods to quantify its point cloud reconstruction performance as shown in Table I. The reconstruction accuracy (Acc.), completeness (Comp.), and overall score (Overall) in mean error distance (mm) of 22 scenes are reported (**lower the better**). N is set to 5 with the image resolution of $H \times W = 864 \times 1152$ for depth map estimation. The fusible [29] is adopted to fuse multi-view depth maps into the final point cloud. All settings follow common practices for a fair comparison. Extensive experiments show that our method achieves state-of-the-art reconstruction performance in terms of overall score by striking an excellent trade-off between reconstruction accuracy and completeness, verifying the effectiveness of our proposed modules for improving point cloud reconstruction. The qualitative comparison between our method and existing methods [12], [18], [22] in Fig. 2 demonstrates that our method achieves much more complete reconstruction with fine-grained details for non-Lambertian and low-textured surfaces under bright light, qualitatively demonstrating the quantitative benchmarking results.

Benchmarking on Tanks and Temples As shown in



(a) Ground Truth (b) Baseline (c) Baseline w/ Normal



(d) w/ Adaptive (e) w/o Normal (f) Full Model

Fig. 4. Ablation reconstruction results of a high-reflection scene (*scan 48* on DTU evaluation set) under bright light exposure with different model settings, demonstrating the effectiveness of proposed modules. The top-left number denotes the overall score in mm (**lower the better**).

TABLE III
QUANTITATIVE BENCHMARKING RESULTS ON BLENDEDMVS VALIDATION SET FOR EVALUATING DEPTH ESTIMATION PERFORMANCE ($N = 5$, $H \times W = 576 \times 768$, **LOWER THE BETTER**)

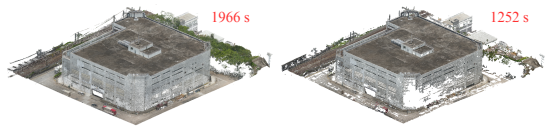
Methods	Year	Mean Depth Error on 7 scenes		
		EPE ↓	e_1 (%) ↓	e_3 (%) ↓
MVSNet [10]	2018	1.49	21.98	8.32
CVP-MVSNet [37]	2020	1.90	19.73	10.24
EPP-MVSNet [39]	2021	1.17	12.66	6.20
CDS-MVSNet [20]	2022	1.80	22.88	9.28
TransMVSNet [18]	2022	1.05	13.74	5.47
UniMVSNet [17]	2022	1.17	11.27	4.96
IterMVS [41]	2022	0.87	12.15	4.48
ET-MVSNet [23]	2023	3.44	23.40	12.18
GeoMVSNet [22]	2023	2.37	23.20	11.76
Vis-MVSNet [21]	2023	1.56	21.68	8.36
LCM-MVSNet [19]	2024	1.02	10.15	4.54
CasMVSNet (Baseline) [12]	2020	1.43	19.73	10.24
Ours		**0.77**	**9.54**	**3.96**

Table II, we further quantify the point cloud reconstruction performance of our method on the intermediate set of the Tanks and Temples benchmark to evaluate its generalization ability on large-scale scenes. The reconstruction precision, recall, and F-score in mean error percentage (%) of 8 scenes are reported (**higher the better**). The model is fine-tuned on the BlendedMVS training set for 20 epochs to improve its generalization ability on large-scale real-world scenes by setting $N = 7$ with $H \times W = 576 \times 768$. For benchmarking, N is set to 11 with $H \times W = 1080 \times 1920$ for depth estimation, and the dynamic fusion [35] is adopted for point cloud reconstruction. All settings follow common practices for a fair comparison. Extensive experiments show that our method achieves the highest F-score compared to traditional and dozens of learning-based methods. We render the reconstruction error as shown in Fig. 3, where our method outperforms recent methods [12], [22], [24] by a large margin.

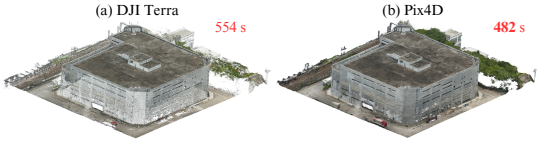
Benchmarking on BlendedMVS Unlike previous benchmarks, BlendedMVS reports the endpoint error (EPE), 1-threshold error (e_1), and 3-threshold error (e_3) to quantify the depth estimation performance. For fine-tuning and benchmarking, N is set to 5 with $H \times W = 576 \times 768$ by following common practices. Extensive experiments in Table III demonstrate that our method obtains the lowest depth estimation error compared to recent learning-based MVS methods.

