Dense Hybrid Recurrent Multi-view Stereo Net with Dynamic Consistency Checking

Jianfeng Yan[1*], Zizhuang Wei[1*], Hongwei Yi[1*†], Mingyu Ding[2], Runze Zhang[3], Yisong Chen[1], Guoping Wang[1], and Yu-Wing Tai[4]

[1] Peking University
{haibao637, weizizhuang, hongweiyi, chenyisong, wgp}@pku.edu.cn
[2] HKU myding@cs.hku.hk
[3] Tencent ryanrzzhang@tencent.com
[4] Kwai Inc. yuwing@gmail.com

Abstract. In this paper, we propose an efficient and effective dense hybrid recurrent multi-view stereo net with dynamic consistency checking, namely D^2HC-RMVSNet, for accurate dense point cloud reconstruction. Our novel hybrid recurrent multi-view stereo net consists of two core modules: 1) a light DRENet (Dense Reception Expanded) module to extract dense feature maps of original size with multi-scale context information, 2) a HU-LSTM (Hybrid U-LSTM) to regularize 3D matching volume into predicted depth map, which efficiently aggregates different scale information by coupling LSTM and U-Net architecture. To further improve the accuracy and completeness of reconstructed point clouds, we leverage a dynamic consistency checking strategy instead of prefixed parameters and strategies widely adopted in existing methods for dense point cloud reconstruction. In doing so, we dynamically aggregate geometric consistency matching error among all the views. Our method ranks 1^{st} on the complex outdoor *Tanks and Temples* benchmark over all the methods. Extensive experiments on the in-door DTU dataset show our method exhibits competitive performance to the state-of-the-art method while dramatically reduces memory consumption, which costs only 19.4% of R-MVSNet memory consumption. The codebase is available at https://github.com/yhw-yhw/D2HC-RMVSNet.

Keywords: Multi-view Stereo, Deep Learning, Dense Hybrid Recurrent-MVSNet, Dynamic Consistency Checking

1 Introduction

Dense point cloud reconstruction from multi-view stereo (MVS) information is a classic and important Computer Vision problem for decades, where stereo correspondences of more than two calibrated images are used to recover dense 3D representation [24,20,28,29,32]. While traditional MVS methods have achieved

* Equal Contribution.
† Corresponding Author

promising results, the recent advance in deep learning [10,16,14,33,15,34,6,21] allows the exploration of implicit representations of multi-view stereo, hence resulting in superior completeness and accuracy in MVS benchmarks [5,19] compared with traditional alternatives without learning.

However, those deep learning based MVS methods still have the following problems. First, due to the memory limitation, some methods like MVSNet [33] cannot deal with images with large resolutions. Then, RMVSNet [34] are proposed to solve this problem, while the completeness and accuracy of reconstruction are compromised. Second, heavy backbones with downsampling module have to be used to extract features in [33,15,34,6,21], which rely on large memory and lose information in the downsampling process. At last, those deep learning based MVS methods have to fuse the depth maps obtained by different images. The fusion criteria are set in a heuristic pre-defined manner for all data-sets, which lead to low complete results.

To tackle the above problems, we propose a novel deep learning based MVS method called D^2HC-RMVSNet with a network architecture and a dynamic algorithm to fuse depth maps in the postprocessing. The network architecture consists of 1) a newly designed lightweight backbone to extract features for the dense depth map regression, 2) a hybrid module coupling LSTM and U-Net to regularize 3D matching volume into predicted depth maps with different level information into LSTM. The dynamic algorithm to fuse depth maps attempts to aggregate the matching consistency among all the neighbor views to dynamically remain accurate and more reliable dense points in the final results.

Our main contributions are listed below:

- We propose a new lightweight DRENet to extract dense feature map for dense point cloud reconstruction.
- We design a hybrid architecture DHU-LSTM which absorbs both the merits of LSTM and U-Net to reduce the memory cost while maintains the reconstruction accuracy.
- We design a non-trivial dynamic consistency checking algorithm for filtering to remain more reliable and accurate depth values and obtain more complete dense point clouds.
- Our method ranks 1^{st} on ***Tanks and Temples*** [19] among all methods and exhibits competitive performance to the state-of-the-art method on **DTU** [5] while dramatically reduces memory consumption.

2 Related Work

Deep neural network has made tremendous progress in many vision task [9,17,8,36], including several attempts on multi-view stereo. The deep learning based MVS methods [10,16,14,33,15,34,6,21,37] generally first use the backbones with some downsampling modules to extract features and the final layer of the backbones with the most downsampled feature maps are output to the following module. Hence, those methods cannot directly output depth maps with the same

resolution as the input images and may lose some information in those higher resolutions, which may influence the accuracy of reconstructed results.

Then, plane-sweep volumes are pre-wraped from images as the input to those networks [10,16,14]. The plane-sweep volumes are memory-consuming and those methods cannot be trained end-to-end. To train the neural network in an end-end fasion, MVSNet [33] and DPSNet [15] implicitly encodes multi-view camera geometries into the network to build the 3D cost volumes by introducing the differential homography warping. P-MVSNet [21] utilizes a patch-wise matching module to learn the isotropic matching confidence inside the cost volume. PointMVSNet [6] proposes a two-stage coarse-to-fine method to generate high resolution depth maps, where a coarse depth map is first yielded by the lower-resolution version MVSNet [33] and depth errors are iteratively refined in the point cloud format. However, this method is time-consuming and complicated to employ in real applications since it consists of two different network architectures. In addition, the memory-consuming 3D-CNN modules adopted in those methods limit their application for scalable 3D reconstruction from high resolution images. To reduce memory consumption during the inference phase, R-MVSNet [34] leverages the recurrent gated recurrent unit (GRU) instead of 3D-CNN, whereas compromises completeness and accuracy on 3D reconstruction.

