This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3165860, IEEE Transactions on Visualization and Computer Graphics

Bidirectional Hybrid LSTM Based Recurrent Neural Network for Multi-view Stereo

Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang

Abstract—Recently, deep learning based multi-view stereo (MVS) networks have demonstrated their excellent performance on various benchmarks. In this paper, we present an effective and efficient recurrent neural network (RNN) for accurate and complete dense point cloud reconstruction. Instead of regularizing the cost volume via conventional 3D CNN or unidirectional RNN like previous attempts, we adopt a bidirectional hybrid Long Short-Term Memory (LSTM) based structure for cost volume regularization. The proposed bidirectional recurrent regularization is able to perceive full-space context information comparable to 3D CNNs while saving runtime memory. For post-processing, we introduce a visibility based approach for depth map refinement to obtain more accurate dense point clouds. Extensive experiments on DTU, Tanks and Temples and ETH3D datasets demonstrate that our method outperforms previous state-of-the-art MVS methods and exhibits high memory efficiency at runtime.

Index Terms—3D reconstruction, deep learning, multi-view stereo, recurrent neural network, point clouds.

1 Introduction

Multi-view stereo (MVS), which aims to reconstruct a dense representation for an observed 3D scene from more than two calibrated images, is one of the fundamental issues of computer vision and photogrammetry. Developing general and efficient MVS algorithms has received a wide concern for decades in various applications, such as cultural heritage, urban planning, remote sensing, and autonomous driving.

Traditional MVS algorithms [1], [2], [3], [4], [5], [6], [7] mostly adopt hand-crafted features and similarity metrics to measure multi-view matching consistency, e.g. Sum of Squared Differences (SSD) and Normalized Cross-Correlation (NCC). Pixel-wise depth values are estimated and optimized according to these metrics through spatial propagation and stochastic optimization. Traditional MVS methods are able to achieve promising results on simple scenes. However, these methods suffer from low-textured regions and severe occlusions where mismatches inevitably occur, leading to incomplete and inaccurate depth estimation and point cloud reconstruction. Meanwhile, the propagation procedure sequentially optimizes depth values pixel by pixel, which is computationally expensive and hard to be fully parallelized.

Recent attempts [8], [9], [10], [11], [12], [13], [14], [15] introduce convolutional neural networks (CNNs) for powerful feature extraction and multi-view information aggregation. These advanced MVS networks exhibit superior performance compared to traditional methods and gain growing interests in the computer vision society. Generally, deep learning based MVS methods encode the scene geometries into cost volumes upon CNN based features, to make stereo matching more reliable and robust.

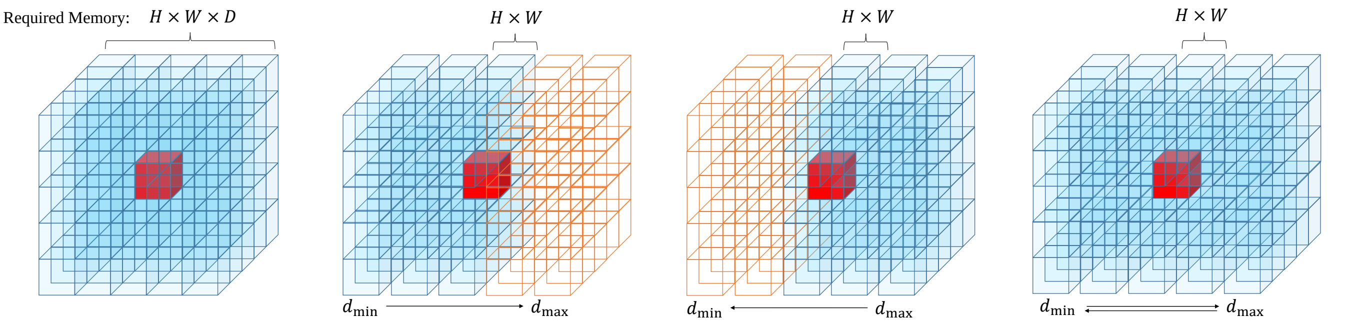
MVSNet [8] and its variants [10], [16], [17], [18], [19], [20] utilize 3D CNNs for cost volume regularization, which is effective to gather full-space information but occupies massive runtime memory, as shown in Fig. 1(a). To reduce memory consumption during cost volume regularization, recent attempts [9], [21] introduce recurrent neural networks (RNNs) to sequentially regularize 2D cost maps along the depth direction. However, as illustrated in Fig. 1(b)(c), the unidirectional RNN structure used in these methods leads to a loss of global context information, which might lower the depth prediction accuracy and the final reconstruction quality.

To this end, we propose a novel bidirectional hybrid Long Short-Term Memory (LSTM) based recurrent MVS network, namely BH-RMVSNet. The proposed network is trained end-to-end simultaneously in forward and backward depth directions, which enables hierarchical context information to be fully considered as 3D CNNs without requiring massive computational resources, as shown in Fig.1(d). The bidirectional MVS network is able to estimate more accurate and complete depth maps than the unidirectional schemes. Besides, a depth map refinement approach is adopted to further improve the original coarse depths according to multi-view visibility information, which contributes to a robust reconstruction in complex scenarios.

We conduct extensive experiments on DTU dataset [22], Tanks and Temples benchmark [23], BlendedMVS dataset [24], and ETH3D benchmark [25]. The proposed BH-RMVSNet achieves state-of-the-art performance for point cloud reconstruction. Specifically, our method ranks 1^{st} on the $intermediate$ set of Tanks and Temples online benchmark (date: May 15, 2021), which consists of complex outdoor scenes. We also investigate the network architecture through ablation experiments to validate the contributions of different key components.

In summary, our main contributions are as follows:

- *Z. Wei, Y. Chen and G. Wang are with the Department of EECS, Peking University, Beijing, China, 100871.*
E-mail: {weizizhuang,chenyisong,wgp}@pku.edu.cn
- *Q. Zhu and C. Min are with the Department of EECS, Peking University, Beijing, China, 100871.*
E-mail: {zqt,minchen}@stu.pku.edu.cn



(a) 3D CNNs Regularization (b) Forward Recurrent Regularization (c) Backward Recurrent Regularization (d) Bidirectional Recurrent Regularization

Fig. 1: Illustrations of different cost regularization schemes. Blue voxels denote the receptive field of a particular red voxel, while orange voxels do not contribute to the regularized cost. 3D CNNs regularization enables the cost information across the whole space to be fully considered, but requires a cubical runtime memory. Either forward or backward recurrent regularization dramatically reduces the memory consumption, but cannot gather complete context information along the depth direction. The proposed bidirectional recurrent regularization is able to gather full-space context information comparable to 3D CNNs in a more efficient manner regarding runtime memory consumption.

- We introduce a novel multi-view stereo framework using bidirectional hybrid LSTM based recurrent structure for cost volume regularization, which enables context information along depth directions to be fully considered.
- We design a visibility-based depth map refinement approach for post-processing, which effectively improves the overall quality of depth values to obtain more accurate dense point clouds.
- Our method outperforms most previous methods and achieves state-of-the-art results on DTU dataset and Tanks and Temples benchmark, showing strong generalization ability.

2 Related Work

2.1 Leaning-based MVS Methods

Recently, learning-based MVS methods have demonstrated great potentials to solve the problem of MVS. The earlier attempts, such as SurfaceNet [16] and LSM [17], first warp the multi-view image features into voxel-based cost volumes and regularize them using 3D CNNs. However, these approaches suffer from the common deficiency of voxel-based representation. In contrast, MVSNet [8] introduces differentiable homography to construct the cost volume upon multi-view image features, and then utilizes 3D regularization for depth inference. MVSNet is regarded as a pioneer of end-to-end learning-based MVS algorithms, which achieves leading performance for point cloud reconstruction at its release. In the last three years, there are several variants followed on MVSNet [8] to improve the reconstruction quality. P-MVSNet [19] adopts a patch-wise confidence aggregation module to improve multi-view matching accuracy and robustness. AttMVS [13] introduces a self-attention cost volume aggregation mechanism in MVS networks. Vis-MVSNet [14] effectively improves the uncertain depth values based on pixel-wise visibility. PVA-MVSNet [20] infers multi-scale depth maps and selects reliable depths from different resolutions to produce depth maps. The aforementioned methods mainly focus on developing accurate and robust learning-based MVS algorithms. However, these methods rely on 3D CNNs for cost volume regularization, which requires a cubical runtime memory. This issue constrains their scalability for high resolution images and large-scale scenes.

2.2 Scalable Learning-based MVS

In order to mitigate the scalability issue, there are mainly two routes for memory efficiency improvement. One is to adopt the coarse-to-fine pattern, in which coarse depth maps are predicted firstly and used to narrow the depth range for finer depth maps. CasMVSNet [11], UCS-Net [26], CVP-MVSNet [12], Vis-MVSNet [14] and PatchmatchNet [15] all follow this coarse-to-fine ideology to build multi-stage cost volumes. Though achieving promising results, the regularization process on a certain stage fails to gather global spatial information, which causes unsatisfactory performance on challenging datasets, such as the $advanced$ set of Tanks and Temples [23] benchmark and BlendedMVS [24] dataset.