C. Ablation Experiments

As shown in Fig. 4 and Table IV, we respectively ablate different components of our method on the DTU evaluation set to verify their effectiveness and efficiency, including cross-view



(a) DJI Terra (b) Pix4D



(c) Colmap (d) **Ours**

Fig. 5. Real-world large-scale reconstruction results. Our method achieves much more complete and faster point cloud reconstruction than the prevalent industrial reconstruction solutions DJI Terra [47], Pix4D [44], and Colmap [9].

TABLE IV
ABLATION EXPERIMENTS ON DTU EVALUATION SET

Model Settings				Mean Error Distance on 22 Scenes			Computational Costs	
	Ada.	Dep.	Nor.	Acc. ↓ (mm)	Comp. ↓ (mm)	Overall* ↓ (mm)	Train / Test Memory†	Train / Test Runtime‡ (s)
(a)				0.359 (+0.0009)	0.339 (+0.0008)	0.349 (+0.0009)	13449 / 4863	0.465 / 0.222
(b)	✓			0.357 (+0.0001)	0.314 (+0.0005)	0.335 (+0.0008)	13507 / 4239	0.434 / 0.185
(c)	✓	✓		**0.344** (−0.0002)	0.272 (+0.0006)	0.308 (+0.0002)	13653 / 4239	0.456 / 0.185
(d)	✓	✓	✓	0.354 (+0.0007)	**0.246** (+0.0009)	**0.300** (+0.0008)	12709 / 10883	0.471 / 0.241

* **Note that the overall score is the summary measure of the overall reconstruction performance**.
† The batch size is set to 2 and 1 to measure the memory footprint in the train ($H \times W = 512 \times 640$) and test ($H \times W = 864 \times 1152$) mode.
‡ The batch size is set to 1 to measure the runtime in the train ($H \times W = 512 \times 640$) and test ($H \times W = 864 \times 1152$) mode.

adaptive cost volume aggregation (Ada.), depth consistency optimization (Dep.), and normal-assisted depth hypotheses refinement (Nor.). The experiments show that the Ada. module effectively enhances overall reconstruction by considering per-view per-pixel visibility compared to the baseline method [12]. The Dep. module significantly improves the reconstruction accuracy and completeness to the state-of-the-art performance by explicitly suppressing depth inconsistency. The Nor. module further boosts the overall reconstruction performance by encouraging local depth consistency. For training, the Nor. module reduces the memory footprint when combined with the classification loss-based Dep. module by achieving more consistent and compact depth ranges. For testing, the Nor. module imposes considerable computational costs due to the increase in image and normal resolution but the efficiency is still comparable to existing methods [17], [19], [24], [37].

IV. REAL-WORLD EXPERIMENTS

We perform real-world experiments to demonstrate the generalization ability of GE-MVS for large-scale scene reconstruction. We capture 190 aerial images ($H \times W = 864 \times 1152$) of a low-textured warehouse and deploy the GE-MVS for multi-view depth estimation and point cloud reconstruction as shown in Fig. 5, where our method obtains much denser and faster reconstruction compared to prevalent industrial reconstruction solutions including DJI Terra, Pix4D, and Colmap on the NVIDIA RTX 4090 GPU. Table II quantitatively verifies the superiority of GE-MVS over industrial solutions.

V. CONCLUSION

In this paper, we have presented the GE-MVS to strengthen the geometric cues encoding during network learning for more accurate and complete depth estimation and point cloud reconstruction. Extensive experiments on the standard MVS benchmarks demonstrate the state-of-the-art performance of GE-MVS. Real-world large-scale reconstruction experiments witness the generalization ability and superiority of GE-MVS over the most prevalent industrial reconstruction solutions.

REFERENCES

[1] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning continuous 3d reconstructions for geometrically aware grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 11 516–11 522.

[2] J. Lim, N. Lawrance, F. Achermann, T. Stastny, R. Bähnemann, and R. Siegwart, "Fisher information based active planning for aerial photogrammetry," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1249–1255.

[3] M. Lv, D. Tu, X. Tang, Y. Liu, and S. Shen, "Semantically guided multi-view stereo for dense 3d road mapping," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 189–11 195.

[4] Y. Wang, Y. Ji, H. Tsuchiya, H. Asama, and A. Yamashita, "Learning pseudo front depth for 2d forward-looking sonar-based multi-view stereo," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8730–8737.