All of the above methods have to fuse the depth maps from different reference images to obtain the final reconstructed dense point clouds by following the post-processing in the non-learning based MVS method COLMAP [25]. In the post-processing, consistency is checked in a pre-defined manner, which is not robust for different scenes and may miss some good points viewed by few images.

To improve the deep learning based MVS methods based on above analysis, we propose a light DRENet specifically designed for the dense depth reconstruction, which outputs the same feature map size as input images with large receptive fields. Then a HU-LSTM (Hybrid U-LSTM) module is designed to reduce the memory consumption while maintains the 3D reconstruction accuracy. At last, we design a dynamic consistency checking algorithm for filtering to obtain more accurate and complete dense point clouds.

3 Reconstruction Pipeline

Given a set of multi-view images and corresponding calibrated camera parameters calculated from Structure-from-Motion [26], our goal is to estimate the depth map of each reference image and reconstruct dense 3D point cloud. First, each input image is regarded as the reference image and fed to the effective Dense Hybrid Recurrent MVSNet (DH-RMVSNet) with several neighbor images to regress the corresponding dense depth map. Then, we use a dynamic consistency checking algorithm to filter all the estimated depth maps of multi-view images to obtain more accurate and reliable depth values, by leveraging geometric consistency through all neighbor views. After achieving dense filtered reliable depth maps, we directly re-project and fuse all pixels with reliable depth values into 3D space to generate corresponding dense 3D point clouds.

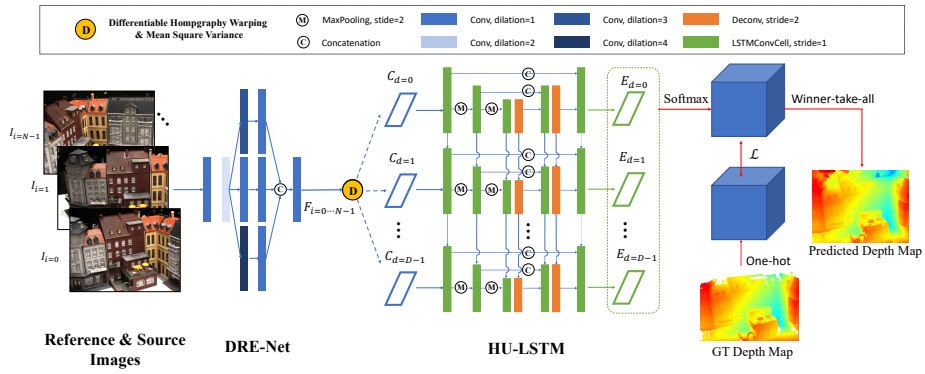

Fig. 1. The network architecture of DH-RMVSNet. 2D feature maps extracted from multi-view images by DRENet go through differentiable homography warping and mean square variance to generate 3D cost volumes. Our HU-LSTM processes 3D cost volume sequentially in depth direction for further training or depth prediction.

In the following sections, we first introduce our efficient DH-RMVSNet and novel dynamic consistency checking. Then we evaluate our method on DTU [5] and *Tanks and Temples* [19] to prove the efficacy of our method. To evaluate the practicality and generalization on the large-scale dataset with wide range for wide real application, we extend our method on aerial photos in the *BlendMVS* [35] dataset to reconstruct a large-scale scene.

4 Dense Hybrid Recurrent MVSNet

This section describes the details of our proposed network DH-RMVSNet as visualized in Fig. 1. We design a novel hybrid recurrent multi-view stereo network which absorbs both advantages of 3DCNN in MVSNet [33] and recurrent unit in R-MVSNet [34]. Specifically, our DH-RMVSNet leverages well both the accuracy of 3DCNN processing 3D dimension data and the efficiency of recurrent unit by sequentially processing. Therefore, our network can generate dense accurate depth maps and corresponding dense 3D reconstruction point clouds on the large-scale datasets. We first introduce our lightweight efficient image feature extractor DRENet in Sec. 4.1. Then we present HU-LSTM sub-network to sequentially regularize feature matching volumes in the depth hypothesis direction into 3D probability volume in Sec. 4.2. At last we introduce our training loss in Sec. 4.3.

4.1 Image Feature Extractor

We design a Dense Receptive Expansion sub-network by concatenating feature maps from different dilated convolutional [38] layers to aggregate multi-scale contextual information without losing resolution. We term it DRENet whose