The second route to reduce runtime memory is to divide a 3D cost volume into 2D cost maps and regularize them along the depth direction. R-MVSNet [9] first constructs the recurrent structured cost volume using convolutional gated recurrent unit (GRU). Inspired by R-MVSNet [9], D^2HC-RMVSNet [21] replaces the stacked GRU with a hybrid LSTM module to sequentially regularize cost volumes and exhibits stronger generalizability. However, the aforementioned recurrent MVS networks only involve a unidirectional depth sequence for a certain round of network training or testing, which limits the usage of global context information.

Based on the above analysis for learning-based MVS methods, we follow the previous successes of memory-efficient recurrent MVS networks and extend them by a bidirectional recurrent cost regularization scheme. Our method takes spatial information into adequate consideration and reduces runtime memory. As a result, our method exhibits excellent performance for MVS reconstruction.

3 Network Architecture

This section describes the detailed network architecture of our proposed BH-RMVSNet, as is visualized in Fig. 2. We

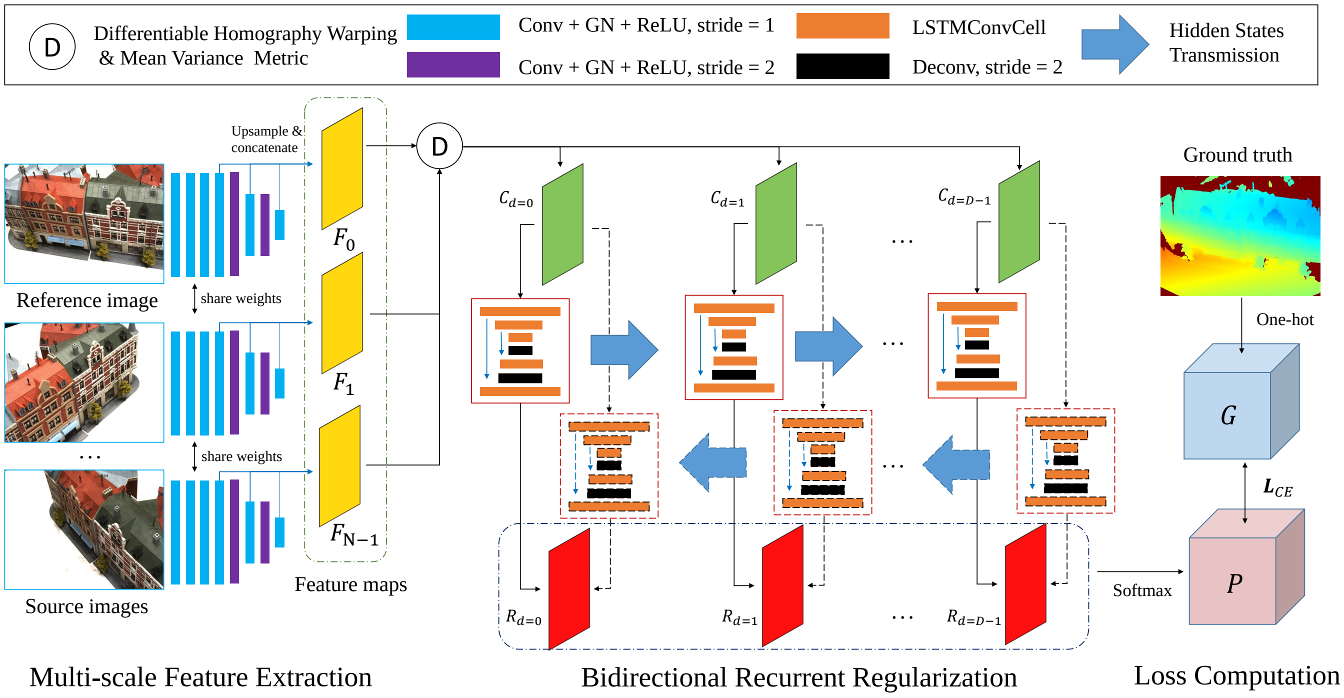


Fig. 2: The overall architecture of the proposed BH-RMVSNet. For each input image, its feature map is obtained by the concatenation of multi-scale CNN features, and is warped to the reference camera frustum through differentiable homography to compute 2D cost maps. These cost maps are sequentially regularized by U-shape hybrid LSTM for both forward and backward depth directions, where the combined cost maps from bidirectional regularization are used for depth inference. The cross-entropy loss is adopted for back propagation.

design a novel bidirectional hybrid LSTM based recurrent MVS network, which adopts both the advantages of 3D CNNs and recurrent processing. Our network can predict accurate and complete depth maps to further reconstruct dense 3D point clouds of large-scale scenes. We first introduce the multi-scale feature extraction module in Sec. 3.1. Then, we present the bidirectional cost volume regularization module based on hybrid LSTM sub-networks in Sec. 3.2. At last we define our training loss in Sec. 3.3.

3.1 Multi-scale Feature Extraction

Given a reference image I_0 and $N-1$ source images $\{I_i\}_{i=1}^{N-1}$ of size $H \times W$, the feature extraction module aims to extract representative image features, which are essential for accurate and robust multi-view matching. Generally, the features extracted on a finer scale contain more local details, while the coarser scale features own larger receptive fields to gather global contexts. Therefore, we design a feature extraction network that combines features from different scales to contain both local and global information. The implementations and parameters of our feature extraction module are detailed in Tab. 1. We obtain the final feature maps $F_i \in \mathbb{R}^{H \times W \times C}$ from 3 different spatial scales, whose sizes are $H \times W$, $\frac{H}{2} \times \frac{W}{2}$, and $\frac{H}{4} \times \frac{W}{4}$, respectively. The smaller feature maps are bilinearly upsampled to $H \times W$ before concatenation. The number of concatenated feature channels C is 32. Furthermore, we replace the Batch Normalization [27] used in the original MVSNet [8] and R-MVSNet [9] by Group Normalization [28] for better performance under a relatively smaller batch size.

TABLE 1: Detailed information of the feature extraction module. Each convolutional layer includes a block of convolution, group normalization, and ReLU non-linearization. '$\oplus$' represents concatenation operation on channels and '$\otimes x$' represents bilinear upsampling with a scale of x for the corresponding feature map, respectively.

Feature Extraction: $I_i(H \times W \times 3) \rightarrow F_i(H \times W \times 32)$

Name	Layer Description	Output Tensor
Conv0_0	3×3 conv, stride 1	$H \times W \times 8$
Conv0_1	3×3 conv, stride 1	$H \times W \times 16$
Conv0_2	3×3 conv, stride 1	$H \times W \times 32$
Conv0_3	3×3 conv, stride 1	$H \times W \times 16$
Conv1_1	3×3 conv, stride 2	$\frac{H}{2} \times \frac{W}{2} \times 16$
Conv1_2	3×3 conv, stride 1	$\frac{H}{2} \times \frac{W}{2} \times 8$
Conv2_1	3×3 conv, stride 2	$\frac{H}{4} \times \frac{W}{4} \times 8$
Conv2_2	3×3 conv, stride 1	$\frac{H}{4} \times \frac{W}{4} \times 8$
Output	Conv0_3$\oplus$Conv1_2($\otimes$2) $\oplus$Conv2_2($\otimes$4)	$H \times W \times 32$

Based on the extracted multi-view feature maps F_i and their camera parameters, we warp the features from the source views to the reference view for stereo matching. Following common practices [8], [9], [13], [19], [20], [21], we sample D depth hypotheses between d_{min} and d_{max} where d_{min}, d_{max} are determined according to the results of Structure from Motion (SfM) [29]. Then the extracted features are warped through differentiable homography. With a particular depth hypothesis d, the corresponding homography relationship between the i-th source features

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3165860, IEEE Transactions on Visualization and Computer Graphics

and the reference is described as:

$$\mathbf{H}_i\left(d\right) = d\mathbf{K}_i\mathbf{T}_i\mathbf{T}_0^{-1}\mathbf{K}_0^{-1}, \tag{1}$$

where $\mathbf{K}_i$ and $\mathbf{T}_i$ denote the corresponding camera intrinsics and extrinsics respectively.

3.2 Bidirectional Recurrent Regularization

When multi-view feature maps have been obtained and warped to the reference camera frustum, the next step is to compute and regularize the multi-view matching cost for depth inference. First of all, we follow a widely used variance based metric [8], [9], [10], [11], [12] for cost volume generation. Typically, given a depth hypothesis d, the multi-view matching cost is defined by:

$$C(d) = \frac{1}{N}\sum_{i=0}^{N-1}\left(F_i(d) - \overline{F}\right)^2, \tag{2}$$

where $F_i(d)$ denotes the i-th warped feature map with depth d, and $\overline{F}$ represents the average among all feature maps. For cost volume regularization, instead of using 3D CNNs with cubical memory consumption, we take the advantage of memory-efficient recurrent MVS networks [9], [21] by sequentially processing the volume through the depth direction. In addition, we extend the previous unidirectional recurrent pipeline to bidirectional regularization by a hybrid LSTM, which keeps efficient in runtime memory but gathers context information comparable to the full-space 3D CNNs (Fig. 1(d)).