[5] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla, "Using multiple hypotheses to improve depth-maps for multi-view stereo," in *European Conference on Computer Vision*. Springer, 2008, pp. 766–779.

[6] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[7] E. Tola, C. Strecha, and P. Fua, "Efficient large-scale multi-view stereo for ultra high-resolution image sets," *Machine Vision and Applications*, vol. 23, no. 5, pp. 903–920, 2012.

[8] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 873–881.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *International Journal of Computer Vision*, vol. 120, no. 2, pp. 153–168, 2016.

[14] Y. Yao, Z. Luo, S. Li, J. Zhang, Y. Ren, L. Zhou, T. Fang, and L. Quan, "Blendedmvs: A large-scale dataset for generalized multi-view stereo networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1787–1796.

[15] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[16] 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.

[17] 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.

[18] 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.

[19] G. Yang, X. Zhou, C. Gao, X. Chen, and B. M. Chen, "Learnable cost metric-based multi-view stereo for point cloud reconstruction," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 9, pp. 11 519–11 528, 2024.

[20] K. T. Giang, S. Song, and S. Jo, "Curvature-guided dynamic scale networks for multi-view stereo," *arXiv preprint arXiv:2112.05999*, 2021.

[21] J. Zhang, S. Li, Z. Luo, T. Fang, and Y. Yao, "Vis-mvsnet: Visibility-aware multi-view stereo network," *International Journal of Computer Vision*, vol. 131, no. 1, pp. 199–214, 2023.

[22] Z. Zhang, R. Peng, Y. Hu, and R. Wang, "Geomvsnet: Learning multi-view stereo with geometry perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 508–21 518.

[23] T. Liu, X. Ye, W. Zhao, Z. Pan, M. Shi, and Z. Cao, "When epipolar constraint meets non-local operators in multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 18 088–18 097.

[24] 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*, vol. 30, no. 7, pp. 3062–3073, 2024.

[25] U. Kusupati, S. Cheng, R. Chen, and H. Su, "Normal assisted stereo depth estimation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2189–2199.

[26] W. Tong, X. Guan, J. Kang, P. Z. Sun, R. Law, P. Ghamisi, and E. Q. Wu, "Normal assisted pixel-visibility learning with cost aggregation for multiview stereo," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 686–24 697, 2022.

[27] J. Wu, R. Li, H. Xu, W. Zhao, Y. Zhu, J. Sun, and Y. Zhang, "Gomvs: Geometrically consistent cost aggregation for multi-view stereo," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 207–20 216.

[28] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 283–291.

[29] S. Galliani, K. Lasinger, and K. Schindler, "Fusible," *https://github.com/kysucix/fusibile*, 2015.

[30] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[31] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 786–10 796.

[32] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3d neural network for multiview stereopsis," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2326–2334.

[33] K. Luo, T. Guan, L. Ju, H. Huang, and Y. Luo, "P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[34] R. Chen, S. Han, J. Xu, and H. Su, "Point-based multi-view stereo network," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1538–1547.

[35] J. Yan, Z. Wei, H. Yi, M. Ding, R. Zhang, Y. Chen, G. Wang, and Y.-W. Tai, "Dense hybrid recurrent multi-view stereo net with dynamic consistency checking," in *European conference on computer vision*. Springer, 2020, pp. 674–689.

[36] 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.

[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] S. Cheng, Z. Xu, S. Zhu, Z. Li, L. E. Li, R. Ramamoorthi, and H. Su, "Deep stereo using adaptive thin volume representation with uncertainty awareness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[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] 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.

[41] 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.

[42] S. Fuhrmann, F. Langguth, N. Moehrle, M. Waechter, and M. Goesele, "Mve—an image-based reconstruction environment," *Computers & Graphics*, vol. 53, pp. 44–53, 2015.

[43] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "OpenMVG: Open multiple view geometry," in *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2016, pp. 60–74.
[44] EPFL, "Pix4dmapper: The leading photogrammetry software for professional drone mapping," *https://www.pix4d.com/*, 2024.
[45] C. Wu, "Visualsfm: A visual structure from motion system," *http://www.cs. washington. edu/homes/ccwu/vsfm*, 2011.
[46] D. Cernea, "OpenMVS: Multi-view stereo reconstruction library," 2020. [Online]. Available: https://cdcseacave.github.io/openMVS
[47] DJI, "Dji terra: Make the world your digital asset," *https://enterprise.dji.com/zh-tw/dji-terra*, 2024.