Input	Layer Destription	Output	Output Size
Input multi-view image size: $N \times H \times W \times 3$			
DRENet			
$I_{i=0\cdots N-1}$	ConvGR, filter=3×3, stride=1	2D0_0	$H \times W \times 16$
2D0_0	ConvGR, filter=3×3, stride=1	2D0_1	$H \times W \times 16$
2D0_1	ConvGR, filter=3×3, stride=1, dilation=2	2D0_2	$H \times W \times 32$
2D0_2	ConvGR, filter=3×3, stride=1	2D0_3	$H \times W \times 32$
2D0_2	ConvGR, filter=3×3, stride=1, dilation=3	2D1_1	$H \times W \times 32$
2D1_1	ConvGR, filter=3×3, stride=1	2D1_2	$H \times W \times 32$
2D0_2	ConvGR, filter=3×3, stride=1, dilation=4	2D2_1	$H \times W \times 32$
2D2_1	ConvGR, filter=3×3, stride=1	2D2_2	$H \times W \times 32$
[2D0_3, 2D1_2, 2D2_2]	ConvGR, filter=3×3, stride=1	$F_{i=0\cdots N-1}$	$H \times W \times 32$
HU-LSTM			
$\mathcal{C}(i)$	ConvLSTMCell, filter=3×3	$\mathcal{C}_0(i)$	$H \times W \times 32$
$\mathcal{C}_0(i)$	MaxPooling, stride=2	$\mathcal{C}_0^{'}(i)$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
$\mathcal{C}_0^{'}(i)$ &$\mathcal{C}_2(i-1)$	ConvLSTMCell, filter=3×3	$\mathcal{C}_1(i)$	$H \times W \times 32$
$\mathcal{C}_1(i)$	MaxPooling, stride=2	$\mathcal{C}_1^{'}(i)$	$\frac{1}{4}H \times \frac{1}{4}W \times 32$
$\mathcal{C}_1^{'}(i)$ &$\mathcal{C}_2(i-1)$	ConvLSTMCell, filter=3×3	$\mathcal{C}_2(i)$	$H \times W \times 32$
$\mathcal{C}_2(i)$	DeConv, filter=3×3, stride=2	$\hat{\mathcal{C}}_2(i)$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
$[\mathcal{C}_1(i), \hat{\mathcal{C}}_2(i)[$ & $\mathcal{C}_3(i-1)$	ConvLSTMCell, filter=3×3	$\mathcal{C}_3(i)$	$H \times W \times 32$
$\mathcal{C}_3(i)$	DeConv, filter=3×3, stride=2	$\hat{\mathcal{C}}_3(i)$	$\frac{1}{2}H \times \frac{1}{2}W \times 32$
$[\mathcal{C}_1(i), \hat{\mathcal{C}}_3(i)]$& $\mathcal{C}_4(i-1)$	ConvLSTMCell, filter=3×3	$\mathcal{C}_4(i)$	$H \times W \times 32$
$\mathcal{C}_4(i)$	Conv, filter=3×3, stride=1	$\mathcal{C}_H(i)$	$H \times W \times 1$

Table 1. The details of our DH-RMVSNet architecture which consists of DRENet and HU-LSTM. Conv and Deconv denote 2D convolution and 2D deconvolution respectively, GR is the abbreviation of group normalization and the ReLU. MaxPooling represents 2D max-pooling layer and ConvLSTMCell represent LSTM recurrent cell with 2D convolution. N, H, W, D are input multi-view number, image height, width and depth hypothesis number.

weights are shared by multi-view images $\mathbf{I}_{i=0\cdots N-1}$. Most of previous multi-view stereo network, such as [33,34,21], usually use 2D convolutional layers with stride larger or equal than 2 to enlarge the receptive field and reduce the resolution at the same time for satisfying memory limitation. We introduce different dilated convolutional layers to generate multi-scale context information and preserve the resolution which leads to the possibility of dense depth map estimation. The details of DRENet are presented in Tab. 1.

Given N-view images, let $\mathbf{I}_{i=0}$ and $\mathbf{I}_{i=1\cdots N-1}$ denote the reference image and the neighbor source images respectively. We first use two usual convolutional layers to sum up local-wise pixel information, then we utilize three dilated convolutional layers with different dilated ratio $2, 3, 4$ to extract multi-scale context information without scarifying resolution. Thus, after concatenation, DRENet can extract the dense feature map $F_i \in \mathbb{R}^{C \times H \times W}$ efficiently, where C denotes the feature channel and H, W represent the height and width of the input image.

Following common practices [33,15,34,6,21], to build a 3D feature volume $\{\boldsymbol{V}_i\}_{i=0}^{N-1}$, we utilize the differentiable homography to warp the extracted feature map between different views. And we adopt the same mean square variance to aggregate them into one cost volume $\boldsymbol{C}$.

4.2 Hybrid Recurrent Regularization

There exists two different ways to regularize the cost volume $\boldsymbol{C}$ into one probability map $\mathcal{P}$. One is to utilize the 3DCNN U-Net in MVSNet [33] which can well leverage local wise information and multi-scale context information, but it can not directly be used to regress the original dense depth map estimation due to limited GPU memory especially for large resolution images. The other is to use stacked convolutional GRU in R-MVSNet [34] which is quiet efficient by sequentially processing the 3D volume through the depth direction but loss the aggregation of multi-scale context information.

Therefore, we absorb the merits in both two methods to propose a hybrid recurrent regularization network with more powerful recurrent convolutional cell than GRU, namely LSTMConvCell [31]. We construct a hybrid U-LSTM which is a novel 2D U-net architecture where each layer is LSTMConvCell, which can be processed sequentially. We term this module HU-LSTM. Our HU-LSTM can well aggregate multi-scale context information and easily process dense original size cost volumes with high efficiency at the same time. It costs 19.4% GPU memory of the previous recurrent method R-MVSNet [34]. The detailed architecture of HU-LSTM is demonstrated in Tab. 1.

Cost volume $\boldsymbol{C}$ can be viewed as D number 2D cost matching map $\{\mathcal{C}(i)\}_{i=0}^{D-1}$ which are concatenated in the depth hypothesis direction. We denote the output of regularized cost matching map as $\{\mathcal{C}_H(i)\}_{i=0}^{D-1}$ at i^{th} step during sequential processing. Therefore, $\mathcal{C}_H(i)$ relies on the both current input cost matching map $\mathcal{C}(i)$ and all previous states $\mathcal{C}_H(0,\cdots,i-1)$. Different from GRU in R-MVSNet [34], we introduce more powerful recurrent unit named ConvLSTMCell which has three gates map to control the information flow and can well aggregate different scale context information.