Since the multi-view matching cost volume $\boldsymbol{C}$ can be viewed as plane-sweeping 2D cost maps $\left\{C\left(i\right)\right\}_{i=0}^{D-1}$ along the depth direction, we denote the regularized matching cost maps as $\left\{R\left(i\right)\right\}_{i=0}^{D-1}$ at each depth hypothesis for sequential processing. In recent attempts [9], [21], the i-th output $R(i)$ only depends on the current input cost map $C(i)$ and the previous states $0, 1, \cdots, (i-1)$. As a result, the subsequent information is ignored, which leads to incomplete regularized results. Therefore, we introduce the bidirectional recurrent neural networks for MVS cost regularization. In our bidirectional scheme, all previous and subsequent states are involved to produce the final regularized cost. Different from the stacked GRU used in R-MVSNet [9], we follow the implementation in D^2HC-RMVSNet [21] using hybrid U-shape LSTM to control the information flow, which can better aggregate hierarchical context information.

As illustrated in Fig. 3, we apply 5-layer parallel LSTM-RNNs for both forward phase and backward phase to deliver intermediate outputs. At a certain depth slice d, each layer is a 3×3 convolutional LSTM unit connected by an encoder-decoder structure with $maxpooling$ and $deconvolution$ procedures. For a single LSTM unit, it consists of three key components, namely an input gate, a forget gate, and an output gate. The input gate is used to select valid information from the previous outputs. As for the j-th layer in the forward phase at depth d, the input gate $\mathbf{I}_d^j$ is defined by:

$$\mathbf{I}_d^j = \sigma\left(\mathbf{U_I}x_d^j + \mathbf{W_I}h_{d-1}^j + \mathbf{B_I}\right), \tag{3}$$

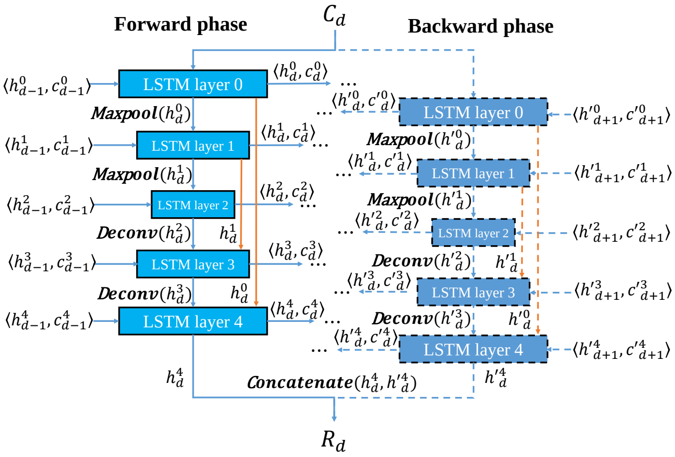

Forward phase
C_d
Backward phase
$\langle h_{d-1}^0, c_{d-1}^0\rangle$
LSTM layer 0
$\langle h_d^0, c_d^0\rangle$ …
… $\langle h'^0_d, c'^0_d\rangle$
LSTM layer 0
$\langle h'^0_{d+1}, c'^0_{d+1}\rangle$
Maxpool(h_d^0)
$\langle h_{d-1}^1, c_{d-1}^1\rangle$
LSTM layer 1
$\langle h_d^1, c_d^1\rangle$…
Maxpool(h'^0_d)
… $\langle h'^1_d, c'^1_d\rangle$
LSTM layer 1
$\langle h'^1_{d+1}, c'^1_{d+1}\rangle$
Maxpool(h_d^1)
$\langle h_{d-1}^2, c_{d-1}^2\rangle$
LSTM layer 2
$\langle h_d^2, c_d^2\rangle$…
Maxpool(h'^1_d)
… $\langle h'^2_d, c'^2_d\rangle$
LSTM layer 2
$\langle h'^2_{d+1}, c'^2_{d+1}\rangle$
Deconv(h_d^2)
h_d^1
$\langle h_{d-1}^3, c_{d-1}^3\rangle$
LSTM layer 3
$\langle h_d^3, c_d^3\rangle$…
Deconv(h'^2_d)
h'^1_d
… $\langle h'^3_d, c'^3_d\rangle$
LSTM layer 3
$\langle h'^3_{d+1}, c'^3_{d+1}\rangle$
Deconv(h_d^3)
h_d^0
$\langle h_{d-1}^4, c_{d-1}^4\rangle$
LSTM layer 4
$\langle h_d^4, c_d^4\rangle$…
Deconv(h'^3_d)
h'^0_d
… $\langle h'^4_d, c'^4_d\rangle$
LSTM layer 4
$\langle h'^4_{d+1}, c'^4_{d+1}\rangle$
h_d^4
Concatenate(h_d^4, h'^4_d)
h'^4_d
R_d

Fig. 3: Illustration of the information flow of the proposed bidirectional hybrid LSTM sub-network at depth hypothesis d. The next layer is updated according to the current inputs and the previous LSTM states. The bidirectional results are concatenated as the final regularized cost.

where σ is the $sigmoid$ function; $\mathbf{U}$ and $\mathbf{W}$ are the weight matrix; $\mathbf{B}$ is the bias term; x_d^j denotes the input data generated by the previous layer, and h_{d-1}^j stands for the output from the previous LSTM unit at $(d-1)$. The forget gate $\mathbf{F}_d^j$ determines whether to discard useless information, which is calculated by:

$$\mathbf{F}_d^j = \sigma\left(\mathbf{U_F}x_d^j + \mathbf{W_F}h_{d-1}^j + \mathbf{B_F}\right). \tag{4}$$

With the input gate and the forget gate, we update the current LSTM memory state c_d^j by:

$$c_d^j = \mathbf{F}_d^j c_{d-1}^j + \mathbf{I}_d^j \widetilde{c}_d^j, \tag{5}$$

where c_{d-1}^j is the memory content of the previous LSTM unit, and $\widetilde{c}_d^j$ represents the new memory gate which is defined by:

$$\widetilde{c}_d^j = tanh(\mathbf{U_C}x_d^j + \mathbf{W_C}h_{d-1}^j + \mathbf{B_C}). \tag{6}$$

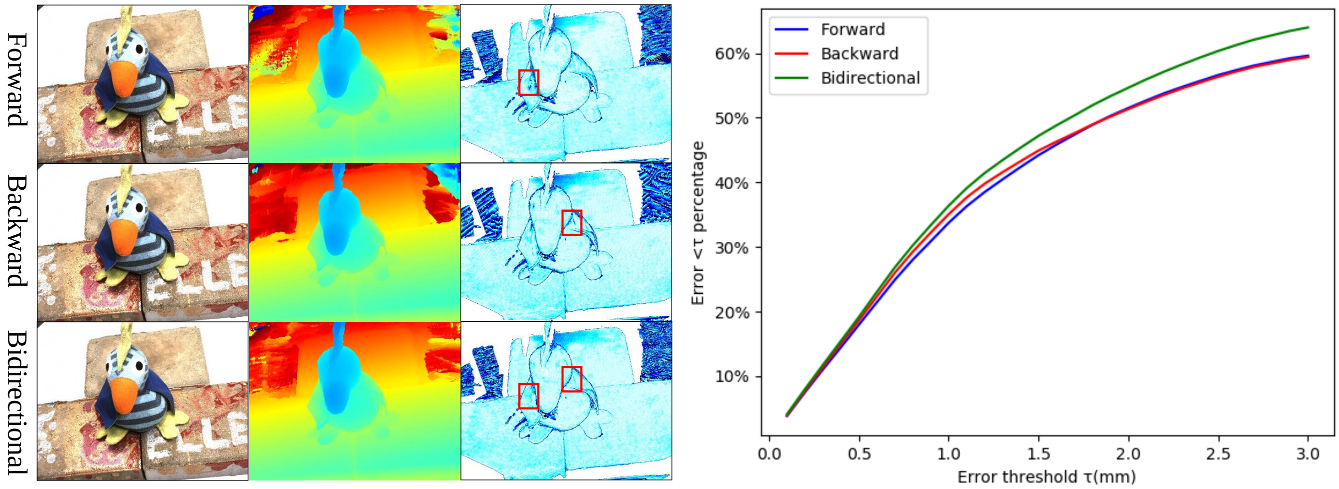
The output gate $\mathbf{F}_d^j$ controls the output flow, h_d^j is the final LSTM output activated at layer j:

$$\mathbf{O}_d^j = \sigma(\mathbf{U_O}x_d^j + \mathbf{W_O}h_{d-1}^j + \mathbf{B_O}), \tag{7}$$

$$h_d^j = \mathbf{O}_d^j tanh(c_d^j). \tag{8}$$

For the backward propagation, the current LSTM unit is updated according to the states of the LSTM unit at $(d+1)$, namely $\left\langle h_{d+1}^j, c_{d+1}^j\right\rangle$.

