# Non-local Recurrent Regularization Networks for Multi-view Stereo

# Qingshan Xu<sup>1\*</sup>, Martin R. Oswald<sup>2</sup>, Wenbing Tao<sup>1†</sup>, Marc Pollefeys<sup>2,4</sup> and Zhaopeng Cui<sup>3</sup>

<sup>1</sup>National Key Laboratory of Science and Technology on Multispectral Information Processing,
School of Artifical Intelligence and Automation, Huazhong University of Science and Technology, China

<sup>2</sup>Computer Vision and Geometry Group, ETH Zurich, Switzerland

<sup>3</sup>State Key Lab of CAD & CG, Zhejiang University, China

<sup>4</sup>Microsoft Artificial Intelligence and Mixed Reality Lab, Zurich, Switzerland

#### **Abstract**

In deep multi-view stereo networks, cost regularization is crucial to achieve accurate depth estimation. Since 3D cost volume filtering is usually memory-consuming, recurrent 2D cost map regularization has recently become popular and has shown great potential in reconstructing 3D models of different scales. However, existing recurrent methods only model the local dependencies in the depth domain, which greatly limits the capability of capturing the global scene context along the depth dimension. To tackle this limitation, we propose a novel non-local recurrent regularization network for multi-view stereo, named NR2-Net. Specifically, we design a depth attention module to capture non-local depth interactions within a sliding depth block. Then, the global scene context between different blocks is modeled in a gated recurrent manner. This way, the long-range dependencies along the depth dimension are captured to facilitate the cost regularization. Moreover, we design a dynamic depth map fusion strategy to improve the algorithm robustness. Our method achieves state-of-the-art reconstruction results on both DTU and Tanks and Temples datasets.

# Introduction

Multi-view stereo (MVS) has been a hot research topic in computer vision for decades. It aims to recover the 3D geometry of a scene from a set of images with known camera parameters. Nowadays, multi-view stereo reconstruction is typically decomposed into two separate steps: depth map estimation and fusion (Galliani, Lasinger, and Schindler 2015; Schönberger et al. 2016; Xu and Tao 2019, 2020a,b; Yao et al. 2018; Gu et al. 2020). For these two steps, accurate depth map estimation is often challenging due to a variety of real-world problems, *e.g.*, low-textured areas, thin structures, occlusions and reflective surfaces.

Recently, many learning-based MVS methods (Yao et al. 2018, 2019; Xu and Tao 2020a; Gu et al. 2020) have been presented to achieve competitive or better results than traditional methods that rely on optimization of photometric consistency (Furukawa and Ponce 2010; Schönberger et al. 2016). Inspired by plane-sweeping stereo (Collins 1996), most learning-based methods first construct multi-channel

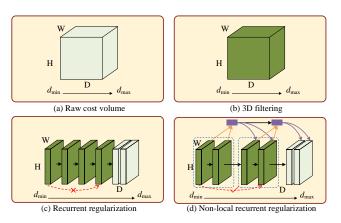


Figure 1: Illustration of different regularization methods. 3D filtering methods require cubic memory complexity while recurrent methods only have quadratic complexity. Our non-local recurrent regularization can explicitly establish interactions for non-adjacent depth values to capture more global context along the depth dimension which previous recurrent regularization methods cannot achieve.

cost volumes based on extracted deep image features, then perform deep cost volume processing to predict depth maps. The core of the successful learning-based MVS methods is the deep regularization of cost volumes to encode local and global information of a scene. The deep regularization converts the cost volume into a latent probability volume, directly influencing the final depth inference.

At present, the deep regularization of cost volumes can be categorized into two types: 3D cost volume filtering and recurrent 2D cost map regularization. The former one regularizes the entire 3D cost volume in one go, thus 3D convolutions can be seamlessly utilized to incorporate the context of a scene (Yao et al. 2018; Xu and Tao 2020a; Luo et al. 2019). In general, this kind of method is memory-consuming since its memory requirement is cubic to the model resolution (cf. Fig 1(b)). Although many follow-up works (Chen et al. 2019; Gu et al. 2020; Cheng et al. 2020; Yang et al. 2020) have been proposed to alleviate this problem, the scalability of 3D cost volume filtering is still limited when tackling high-resolution and large-scale 3D reconstruction. As another line of research, the latter one (Yao et al. 2019; Yan

<sup>\*</sup>Work done while Qingshan Xu was a visiting PhD student at ETH Zurich.