Let $\mathbb{I}(i)$, $\mathbb{F}(i)$ and $\mathbb{O}(i)$ denote the input gate map, forget gate map and output gate map respectively. In the following part, $\odot$, '[]' and '*' represent the element-wise multiplication, the concatenation and the matrix multiplication respectively in convolutional layer.

The input gate map is used to select valid information from current input $\hat{\mathcal{C}}(i)$ into the current state cell $\mathcal{C}(i)$:

$$\mathbb{I}(i) = \sigma(\mathbb{W}_{\mathbb{I}} * [\mathcal{C}(i), \mathcal{C}_H(i-1)] + \mathbb{B}_{\mathbb{I}}), \tag{1}$$

$$\hat{\mathcal{C}}(i) = tanh(\mathbb{W}_{\mathbb{C}} * [\mathcal{C}(i), \mathcal{C}_H(i-1)] + \mathbb{B}_{\mathbb{C}}), \tag{2}$$

while the forget gate map $\mathbb{F}(i)$ decides to filter useless information from previous state cell $\mathcal{C}(i-1)$ and combines the input information from the input gate map $\mathbb{I}(i)$ to generate current new state cell $\mathcal{C}(i)$:

$$\mathbb{F}(i) = \sigma(\mathbb{W}_{\mathbb{F}} * [\mathcal{C}(i), \mathcal{C}_H(i-1)] + \mathbb{B}_{\mathbb{F}}), \tag{3}$$

$$\mathcal{C}(i) = \mathbb{F}_i \odot \mathcal{C}_H(i-1) + \mathbb{I}_i \odot \hat{\mathcal{C}}(i), \tag{4}$$

Finally, the output gate map controls how much information from new current state cell $\mathcal{C}(i)$ will output, which is $\mathcal{C}_H(i)$:

$$\mathbb{O}(i) = \sigma(\mathbb{W}_{\mathbb{O}} * [\mathcal{C}(i), \mathcal{C}_H(i-1)] + \mathbb{B}_{\mathbb{O}}), \tag{5}$$

$$\mathcal{C}_H\left(i\right) = \mathbb{O}\left(i\right) \odot tanh(\mathcal{C}\left(i\right)), \tag{6}$$

where σ and *tanh* represent *sigmoid* and *tanh* non-linear activation function respectively, $\mathbb{W}$ and $\mathbb{B}$ are learnable parameters in LSTM convolutional filter.

In our proposed HU-LSTM, by aggregating different scale context information to improve the robustness and accuracy of depth estimation, we adopt three LSTMConvCells to propragate different scale input feature maps with downsampling scale 0.5 and two LSTMConvCell to aggregate multi-scale context information as denoted in Tab. 1. Specifically, we input the 32-channel input cost map $\mathcal{C}_i$ to the first LSTMConvCell, and the output of each LSTMConvCell will be fed into next LSTMConvCell. Then the regularized cost matching volume $\{\mathcal{C}_H\left(i\right)\}_{i=0}^{D-1}$ goes through by a *softmax* layer to generate corresponding the probability volume $\mathcal{P}$ for further caculating training loss.

4.3 Training Loss

Following MVSNet [33], we treat the depth regression task as multiple classification task and use the same cross entropy loss function $\mathcal{L}$ between the probability volumes $\mathcal{P}$ and ground truth depth map $\mathcal{G}$:

$$\mathcal{L} = \sum_{\boldsymbol{x} \in \boldsymbol{x}_{valid}} \sum_{i=0}^{D-1} -G(i,x) * log(P(i,x)), \tag{7}$$

where $\boldsymbol{x}_{valid}$ is the set of valid pixels in the ground truth, $G(i,x)$ represents the one-hot vector generated by the depth value of the ground truth $\mathcal{G}$ at pixel x and $P(i,x)$ is the corresponding depth estimated probability. During test phase, we do not need to save the whole probability map. To further improve the efficiency, the depth map is processed sequentially and the winner-take-all selection is used to generate the estimated depth map from regularized cost matching volume.

5 Dynamic Consistency Checking

The above DH-RMVSNet generates dense pixel-wise depth map for each input multi-view images. Before fusing all the estimated multi-view depth maps, it is necessary to filter out mismatched errors and store correct and reliable depths. All previous methods [33,34,6,21] just follow [27] to apply the geometric constraint to measure the depth estimation consistency among multiple views. However, those methods only use prefixed constant parameters. Specifically, the reliable depth value should satisfy both conditions: the pixel reprojection error less than τ_1 and and the depth reprojection error less than τ_2 in at least three views, where $\tau_1 = 1$ and $\tau_2 = 0.01$ are pre-defined. These parameters are defined intuitively and not robust for different scenes, though they have large influence on the quality of reconstruct point cloud. For example, those depth values with much high reliable consistency in two views are filtered and a fixed number valid views also lose information in the views with slightly worse errors. Beside, using the fixed τ_1 and τ_2 may not filter enough mismatched pixels in different scenes.