Finally, the results from two directions are concatenated as the regularized cost map R_d. The bidirectional network can access context in both forward and backward directions, improving the robustness and accuracy of depth prediction, as shown in Fig. 4. We set $stride = 2$ in $maxpooling$ and $deconvolution$ to aggregate multi-scale context information. Notice that the outputs of the first and the second layers are also used as inputs to the fifth and fourth layers. Specifically, for the 32-channel input cost maps $\left\{C\left(i\right)\right\}_{i=0}^{D-1}$, the input channels and output channels of each LSTM layers are $\{32, 16, 16, 32, 32\}$ and $\{16, 16, 16, 16, 8\}$ respectively. At



(a) Ref. image (b) Depth map (c) Error map (d) Error distribution

Fig. 4: Comparisons on the depth map, absolute error map and error distribution with different recurrent regularization schemes. (a) One reference image of $Scan4$ in DTU dataset [22]; (b) the inferred depth maps; (c) the absolute error maps; (d) distributions of the error maps. Our bidirectional cost regularization module outperforms the unidirectional MVS networks to obtain more delicate and accurate depth estimations.

last, the final outputs go through a 3×3 2D convolution layer to generate single-channel pixel-wise matching costs $\{R\,(i)\}_{i=0}^{D-1}$. Due to data dependencies for integrating the two directional results, some additional memory of size $H \times W \times D$ is needed to store all intermediate results in at least one direction. This makes extra cost compared to the original recurrent MVS network.

3.3 Training Loss

Since regression based loss causes over smoothing problems on the boundary areas, we follow R-MVSNet [9] and D^2HC-RMVSNet [21] to treat depth inference as a multi-class classification task for each depth hypothesis. Given the regularized costs $\{R\,(i)\}_{i=0}^{D-1}$, the probability volume P is obtained by a $softmax$ layer. The ground truth depth maps are encoded as binary occupancy volume G using *one-hot*. We apply the cross entropy loss for network training:

$$Loss = \sum_{\mathbf{p}\in\Omega}\sum_{d=0}^{D-1} -G_d\,(\mathbf{p}) \cdot \log\left(P_d\,(\mathbf{p})\right), \tag{9}$$

where Ω denotes the valid pixel set of ground truth depth maps. $G_d\,(\mathbf{p})$ and $P_d\,(\mathbf{p})$ represent the *one-hot* encoded ground truth and the predicted probability for depth hypothesis d at pixel $\mathbf{p}$, respectively.

4 Refinement & Fusion

By sequentially processing and $winner\text{-}take\text{-}all$ selection in the testing phase, the proposed network estimates multi-view depth maps from the regularized matching cost volume. These depth maps yielded by our BH-RMVSNet are already good enough for point cloud reconstruction. To further enhance the robustness in complex scenarios, we propose a visibility based depth map refinement to improve the initial depth maps according to the visible pixels of their neighbor views. At last, the refined multiple depth maps are fused to produce dense point clouds. We first illustrate the depth map refinement strategy in Sec. 4.1. Then, in Sec. 4.2, we introduce the final depth map fusion process.

4.1 Visibility-based Depth Refinement

Given a reference image I_0 and its neighbors $\{I_i\}_{i=1}^{N-1}$ with corresponding depth maps $\{D_i\}_{i=0}^{N-1}$ predicted by our BH-RMVSNet, we apply a pair-wise cross-view validation algorithm to figure out which pixels of the source images are visible in the reference. For a particular image pair $\langle I_0, I_j\rangle$, we firstly project each pixel $\mathbf{p}$ with depth $D_0(\mathbf{p})$ of I_0 into the 3D space:

$$\mathbf{X_p} = \mathbf{M}_0^{-1}\left(\mathbf{p}\cdot D_0(\mathbf{p}) - \mathbf{t}_0\right), \tag{10}$$

where $\mathbf{X_p}$ is the projected 3D point. $\left[\mathbf{M}_i^{(3\times3)}|\mathbf{t}_i^{(3\times1)}\right] = \mathbf{K}_i\mathbf{T}_i$ is the matrix product of camera intrinsics $\mathbf{K}_i$ and extrinsics $\mathbf{T}_i$. Then we find the corresponding pixel $\mathbf{p}'$ on the neighbor image I_j by:

$$\mathbf{p}' = \frac{1}{d'}\mathbf{K}_j\mathbf{T}_j\mathbf{X_p}, \tag{11}$$

where d' is the projected depth value from $\mathbf{X_p}$ to $\mathbf{p}'$. By the same way, we back-project $\mathbf{p}'$ as a 3D point $\mathbf{X'}_{\mathbf{p}'}$ according to the predicted depth $D_j(\mathbf{p}')$:

$$\mathbf{X'}_{\mathbf{p}'} = \mathbf{M}_j^{-1}\left(\mathbf{p}'\cdot D_j(\mathbf{p}') - \mathbf{t}_j\right). \tag{12}$$

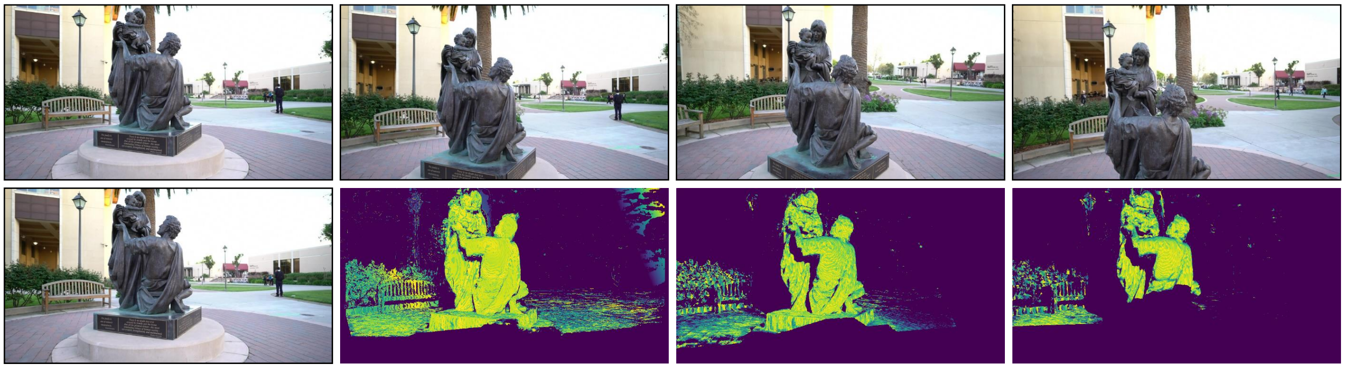
At last, the back-projected pixel $\mathbf{p}_{proj}$ and depth $\mathbf{d}_{proj}$ are obtained by:

$$d_{proj}\cdot\mathbf{p}_{proj} = \mathbf{K}_0\mathbf{T}_0\mathbf{X}'_{\mathbf{p}'}. \tag{13}$$

We validate the differences between the cross-projected results and the origins to measure the two-view visibility. In our experiments, a reference pixel $\mathbf{p}$ which is considered visible in the neighbor view I_j as $\mathbf{p}'$ should satisfy:

$$\begin{cases} \left|\mathbf{p} - \mathbf{p}_{proj}\right| < 1, \\ \left|1 - \frac{d_{proj}}{D_0(\mathbf{p})}\right| < 0.01. \end{cases} \tag{14}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3165860, IEEE Transactions on Visualization and Computer Graphics



Reference image

Source images & visibility maps

Fig. 5: Visualization of the visibility maps generated by the proposed cross-view validation algorithm. The left column shows the reference image of $Family$ in Tanks and Temples benchmark [23], while the right three columns show the source images from the neighbor views with the corresponding visibility maps. The color maps of all visible pixels represent different weights measured by the ZNCC metrics.

According to the visibility relations with the neighbor views, we update the reference depths by depth aggregation of all visible pixels. However, different views have different quality of observation. Considering this, we adopt the zero-mean normalized cross correlation (ZNCC) to measure the pixel-wise photo-consistency for different views. For an interest pixel $\mathbf{p}$ of the reference image I_0, Γ denotes the set of visible views and the refined depth value $\overline{D}_0(\mathbf{p})$ is obtained by:

$$\overline{D}_0(\mathbf{p}) = \frac{D_0(\mathbf{p}) + \sum_{i \in \Gamma} \omega_i d_i}{1 + \sum_{i \in \Gamma} \omega_i}, \tag{15}$$

where d_i is the depth through back-projection from the visible neighbor view I_i, and $\omega_i = e^{Z(\mathbf{p},\mathbf{p}'_i)-1}$ is the corresponding weight coefficient, $Z(\mathbf{p},\mathbf{p}'_i)$ represents the ZNCC score calculated by the visible-pixel pair $\langle \mathbf{p}, \mathbf{p}'_i \rangle$.

Fig. 5 visualizes the visibility maps measured by the ZNCC metrics. During the refinement, we iteratively update the multi-view depth maps. In each iteration step, all the depth maps are refined according to Equation (15). After several iterations, the final depths are used for further depth map fusion.

4.2 Depth Map Fusion

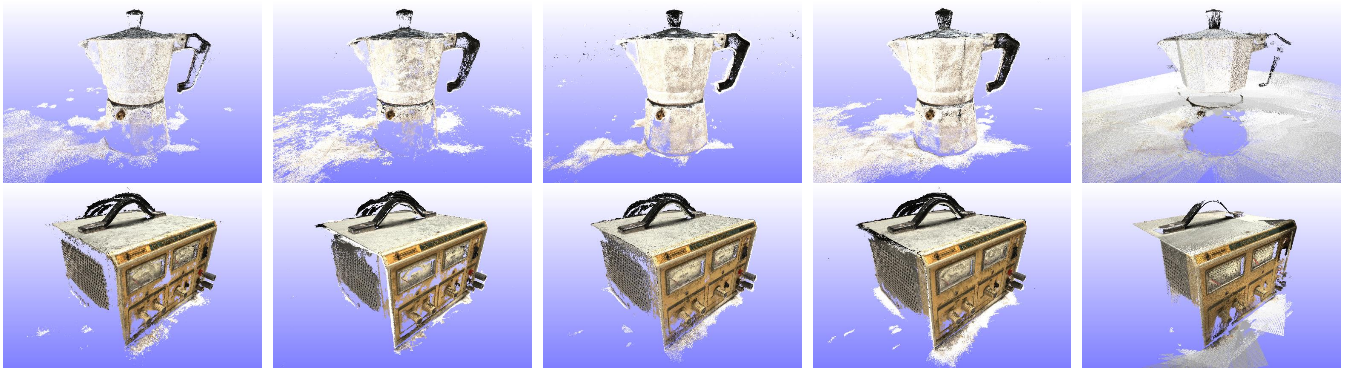
After the initial depth maps are refined by multi-view visibility, we filter and fuse them into a single 3D point cloud. Similar to the previous learning-based MVS methods [8], [9], [20], [21], we apply both photometric and geometric constraints for depth map filtering. As described in Sec. 3.2, we obtain the probability volume from regularized costs through a $softmax$ layer. It is noteworthy that the probability values reflect the matching quality for different depth hypotheses, and the depths with higher probability are often more accurate than the lower ones. So we regard the $softmax$ probability as the confidence of depth estimation. In our experiments, the depth values with a confidence lower than $\xi = 0.4$ are considered outliers to be discarded. Meanwhile, we follow D^2HC-RMVSNet [21] to filter the depth maps using a dynamic geometric consistency checking algorithm to measure the geometric constraint. At last, we fuse all reliable depth values into 3D space to produce 3D point clouds.

5 Experiments

5.1 Implementation Details

Training We train our BH-MVSNet on the $training$ set of DTU dataset [22], which consists of 124 different scenes with 49 to 64 views under well-controlled laboratory conditions with fixed camera trajectory. Following the common practices [8], [9], [11], [13], by screened Poisson surface reconstruction [30] on the provided laser point clouds, we generate the rendered ground truth depth maps of size 128×160 for network training. The input images are also resized to $H \times W = 128 \times 160$ as same as the corresponding ground truth, while the input view number is $N = 5$. The depth hypotheses are uniformly sampled from $d_{min} = 425mm$ to $d_{max} = 935mm$. The training procedure for the unfolded bidirectional network follows the original bidirectional RNN [31]. For a certain training sample, we firstly update the hidden states just for the forward LSTM regularization from d_{min} to d_{max} and then update the backward states from d_{max} to d_{min}, as well as the backpropagation process. The proposed network is implemented by PyTorch [32] and trained with $Adam$ [33]. The whole training phase takes about 3 days for 8 epochs with an initial learning rate of 0.001, which decays by 0.9 after each epoch. Batch size is set to 4 on 4 NVIDIA TITAN RTX GPUs with $24G$ memory each.

Testing We set $N = 7$ and $D = 512$ during the testing phase for depth maps estimation. The depth range $[d_{min}, d_{max}]$, camera parameters, and neighboring view selection are obtained by authors of MVSNet [8], and the depth hypotheses are uniformly sampled by an inverse depth manner in R-MVSNet [9] and CIDER [34]. For fitting our network, we resize and pad the input images to preserve the context information at image boundaries. For testing on DTU, Tanks and Temples , BlendedMVS and ETH3D



(a) R-MVSNet (b) PVA-MVSNet (c) D^{2}HC-RMVSNet (d) **BH-RMVSNet** (e) Ground Truth

Fig. 6: Visualization of the reconstructed point clouds for $Scan77$ and $Scan11$ of DTU dataset [22] by different methods. In contrast to the previous state-of-the-art methods [9], [20], [21], our proposed BH-RMVSNet achieves more complete and detailed 3D dense reconstructions for the real-world objects with less outliers. Note that the ground truth points obtained by the laser scanner are not always complete.

datasets, input images are resized to 600×800, 544×960, 576×768 and 1536×1024, respectively.

5.2 Benchmarking Results

We demonstrate that the proposed BH-RMVSNet outperforms its prototype D^2HC-RMVSNet with a significant margin on DTU [22], Tanks and Temples $intermediate$ and $advanced$ dataset [23]. Specifically, our method ranks 1^{st} on the $intermediate$ online leaderboard [35] of Tanks and Temples (date: May 15, 2021) among all submissions. To further investigate the scalability and generalizability of our method, we evaluate our method on BlendedMVS dataset [24] and ETH3D benchmark [25], which include wide-range aerial scenes and challenging indoor/outdoor scenes.

DTU Dataset We first evaluate our method on DTU [22] $evaluation$ set which consists of 22 different scans. For all these scans, depth range is set to $[d_{min}, d_{max}] = [425mm, 935mm]$ with the number of depth hypotheses $D = 512$. Our network infers the depth maps of size $H \times W = 600 \times 800$, which are refined and fused to produce 3D point clouds. The qualitative results of $Scan77$ and $Scan11$ compared to [9], [20] and [21] can be found in Fig. 6, our BH-MVSNet generates more complete point clouds with fewer outliers. Following previous conventions [8], [9], We adopt the official MATLAB script [22] for quantitative evaluation. To summarize the reconstruction quality, we calculate the $overall$ score by the average of mean $accuracy$ and mean $completeness$. The quantitative results are shown in Tab. 2, our method achieves the best $overall$ score compared with the previous state-of-the-art methods. In contrast to D^2HC-RMVSNet, our method significantly improves the $overall$ score by 13.2% and the $completeness$ by 19.8%.

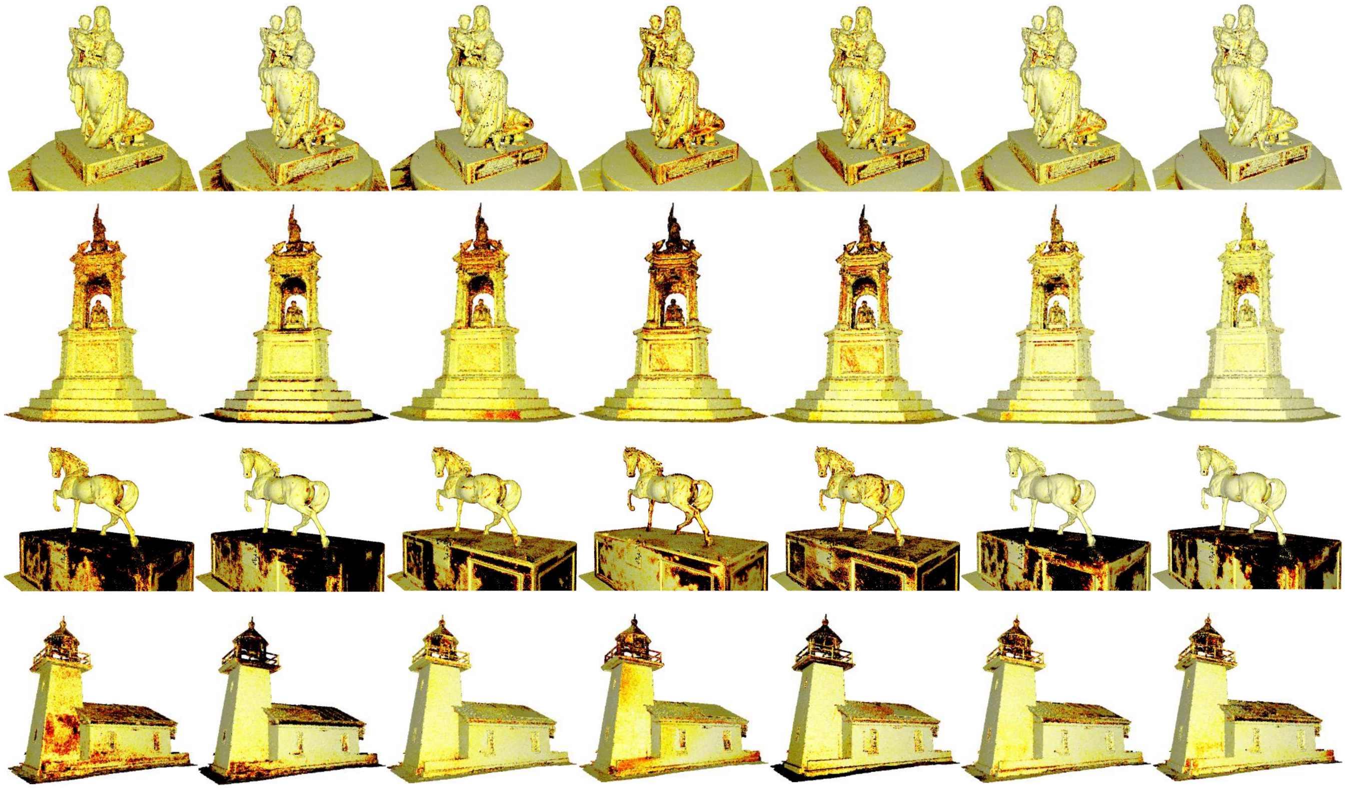
Tanks and Temples Benchmark Different from DTU dataset acquired under well-controlled laboratory environment, Tanks and Temples [23] is a large-scale benchmark captured in more complex realistic situations. Specifically, Tanks and Temples dataset is divided into two subsets, namely an $intermediate$ set and an $advanced$ set. The $intermediate$ set consists of 8 different scenes with varying scales, surfaces and conditions, which is widely used to evaluate the general performance of 3D reconstruction algorithms. The $advanced$ set includes 6 complex architectural scenes that traditional MVS methods often have difficulties to deal with.