In general, a estimated depth value is accurate and reliable when it has a very low reprojection error in few views, or a lower error in majority views. Therefore, we propose a novel dynamic consistency checking algorithm to select valid depth values, which is related to both the reprojection error and view numbers. By considering dynamic geometric matching cost as consistency among all neighbor views, it leads to more robust and complete dense 3D point clouds. We denote the estimated depth value $D_i(\boldsymbol{p})$ of a pixel $\boldsymbol{p}$ on reference image $\mathbf{I}_i$ through our DH-RMVSNET. The camera parameter is represented by $\boldsymbol{P}_i = [\boldsymbol{M}_i|\boldsymbol{t}_i]$ in [13]. First we back-project the pixel p into 3D space to generate the corresponding 3D point $\boldsymbol{X}$ by:

$$\boldsymbol{X} = \boldsymbol{M}_i^{-1}(\boldsymbol{D}_i(\boldsymbol{p}) \cdot \boldsymbol{p} - \boldsymbol{t}_i), \tag{8}$$

Then we project the 3D point $\boldsymbol{X}$ to generate the projected pixel $\boldsymbol{q}$ on the neighbor view $\mathbf{I}_j$:

$$\boldsymbol{q} = \frac{1}{d}\boldsymbol{P}_j \cdot \boldsymbol{X}, \tag{9}$$

where $\boldsymbol{P}_j$ is the camera parameter of neighbor view $\mathbf{I}_j$ and d is the depth from projection. In turn, we back-project the projected pixel $\boldsymbol{q}$ with estimated depth $D_j(\boldsymbol{q})$ on the neighbor view into 3D space and reproject back to the reference image denoted as $\boldsymbol{p}'$:

$$\boldsymbol{p}' = \frac{1}{d'}\boldsymbol{P}_j \cdot (\boldsymbol{M}_j^{-1}(D_j(\boldsymbol{q}) \cdot \boldsymbol{q} - \boldsymbol{t}_j)), \tag{10}$$

where d' is the depth value of the reprojected pixel $\boldsymbol{p}'$ on the reference image. Based on the above mentioned operation, the reprojection errors are calculated by:

$$\begin{aligned} \xi_p &= ||\boldsymbol{p} - \boldsymbol{p}'||_2, \\ \xi_d &= ||D_i(\boldsymbol{p}) - d'||_1 / D_i(\boldsymbol{p}). \end{aligned} \tag{11}$$

In order to quantify the depth matching consistency between two different views, we propose the dynamic matching consistency by considering dynamic matching consistency among all views. The dynamic matching consistency in different views is defined as:

$$c_{ij}(\boldsymbol{p}) = e^{-(\xi_p + \lambda \cdot \xi_d)}, \tag{12}$$

where λ is used to leverage the reprojection error in two different metrics. By aggregating the matching consistency from all the neighbor views to obtain the global dynamic multi-view geometric consistency $C_{geo}(\boldsymbol{p})$ as:

$$C_{geo}(\boldsymbol{p}) = \sum_{j=1}^{N} c_{ij}. \tag{13}$$

We calculate the dynamic geometric consistency for every pixel and filter out the outliers with $C_{geo}(\boldsymbol{p}) < \tau$. Benefiting from our proposed dynamic consistency checking algorithm, the filtered depth map is able to store more accurate and complete depth values compared with the previous intuitive fixed-threshold method. It improves the robustness, completeness and accuracy of 3D reconstructed point clouds.

6 Experiments

6.1 Implementation Details

Training We train DH-RMVSNet on the DTU dataset [5], which contains 124 different indoor scenes which is split to three parts, namely *training*, *validation* and *evaluation*. Following common practices [14,16,33,34,6,37,7], we train our network on the training dataset and evaluate on the evaluation dataset. While the dataset only provides ground truth point clouds generated by scanners, to generate the ground truth depth map, we use the same rendering method as MVSNet [33]. Different from [33,34] which generate the depth map with $\frac{1}{4}$ size of original input image, our method generates the depth map with the same size as input image. Since MVSNet [33] only provides $\frac{1}{4}$ size depth map, thus we resize the training input image to $W \times H = 160 \times 128$ as same as the corresponding groundtruth depth map. We set the number of input images $N = 3$ and the depth hypotheses are sampled from $425mm$ to $745mm$ with depth plane number $D = 128$ in MVSNet [33]. We implement our network on **PyTorch** [23] and train the network end-to-end for 6 epochs using *Adam* [18] with an initial learning rate 0.001 which is decayed by 0.9 every epoch. Batch size is set to 6 on 2 NVIDIA TITAN RTX graphics cards.

Testing. For testing, we use the $N = 7$ views as input, and set $D = 256$ for depth plane hypothesis in an inverse depth manner in [34]. To evaluate *Tanks and Temples* dataset, the camera parameters are computed by OpenMVG [22] following MVSNet [33] and the input image resolution is set to 1920×1056. We test the BlendedMVS [35] dataset using original images of 768×576 resolution.

Filter & Fusion After we generate estimated depth maps from DH-RMVSNet, we filter and fuse them to generate corresponding 3D dense point cloud. First, the depths with probability lower than $\phi = 0.4$ will be discarded. Then, we use our proposed dynamic global geometric consistency checking algorithm as further multi-view depth map filter with $\lambda = 200$ and $\tau = 1.8$. At last, we fuse all reliable depths into 3D space to generate 3D point cloud.

6.2 Datasets and Results

We first demonstrate the state-of-the-art performance of our proposed D^2HC-RMVSNet on the DTU [5] and *Tanks and Temples* [19], which outperforms its original methods, namely MVSNet [33] and R-MVSNet [34] with a significant margin. Specifically, our method ranks 1^{st} in the complex large-scale outdoor *Tanks and Temples* benchmark over all existing methods. To investigate the practicality and scalability of our method, we extend our method on the aerial photos in *BlendedMVS* [35] to reconstruct a larger scale scenes.

Method	Mean Distance (mm)		
	Acc.	Comp.	*overall*
Tola [30]	0.342	1.190	0.766
Gipuma [11]	**0.283**	0.873	0.578
Colmap [25]	0.400	0.664	0.532
SurfaceNet [16]	0.450	1.040	0.745
MVSNet [33]	0.396	0.527	0.462
R-MVSNet [34]	0.385	0.459	0.422
P-MVSNet [21]	0.406	0.434	0.420
PointMVSNet [6]	0.361	0.421	0.391
PointMVSNet-HiRes [6]	0.342	0.411	**0.376**
***D*^{2}HC-RMVSNet**	0.395	**0.378**	0.386

Table 2. Quantitative results on the DTU evaluation dataset [5] (lower is better). Our method D^2HC-RMVSNet exhibits a competitive reconstruction performance compared with state-of-the-art methods in terms of completeness and overall quality.

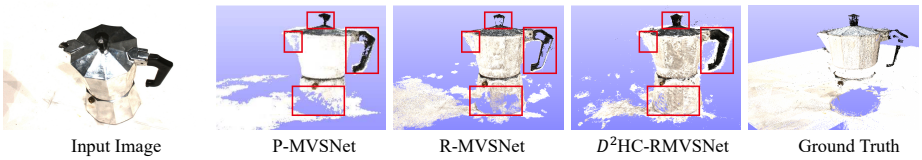

Fig. 2. Comparison on the reconstructed point clouds for the *Scan* 77 from the DTU [5] dataset with other methods [21,34]. Our method generates more complete and denser point cloud than other methods.

DTU Dataset. We evaluate our proposed method on the DTU [5] *evaluation* dataset. We set $D = 256$ within the depth range [425mm, 905mm] for all scans and use the common evaluation metric in other methods [33,34]. Quantitative results are shown in Tab. 2. While Gipuma [11] performs the best regarding to accuracy, our method achieves the best completeness and the competitive *overall* quality of reconstruction results. Our proposed D^2HC-RMVSNet can both improve the accuracy and the completeness significantly compared with its original methods MVSNet [33] and R-MVSNet [34]. We also compare the results on the reconstructed point clouds with [21,34]. As shown in Fig. 2, our method generates more complete and accurate point cloud than other methods. It proves the efficacy of our novel DH-RMVSNet and dynamic consistency checking algorithm.

Tanks and Temples Benchmark. *Tanks and Temples Benchmark* [19] is a large-scale outdoor dataset which consists of more complex environment, and it is quite typical for real captured situation, compared with DTU dataset which is taken under well-controlled environment with fixed camera trajectory.

We evaluate our method **without any fine-tuning** on the *Tanks and Temples* as denoted in Tab. 3. Our proposed D^2HC-RMVSNet ranks 1^{st} over all existing methods. Specifically, our method outperforms all deep-learning based multi-view stereo methods with a big margin. It shows the stronger generalization

Method	Rank	Mean	Family	Francis	Horse	L.H.	M60	Panther	P.G.	Train
COLMAP [26,27]	55.62	42.14	50.41	22.25	25.63	56.43	44.83	46.97	48.53	42.04
Pix4D [4]	53.38	43.24	64.45	31.91	26.43	54.41	50.58	35.37	47.78	34.96
MVSNet [33]	52.75	43.48	55.99	28.55	25.07	50.79	53.96	50.86	47.90	34.69
Point-MVSNet [6]	40.25	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06
Dense R-MVSNet [34]	37.50	50.55	73.01	54.46	43.42	43.88	46.80	46.69	50.87	45.25
OpenMVS [3]	17.88	55.11	71.69	51.12	42.76	58.98	54.72	56.17	59.77	45.69
P-MVSNet [21]	17.00	55.62	70.04	44.64	40.22	**65.20**	55.08	55.17	60.37	54.29
CasMVSNet [12]	14.00	56.84	76.37	58.45	46.26	55.81	56.11	54.06	58.18	49.51
ACMM [32]	12.62	57.27	69.24	51.45	46.97	63.20	55.07	57.64	**60.08**	**54.48**
Altizure-HKUST-2019 [2]	9.12	59.03	**77.19**	**61.52**	42.09	63.50	59.36	58.20	57.05	53.30
DH-RMVSNet	10.62	57.55	73.62	53.17	46.24	58.68	59.38	58.31	58.26	52.77
***D*^{2}HC-RMVSNet**	**5.62**	**59.20**	74.69	56.04	**49.42**	60.08	**59.81**	**59.61**	60.04	53.92

Table 3. Quantitative results on the *Tanks and Temples* benchmark [19]. The evaluation metric is *f-score* which higher is better. (L.H. and P.G. are the abbreviations of *Lighthouse* and *Playground* dataset respectively.)

compared with Point-MVSNet [6] while Point-MVSNet [6] is the state-of-the-art method on the DTU [5]. The mean *f-score* increases significantly from 50.55 to 59.20 (larger is better, date: Mar. 5, 2020) compared with Dense R-MVSNet [34], which demonstrates the efficacy and robustness of D^2HC-RMVSNet on the variant scenes. The reconstructed point clouds are shown in Fig. 3, it shows that our method generates accurate, delicate and complete point cloud. And we compare the Precision / Recall of the model *Family* with its original methods [33,34] at different error threshold in Fig. 4. It demonstrates our method achieves a significant improvement on the precision while maintains better recall than R-MVSNet [34], which leads to the best performance on the *Tanks and Temples*.