TABLE 2: Quantitative results on DTU evaluation set [22] (lower is better). Our method outperforms all previous state-of-the-art methods in term of *overall* quality.

Method	Mean Distance (mm)		
	Acc.	Comp.	*overall*
COLMAP [5]	0.400	0.664	0.532
Gipuma [4]	**0.283**	0.873	0.578
MVSNet [8]	0.396	0.527	0.462
R-MVSNet [9]	0.385	0.459	0.422
P-MVSNet [19]	0.406	0.434	0.420
PointMVSNet [10]	0.361	0.421	0.391
PointMVSNet-HiRes [10]	0.342	0.411	0.376
Vis-MVSNet [14]	0.369	0.361	0.365
PVA-MVSNet [20]	0.379	0.336	0.357
CasMVSNet [11]	0.325	0.385	0.355
PatchMatchNet [15]	0.427	**0.277**	0.352
CVP-MVSNet [12]	0.296	0.406	0.351
UCS-Net [26]	0.338	0.349	0.344
D^2HC-RMVSNet [21]	0.395	0.378	0.386
BH-RMVSNet (ours)	0.368	0.303	**0.335**

We first evaluate our method on the $intermediate$ set of Tanks and Temples [23]. To further boost the performance, we fine-tune the proposed network on BlendedMVS training set before benchmarking. As denoted in Tab. 3, BH-RMVSNet outperforms all existing methods with a significant margin in terms of mean F-$score$, which is the harmonic average of $precision$ and $recall$. It is noteworthy that even without the fine-tuning step on BlendedMVS dataset, our method still performs very competitively. The error visualization of different scenes provided by the online benchmark [35] is shown in Fig. 7. We further validate the generalization ability of our method on the $advanced$ set. Benefiting from the proposed recurrent processing strategy, our method successfully reconstructs the large-scale scenes, and achieves the best performance over all existing learning-based methods, as is demonstrated in Tab. 4.

BlendedMVS Dataset BlendedMVS [24] is a large-scale MVS dataset which contains a variety of challenging scenes



(a) R-MVSNet (b) Point-MVSNet (c) P-MVSNet (d) DeepC-MVS (e) AttMVS (f) D^{2}HC-RMVSNet (g) **BH-RMVSNet**

Fig. 7: Error Visualization of $Family$, $Francis$, $Horse$ and $Lighthouse$ in the Tanks and Temples benchmark [23] calculated according to the corresponding ground truth point clouds, in contrast to the recent advanced MVS methods [9], [10], [13], [19], [21], [36]. Darker regions represent larger errors.

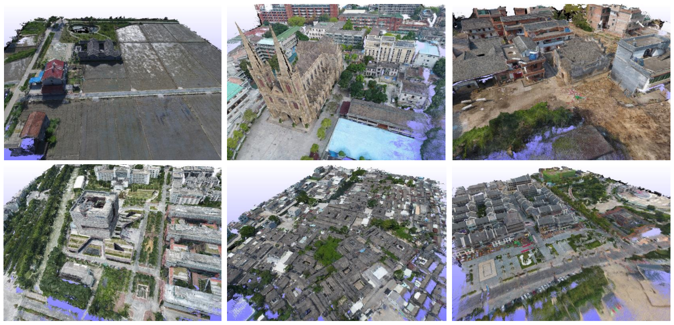


Fig. 8: Reconstruction results of the wide-range aerial scenes of BlendedMVS dataset [24], which demonstrate the scalability and generalizability of our method.

with high-resolution images. Since this dataset does not officially provide evaluation tools, we mainly utilize the aerial scenes of BlendedMVS dataset for qualitative evaluation. The reconstruction results shown in Fig. 8 indicate that our method is able to recover the wide-range real-world scenes, exhibiting strong scalability and robustness.

ETH3D Benchmark We also test our method on ETH3D benchmark [25], which contains much more challenging indoor and outdoor scenes with texture-less regions and varying numbers of viewpoints. We set the resolution of input images as 1536×1024 for evaluation. The quantitative results on the test set are shown in Tab. 5 and our method achieves competitive results in contrast to previous MVS methods. Specifically, BH-RMVSNet outperforms learning-based methods PVSNet [39] and PatchmatchNet [15] in terms of $F_1 score$, and achieves much better completeness compared to traditional MVS algorithms. The visualized comparisons of different methods are shown in Fig. 9. It can be seen that the reconstructed point clouds by our method keep more details for some challenging regions with fewer outliers. These results further demonstrate the generalization capability of our method under complicated scenarios.

5.3 Ablation Studies

In this section, we conduct ablation experiments to analyze the strengths of different components of our method. The following ablation studies are performed on the $evaluation$ set of DTU [22] using the same parameters as Sec. 5.1.

We apply D^2HC-RMVSNet [21] as our baseline method which utilizes DRENet for feature extraction and unidirectional HU-LSTM for cost regularization. Tab. 6 is an overview of ablation studies in terms of all proposed modules and techniques in BH-RMVSNet. As is indicated, the model with full settings achieves the best performance, in which the bidirectional cost volume regularization and the visibility-based depth map refinement play major roles. Our multi-scale feature extraction and depth map fusion methods also improve the overall quality, when compared to the origin extractor DRENet and fusion strategy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3165860, IEEE Transactions on Visualization and Computer Graphics

TABLE 3: Quantitative results on the *intermediate* set of Tanks and Temples benchmark [23] with the evaluation metric *F-score* (higher is better). **Bold** figures indicate the best scores, underlined ones indicate the second best. Our method ranks 1^{st} with mean *F-score* of 61.96 after fine-tuning on BlendedMVS [24].

Method	**Mean**	Family	Francis	Horse	Lighthouse	M60	Panther	Playground	Train
COLMAP [5]	42.14	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
MVSNet [8]	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
CIDER [34]	46.76	56.79	32.39	29.89	54.67	53.46	53.51	50.48	42.85
Point-MVSNet [10]	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
Dense R-MVSNet [9]	50.55	73.01	54.46	43.42	43.88	46.80	46.69	50.87	45.25
PatchMatchNet [15]	53.15	66.99	52.64	43.24	54.87	52.87	49.54	54.21	50.81
CVP-MVSNet [12]	54.03	76.50	47.74	36.34	55.12	57.28	54.28	57.43	47.54
PVA-MVSNet [20]	54.46	69.36	46.80	46.01	55.74	57.23	54.75	56.70	49.06
UCS-Net [26]	54.83	76.09	53.16	43.03	54.00	55.60	51.49	57.38	47.89
OpenMVS [37]	55.11	71.69	51.12	42.76	58.98	54.72	56.17	59.77	45.69
P-MVSNet [19]	55.62	70.04	44.64	40.22	65.20	55.08	55.17	60.37	54.29
CasMVSNet [11]	56.84	76.37	58.45	46.26	55.81	56.11	54.06	58.18	49.51
ACMM [6]	57.27	69.24	51.45	46.97	63.20	55.07	57.64	60.08	54.48
ACMP [7]	58.41	70.30	54.06	**54.11**	61.65	54.16	57.60	58.12	57.25
DeepC-MVS [36]	59.79	71.91	54.08	42.29	**66.54**	55.77	**67.47**	60.47	**59.83**
Vis-MVSNet [14]	60.03	77.40	60.23	47.07	63.44	62.21	57.28	60.54	52.07
AttMVS [13]	60.05	73.90	62.58	44.08	64.88	56.08	59.39	**63.42**	56.06
D^2HC-RMVSNet [21]	59.20	74.69	56.04	49.42	60.08	59.81	59.61	60.04	53.92
BH-RMVSNet (not fine-tuned)	61.14	77.74	59.36	49.16	63.77	63.11	60.78	59.87	55.29
BH-RMVSNet (fine-tuned)	**61.96**	**78.62**	**62.73**	51.21	62.13	**63.59**	61.09	60.85	55.50

TABLE 4: Quantitative results on the *advanced* set of Tanks and Temples benchmark [23] (higher is better), where many advanced MVS methods refrain to be evaluated. **Bold** figures indicate the best scores, underlined ones indicate the second best. Our method achieves the best performance over all learning-based methods. Note that ACMP [7] is a traditional MVS method that introduces plane prior for assistance.

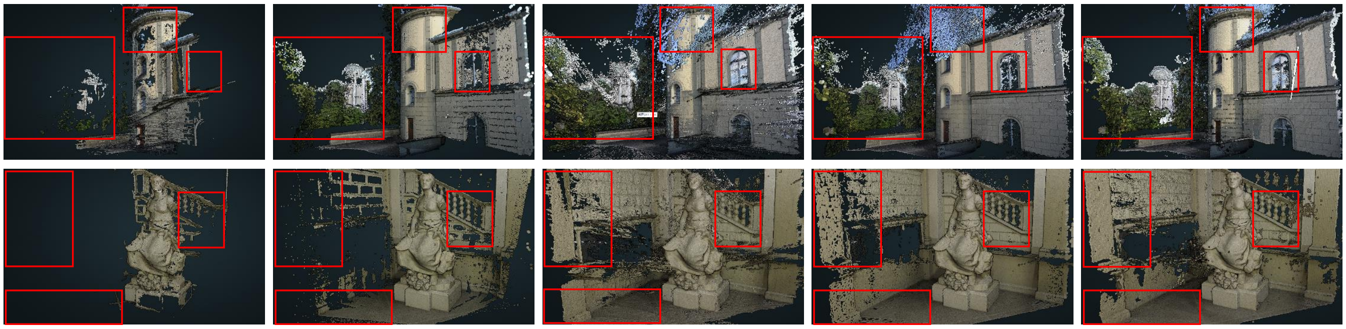
Method	**Mean**	Auditorium	Ballroom	Courtroom	Museum	Palace	Temple
CIDER [34]	23.12	12.77	24.94	25.01	33.64	19.18	23.15
COLMAP [5]	27.24	16.02	25.23	34.70	41.51	18.05	27.94
Dense R-MVSNet [9]	29.55	19.49	31.45	29.99	42.31	22.94	31.10
CasMVSNet [11]	31.12	19.81	38.46	29.10	43.87	27.36	28.11
IDCF [38]	32.28	23.66	33.01	36.05	42.10	23.95	34.92
AttMVS [13]	31.93	15.96	27.71	37.99	**52.01**	29.07	28.84
PatchmatchNet [15]	32.31	23.69	37.73	30.04	41.80	28.31	32.29
ACMM [6]	34.02	23.41	32.91	41.17	48.13	23.87	34.60
DeepC-MVS [36]	34.54	26.30	34.66	43.50	45.66	23.09	34.00
ACMP [7]	**37.44**	**30.12**	34.68	**44.58**	50.64	27.20	**37.43**
BH-RMVSNet (not fine-tuned)	32.72	24.94	36.90	36.33	41.14	26.97	30.07
BH-RMVSNet (fine-tuned)	34.81	25.79	**40.09**	34.50	44.89	**29.08**	34.51

Bidirectional Processing As described in Sec. 3.2, we design the bidirectional recurrent regularization scheme to aggregate full-space context information across all depth hypotheses, which enhances the unidirectional MVS networks to predict more delicate and accurate depth maps. Fig. 10(a) summarizes the descent of mean absolute depth error on DTU *training* dataset [22] with or without bidirectional processing during training, and demonstrates that the model with bidirectional RNN yields better depth estimations. As for testing, the mean absolute depth error drops from 6.23 to 5.30 on DTU *evaluation* set [22] with bidirectional processing, which contributes to the improvement of the *overall* quality by 10.6%, as denoted in Tab. 6.

Depth Map Refinement As mentioned in Sec. 4.1, based on the proposed visibility-based depth map refinement approach, the multi-view depth maps are updated iteratively to approach a robust overall depth estimation. Fig. 10(b) illustrates that the mean absolute depth errors for testing on DTU dataset [22] vary with iterations. On average, each iteration takes about extra 0.2s per image. For efficiency considerations, we terminate the refinement process at the 5-th iteration and take the results as the final outputs. It is also shown in Fig. 10(b) that the utilization of visibility and ZNCC enhances the process of iterative convergence. Besides, since depth refinement is a general step in many depth based MVS methods, we also test the proposed depth map refinement on other methods, whose evaluation results on DTU are shown in Fig. 11. It can be seen that, the depth maps estimated by [8], [20], [21] can also benefit from our depth map refinement to obtain more accurate and complete point clouds. As for testing on Tanks and Temples benchmark [23], the proposed refinement module contributes to an increase of 0.3 in mean *F-score* for BH-RMVSNet, as well as for D^2HC-RMVSNet [21].

Experimental Settings We further conduct ablation experiments on the important experimental settings of our method, such as the number of input views N, the number of depth hypotheses D and the threshold of confidence for depth fusion ξ. As shown in Tab. 7, our model performs comprehensively better when N increases within 7 views. Tab. 8 indicates that a finer division of space, by a larger D, helps to obtain better *accuracy*. However, dividing too

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVCG.2022.3165860, IEEE Transactions on Visualization and Computer Graphics



(a) MVE (b) COLMAP (c) PatchMatchNet (d) EPP-MVSNet (e) **BH-RMVSNet**

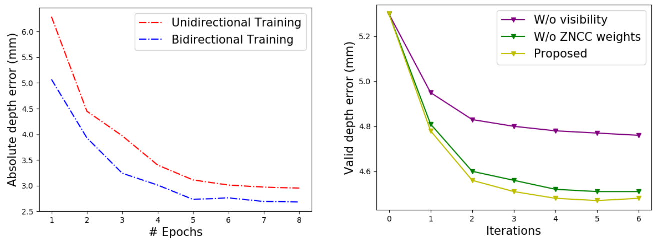
Fig. 9: Visualized comparisons of different methods for the *observatory* and *statue* of ETH3D benchmarks [25]. The point clouds reconstructed by our method are competitive compared to PatchMatchNet [15] and EPP-MVSNet [40]. Our method also exhibits much better completeness than traditional MVS algorithms [5], [41].

TABLE 5: Evaluation on high-resolution multi-view scans of ETH3D benchmark [25] by the metric of $F_1 score$ (larger is better) with different error thresholds. Our method achieves competitive results in contrast to previous MVS methods on this challenging benchmark.

Method	$F_1 score$			$Accuracy$			$Completeness$		
	τ=2 cm	τ=5 cm	τ=10 cm	τ=2 cm	τ=5 cm	τ=10 cm	τ=2 cm	τ=5 cm	τ=10 cm
MVE [41]	30.37	43.39	53.25	51.82	73.86	86.48	24.35	33.85	41.70
Gipuma [4]	45.18	57.99	67.86	84.44	95.31	98.07	34.91	45.11	54.77
COLMAP [5]	73.01	83.96	90.40	**91.97**	**96.75**	**98.25**	62.98	75.74	84.54
PVSNet [39]	72.08	85.55	92.24	66.41	82.89	91.10	80.05	89.02	93.69
PatchMatchNet [15]	73.12	85.85	91.91	69.71	85.22	91.98	77.46	86.83	92.05
EPP-MVSNet [40]	83.40	91.70	95.22	85.47	93.91	96.84	81.79	89.76	93.75
DeepC-MVS [36]	**87.08**	**93.31**	**96.06**	89.15	95.43	97.82	**85.52**	**91.53**	**94.56**
BH-RMVSNet (not fine-tuned)	78.21	87.57	92.14	76.57	87.21	92.14	80.49	88.24	92.32
BH-RMVSNet (fine-tuned)	79.61	88.37	92.72	80.53	90.47	94.56	79.46	86.84	91.25

TABLE 6: Ablation experiments on different experimental settings. ”MFE” refers to multi-scale feature extraction; ”BCR” refers to bidirectional cost regularization; ”VDR” refers visibility-based depth refinement; ”DMF” refers to depth map fusion.

Settings				Mean Distance (mm)		
MFE	BCR	VDR	DMF	Acc.	Comp.	*overall*
✗	✗	✗	✗	0.408	0.374	0.391
✓	✗	✗	✗	0.396	0.378	0.387
✓	✓	✗	✗	0.371	0.321	0.346
✓	✓	✓	✗	**0.361**	0.314	0.338
✓	✓	✓	✓	0.368	**0.303**	**0.335**



(a) (b)

Fig. 10: (a) Validation results of the mean absolute depth error during training. (b) Comparisons of the valid depth (confidence> 0.4) error by different refinement strategies.

many depth layers will lead to the loss in *completeness*. Therefore, we choose $N = 7$ and $D = 512$ to conduct experiments. As for the confidence threshold ξ, Tab. 9 illustrates that a larger threshold leads to better *accuracy* but worse *completeness* and a smaller value works in the opposite manner. As a balanced choice, 0.4 leads to the best *overall* performance.

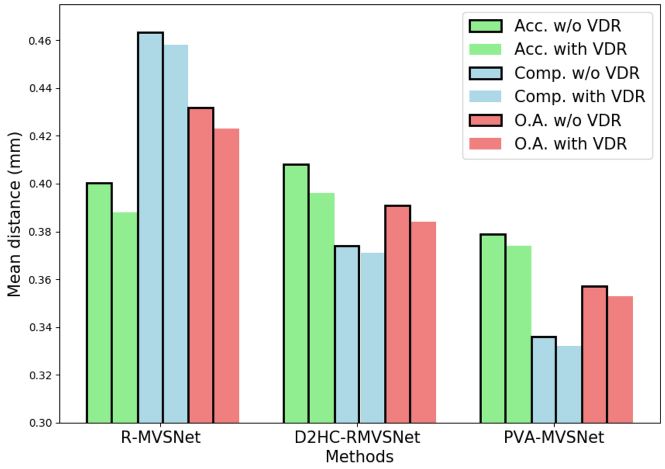


Fig. 11: Comparisons of the evaluation results on DTU dataset [22] with or without the proposed visibility-based depth refinement (VDR) for other methods.