BlendedMVS. BlendedMVS is a new large-scale MVS dataset which is synthesized from 3D reconstructed models from Altizure [2]. The dataset contains over 113 different scenes with a variety of different camera trajectories. And each scenes consists of 20 to 1000 input images including architectures, sculptures and small objects. To further evaluate the practicality and scalability of our propose D^2HC-RMVSNet, we directly test our method and R-MVSNet [34] on the provided *validation* dataset. For fair comparison, both methods are trained on the DTU [5] **without any fine-tuning** and we upsample the $\frac{1}{4}$ depth map from R-MVSNet to the same size of the depth map from our method, which is the original size of input images. As shown in Fig. 5, our method can well reconstruct the whole large scale scene and small cars in it, while R-MVSNet [34] fails on it. Our method can estimate the dense delicate accurate depth map with original size of input image in an inverse depth setting as in [34], because of our novel DRENet and HU-LSTM, which has more accuracy and stronger scalability of 3D point cloud reconstruction by aggregating multi-scale context information on the large-scale dataset. Due to our dynamical consistency checking algorithm, we can directly use our algorithm to remain dense reliable point cloud without any specific adjustment.

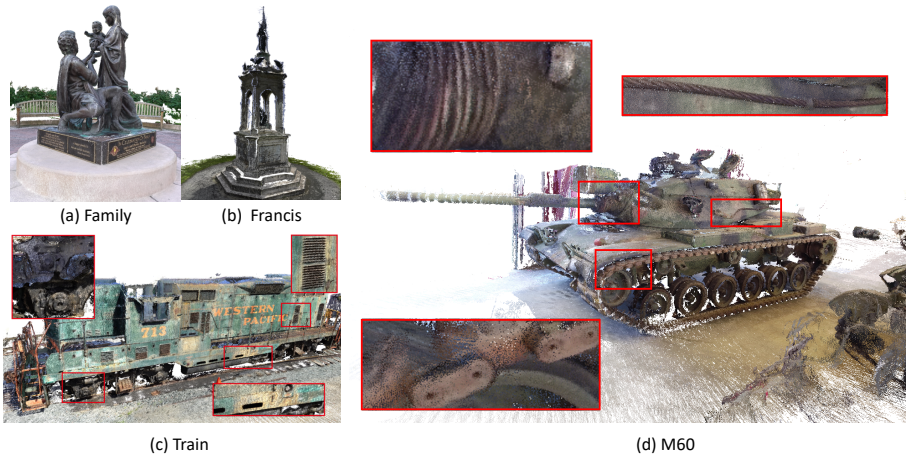

Fig. 3. Point cloud results on the *Tanks and Temples* [19] benchmark, our method generates accurate, delicate and complete reconstructed point clouds which show the strong generalization of our method on complex outdoor scenes.

6.3 Ablation Study

In this section, we provide ablation experiments to analyze the strengths of the key components of our architecture. we perform the following studies on DTU *validation* dataset with same setting in Sec. 6.1.

Variant components of network architecture To quantitatively analyze how different network architecture in DH-RMVSNet affect the depth map reconstruction, we evaluate the average mean absolute error between estimated depth maps and the ground truth on the *validation* DTU dataset during training. The comparison results are illustrated in Fig. 6.

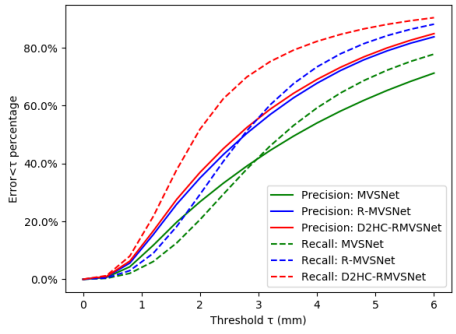

Fig. 4. Comparison on the Precision / Recall (in%) at different thresholds (within 6mm) on the *Family* provided by [1] with MVSNet [33] and R-MVSNet [34].

Fig. 5. Comparison of the reconstruction point cloud results on the *validation* set of *BlendedMVS* [35]. Our method can both well reconstruct a large-scale scene and small cars in it, while R-MVSNet [34] failed on it. *(Both methods are without fine-tuning.)*

Fig. 6. Validation results of the mean average depth error with different network architectures during training.

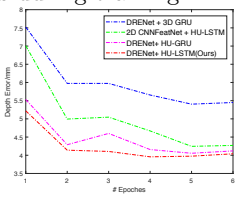

Table 4. Comparison of the running time and memory consumption between our proposed D^2HC-RMVSNet and R-MVSNet [34] on the DTU [5].

Method	Input Size	Output Size	Mem. (GB)	Time (s)
R-MVSNet	1600x1196	400x296	6.7	**2.1**
Ours	400x296	400x296	**1.3**	2.6
Ours	800x600	800x592	2.4	8.0
Ours	1600x1200	1600x1196	6.6	29.15

We replace our "DRENet" and "HU-LSTM" with "2DCNNFeatNet" and "3D GRU" in R-MVSNet [34] respectively to analysis the influence of our proposed feature encoder and cost volume regularization module. Compared with "2DCNNFeatNet + HU-LSTM" in Fig. 6, our proposed "DRENet" can improve the accuracy slightly but with less inference time and memory consumption due to the light architecture. "HU-LSTM" achieves significant improvement with a big margin compared with "DRENet + 3D GRU", which absorbs both the merits of efficacy in [33] and efficiency in [34] by aggregating multi-scale context information in sequential process. To further evaluate the difference in two different powerful recurrent gate units, LSTM and GRU, we replace the LSTM cell in our "HU-LSTM" with "GRU", denoted as "HU-GRU". The comparison between "HU-LSTM" and "HU-GRU" shows "LSTM" is more accurate and robust than "GRU" because of more gate map to control the information flow and have a better performance on the learning matching patterns.

Benefit from Dynamic Consistency Checking To further study the influence and generalization of our dynamic consistency checking, we evaluate our DH-RMVSNet with common filtering algorithm as in previous methods [33,34,6] as shown in Tab. 3. Our proposed dynamic consistency checking algorithm significantly boosts the reconstruction results in all scenes on the *Tanks and Temples*

benchmark, which shows the strong generalization and dynamic adaptation on the different scenes. It improves the *f-score* from 57.55 by DH-RMVSNet to 59.20, which leads to more accurate and complete reconstruction point clouds.

7 Discussion

Running Time & Memory Utility For fair comparison on the running time and memory utility with R-MVSNet [34], we test our method with same depth sample number $D = 256$ on the GTX 1080Ti GPU. As shown in Tab. 4, our method inputs multi-images of only 400×296 resolution to generate the depth map of the same size as R-MVSNet [34], with only 19.4% memory consumption of R-MVSNet. Moreover, our method runs with $2.6s$ per view, with a little extra inference time than R-MVSNet, which needs an extra $6.2s$ refinement to enhance the performance. Our D^2HC-RMVSNet achieves significant improvement over R-MVSNet [34] both on the DTU and the *Tanks and Temples* benchmark, while our novel dynamic consistency checking takes negligible running time. Our method can generate dense depth maps with the same size of the input image with efficient memory consumption. It takes only $6.6GB$ to process multi-view images with 1600×1200 resolution, which leads to a wide practicality for dense point cloud reconstruction.

Scalability and Generalization Due to our light DRENet and HU-LSTM, our method shows more powerful general scalability than R-MVSNet [34] on the dense reconstruction with wide range. Our method can easily extend to aerial photos for the reconstruction of the big scene architectures in Fig. 5 and generate denser, more accurate and complete 3D point cloud reconstruction due to the original size depth map estimation from our D^2HC-RMVSNet.

8 Conclusions

We have presented a novel dense hybrid recurrent multi-view stereo network with dynamic consistency checking, denoted as D^2HC-RMVSNet, for dense accurate point cloud reconstruction. Our DH-RMVSNet well absorbs both the merits of the accuracy of 3DCNN and the efficiency of Recurrent unit, to design a new lightweight feature extractor DRENet and hybrid recurrent regularization module HU-LSTM. To further improve the robustness and completeness of 3D point cloud reconstruction, we propose a no-trivial dynamic consistency checking algorithm to dynamically aggregate geometric matching error among all views rather than use prefixed strategy and parameters. Experimental results show that our method ranks 1^{st} on the complex outdoor *Tanks and Temples* and exhibits the competitive results on the *DTU* dataset, while dramatically reduces memory consumption, which costs only 19.4% of R-MVSNet memory consumption.

Acknowledgements This project was supported by the National Key R&D Program of China (No.2017YFB1002705, No.2017YFB1002601) and NSFC of China (No.61632003, No.61661146002, No.61872398).

References

1. https://www.tanksandtemples.org/
2. Altizure. https://www.altizure.com/
3. Openmvs. https://github.com/cdcseacave/openMVS
4. Pix4d. https://pix4d.com/
5. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. IJCV **120**(2), 153–168 (2016)
6. Chen, R., Han, S., Xu, J., Su, H.: Point-based multi-view stereo network. arXiv preprint arXiv:1908.04422 (2019)
7. Chen, R., Han, S., Xu, J., Su, H.: Point-based multi-view stereo network. In: ICCV (2019)
8. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: CVPR (2020)
9. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV (2015)
10. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: Deepstereo: Learning to predict new views from the world's imagery. In: CVPR (2016)
11. Galliani, S., Lasinger, K., Schindler, K.: Massively parallel multiview stereopsis by surface normal diffusion. In: ICCV (2015)
12. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. arXiv preprint arXiv:1912.06378 (2019)
13. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
14. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: CVPR (2018)
15. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: end-to-end deep plane sweep stereo. arXiv preprint arXiv:1905.00538 (2019)
16. Ji, M., Gall, J., Zheng, H., Liu, Y., Fang, L.: Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In: ICCV (2017)
17. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: ICCV (2017)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. TOG **36**(4), 78 (2017)
20. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. PAMI **27**(3), 418–433 (2005)
21. Luo, K., Guan, T., Ju, L., Huang, H., Luo, Y.: P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In: ICCV (2019)
22. Moulon, P., Monasse, P., Marlet, R., et al.: Openmvg. an open multiple view geometry library (2014)
23. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NeurIPS Autodiff Workshop (2017)
24. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)

25. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
26. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
27. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
28. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR (2006)
29. Strecha, C., Von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR (2008)
30. Tola, E., Strecha, C., Fua, P.: Efficient large-scale multi-view stereo for ultra high-resolution image sets. Machine Vision and Applications **23**(5), 903–920 (2012)
31. Xingjian, S., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NeurIPS. pp. 802–810 (2015)
32. Xu, Q., Tao, W.: Multi-scale geometric consistency guided multi-view stereo. In: CVPR (2019)
33. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: ECCV (2018)
34. Yao, Y., Luo, Z., Li, S., Shen, T., Fang, T., Quan, L.: Recurrent mvsnet for high-resolution multi-view stereo depth inference. In: CVPR (2019)
35. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. arXiv preprint arXiv:1911.10127 (2019)
36. Yi, H., Li, C., Cao, Q., Shen, X., Li, S., Wang, G., Tai, Y.W.: Mmface: A multi-metric regression network for unconstrained face reconstruction. In: CVPR (2019)
37. Yi, H., Wei, Z., Ding, M., Zhang, R., Chen, Y., Wang, G., Tai, Y.W.: Pyramid multi-view stereo net with self-adaptive view aggregation. arXiv preprint arXiv:1912.03001 (2019)
38. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)