Runtime & Memory We also compare our BH-RMVSNet with previous methods in terms of runtime and memory, as denoted in Tab. 10. Benefiting from the recurrent processing strategy as well as R-MVSNet [9] and D^2HC-RMVSNet [21], our method is much more memory efficient to generate

TABLE 7: Ablation experiments on the number of input views N.

N	Acc. (mm)	Comp. (mm)	*Overall* (mm)
3	0.413	0.384	0.398
5	0.371	0.320	0.346
7	**0.368**	0.303	**0.335**
9	0.386	**0.296**	0.341
11	0.394	0.303	0.349

TABLE 8: Ablation experiments on the number of depth hypotheses D.

D	Acc. (mm)	Comp. (mm)	*Overall* (mm)
256	0.389	**0.301**	0.345
512	0.368	0.303	**0.335**
1024	**0.355**	0.333	0.344

TABLE 9: Ablation experiments on the confidence threshold ξ.

ξ	Acc. (mm)	Comp. (mm)	*Overall* (mm)
0.3	0.385	**0.297**	0.341
0.4	0.368	0.303	**0.335**
0.5	**0.358**	0.314	0.336

TABLE 10: Comparisons of the runtime and memory consumption on DTU dataset [22].

Method	Depth Size	Mem.(GB)	Time(s)	O.A.
MVSNet [8]	400 × 296	15.4	1.18	0.462
PVA-MVSNet [20]	800 × 592	24.87	1.01	0.357
R-MVSNet [9]	400 × 296	6.7	2.35	0.422
D^2HC-RMVSNet [21]	800 × 592	2.4	8.0	0.386
Vis-MVSNet [14]	640 × 480	8.7	3.35	0.391
CasMVSNet [11]	1152 × 864	5.35	**0.49**	0.355
CVP-MVSNet [12]	1600 × 1152	8.80	1.72	0.351
BH-RMVSNet (ours)	480 × 360	**1.66**	11.7	0.359
BH-RMVSNet (ours)	800 × 600	3.46	32.2	**0.335**
BH-RMVSNet (ours)	1600 × 1200	10.45	104.1	0.343

high-resolution depth maps compared to the original 3D CNN based methods [8], [20]. Specifically, our method takes only 13.9% memory of PVA-MVSNet [20] but achieves a better $overall$ reconstruction quality. In contrast to the recent cascade 3D CNN methods such as CasMVSNet [11], CVP-MVSNet [12] and Vis-MVSNet [14], our method also takes less memory and achieves a better *overall* score. Compared with the prototype method D^2HC-RMVSNet [21], our method requires extra memory in 3D to store regularized cost maps in at least one direction, and thus involves extra computation.

5.4 Limitations

On the one hand, RNN-based MVS methods suffer from great time consumption for sequentially processing depth layers in exchange for low memory cost. To be specific, when the number of images, image resolution and depth layer division are very large, there will be a bottleneck in time efficiency for our method. On the other hand, though our LSTM network takes only 2D memory for computation and delete the intermediate cost maps, some additional memory is needed to save the regularized cost maps for later integration. This limits the ability to obtain very high-resolution depth maps.

6 Conclusion and Future Work

We have presented a novel hybrid LSTM based recurrent MVS network with bidirectional cost volume regularization, namely BH-RMVSNet. The proposed network combines the advantages of 3D CNNs and recurrent processing, dramatically reducing the runtime memory and fully considering the context information. To further enhance the accuracy and robustness for point cloud reconstruction, we refine the predicted depth maps according to the visible pixels of their neighbor views. Experimental results show that our method achieves excellent performance on DTU, Tanks and Temples and ETH3D datasets, exhibiting strong generalizability and scalability with a relatively low demand of runtime memory.

Since RNNs trade time for space, BH-RMVSNet, like other RNN-based MVS networks, also suffers from slow inference speed. Inspired by the cascade methods, our future work will focus on adapting RNNs into a coarse-to-fine multi-stage fashion. Another potential solution is to adopt a coarse depth map to lower the value of D right before the construction of cost volumes. This coarse depth map could be estimated by a lightweight monocular depth estimation network or be generated from sparse SfM points by a depth completion network.

Acknowledgments

This research is supported by National Key Technology Research and Development Program of China, grant number 2017YFB1002601; National Natural Science Foundation of China (NSFC), grant number 61632003.

References

[1] 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, 2009.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[3] 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.

[4] S. Galliani, K. Lasinger, and K. Schindler, “Massively parallel multiview stereopsis by surface normal diffusion,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 873–881.

[5] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, “Pixelwise view selection for unstructured multi-view stereo,” in *European Conference on Computer Vision*. Springer, 2016, pp. 501–518.

[6] Q. Xu and W. Tao, “Multi-scale geometric consistency guided multi-view stereo,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5483–5492.

[7] ——, “Planar prior assisted patchmatch multi-view stereo,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 516–12 523.

[8] 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)*, 2018, pp. 767–783.

[9] 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*, 2019, pp. 5525–5534.

[10] 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.

[11] 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.

[12] 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.

[13] 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*, 2020, pp. 1590–1599.

[14] J. Zhang, Y. Yao, S. Li, Z. Luo, and T. Fang, "Visibility-aware multi-view stereo network," *British Machine Vision Conference (BMVC)*, 2020.

[15] F. Wang, S. Galliani, C. Vogel, P. Speciale, and M. Pollefeys, "Patchmatchnet: Learned multi-view patchmatch stereo," 2020.

[16] M. Ji, J. Gall, H. Zheng, Y. Liu, and L. Fang, "Surfacenet: An end-to-end 3d neural network for multiview stereopsis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2307–2315.

[17] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," *arXiv preprint arXiv:1708.05375*, 2017.

[18] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "Deepmvs: Learning multi-view stereopsis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2821–2830.

[19] 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*, 2019, pp. 10 452–10 461.

[20] H. Yi, Z. Wei, M. Ding, R. Zhang, Y. Chen, G. Wang, and Y.-W. Tai, "Pyramid multi-view stereo net with self-adaptive view aggregation," in *European Conference on Computer Vision*. Springer, 2020, pp. 766–782.

[21] 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.

[22] 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.

[23] 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.

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1790–1799.

[25] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2538–2547.

[26] 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*, 2020, pp. 2524–2534.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[28] Y. Wu and K. He, "Group normalization," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[29] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[30] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 3, pp. 1–13, 2013.

[31] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," *NeurIPS Autodiff Workshop*, 2017.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2014.

[34] 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. 07, 2020, pp. 12 508–12 515.

[35] "Tanks-and-temples," https://www.tanksandtemples.org/, Accessed on 3 Feb. 2021.

[36] A. Kuhn, C. Sormann, M. Rossi, O. Erdler, and F. Fraundorfer, "Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 404–413.

[37] "Openmvs," https://github.com/cdcseacave/openMVS, Accessed on 3 Feb. 2021.

[38] H. Liu, X. Tang, and S. Shen, "Depth-map completion for large indoor scene reconstruction," *Pattern Recognition*, vol. 99, p. 107112, 2020.

[39] Q. Xu and W. Tao, "Pvsnet: Pixelwise visibility-aware multi-view stereo network," *arXiv preprint arXiv:2007.07714*, 2020.

[40] 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*, 2021, pp. 5732–5740.

[41] S. Fuhrmann, F. Langguth, and M. Goesele, "Mve-a multi-view reconstruction environment." in *GCH*. Citeseer, 2014, pp. 11–18.

Zizhuang Wei received the B.S. degree from Peking University. He is currently a Ph.D candidate at Graphics and Interaction Lab, Dept. of EECS, Peking University. His research interests focus on 3D reconstruction, with a particular interest in multi-view stereo and point cloud alignment.

Qingtian Zhu received the B.Eng. degree from Beijing University of Posts and Telecommunications (BUPT) in 2020. He is currently a master student at Graphics and Interaction Lab (GIL) of Peking University. His research interests include 3D reconstruction and computational photogrammetry.

Chen Min received the B.S. degree from Xi'an Jiaotong University, China, in 2016 and the M.S. degree from Beijing Jiaotong University, in 2020, China. He is currently persuing his Ph.D degree in Peking University. His research interest is in 3D computer vision, with a particular interest in 3D perception for autonomous driving and 3D reconstruction.

Yisong Chen received the Ph.D degree at Nanjing University, majoring computer science. Now he is an associate professor in Graphics and Interaction Lab of Peking University. His research interests include digital image/video processing, computer graphics, computer vision, pattern recognition, machine learning and statistical analysis.

Guoping Wang received the bachelor and master degree from Dept. of Mathematics, Harbin Institute of Technology in 1987 and 1990 respectively and Ph.D degree from Institute of Mathematics, Fudan University in 1997. He was engaged in postdoctoral research in Tsinghua University from 1997 to 1999, and got full professor position in Peking University in 2002. He achieved the National Science Fund for Distinguished Young Scholars in 2009. His research interests include computer graphics, human-computer interaction and virtual reality.