<sup>&</sup>lt;sup>†</sup>Corresponding author.

et al. 2020) sequentially regularizes 2D cost maps along the depth direction. Therefore, this kind of method reduces the memory requirement from cubic to quadratic to the model resolution. Moreover, such methods can adaptively sample sufficient depth hypotheses for scenes of different scales. In order to aggregate spatial as well as temporal context information in the depth direction, recurrent networks, *e.g.*, gated recurrent units (GRU) and long short-term memory (LSTM), are adopted in such methods. However, these methods only explicitly consider the information interaction between adjacent depth values, hence the long-range dependencies along the depth dimension cannot be fully captured (cf. Fig 1(c)).

To tackle the above limitation of recurrent 2D cost map regularization, we propose a novel non-local recurrent regularization network for multi-view stereo, namely NR2-Net. Built upon the regular recurrent neural networks, we divide the depth sampling space into different blocks to model the non-local interactions for non-adjacent depth planes. Within each block, we design a depth attention module to distill latent high-level features. This module samples cost map features at every other depth planes to enable large receptive fields like dilated convolution (Chen et al. 2018). Based on these cost map features, the latent high-level features are distilled via attention mechanism. Then the high-level features between blocks are further interacted in a gated recurrent manner to capture global scene context, which is used to regularize the bottom-level cost map features in the next block. In this way, the long-range dependencies along the depth dimension are modeled and the global scene context is perceived to help the long-range cost map regularization (cf. Fig 1(d)). At last, in order to fuse depth maps into point clouds, existing learning-based methods usually first predefine a constant depth probability threshold to filter out unreliable depth estimates. However, due to the uncertainty of depth prediction networks (Kendall and Gal 2017), this constant threshold will discard many credible depth values, resulting in point clouds of low completeness. Thus, we adaptively associate the depth probability threshold with depth consistency to generate more complete point clouds.

Our contributions can be summarized as follows: 1) We present a **novel non-local recurrent regularization** framework for multi-view stereo. This allows to perceive more global context information in the depth direction to assist cost volume regularization. 2) We propose a **depth attention module** to distill latent high-level features along the depth dimension. This helps to model non-local interactions for non-adjacent depth hypotheses in the gated recurrent manner. 3) We develop a **dynamic depth map fusion** strategy to reconstruct point clouds. This strategy jointly considers the depth probability and depth consistency in a dynamic way, which is robust for different scenes. Our method, NR2-Net, achieves state-of-the-art performance on both DTU and Tanks and Temples datasets.

# **Related Work**

**Traditional multi-view stereo.** In order to estimate depth maps for all input images, multi-view stereo needs to search correspondences across different images. Plane-sweeping methods (Collins 1996; Gallup et al. 2007) sample depth

plane hypotheses in the 3D scene. Then, they employ handcrafted similarity metrics, e.g., sum of absolute differences (SAD) and normalized cross correlation (NCC), to construct cost volumes and extract the final depth maps via the winnertake-all strategy. Since these similarity metrics are ambiguous to some challenging areas, e.g., low-textured areas and reflective surfaces, some methods adopt engineered regularization technologies, such as graph-cuts (Kolmogorov and Zabih 2002) and cost filtering (Hosni et al. 2013), to alleviate this problem. On the other hand, PatchMatch MVS methods (Zheng et al. 2014; Galliani, Lasinger, and Schindler 2015; Schönberger et al. 2016; Xu and Tao 2019, 2020b) adopt the sampling and propagation strategy (Barnes et al. 2009) to efficiently search continuous depth plane hypotheses from the whole depth interval. These methods impose implicit smoothness constraints based on the plane propagation in the 3D scene. Although they have greatly improved the performance of traditional methods, how to handle the ambiguity in the challenging areas is still an open problem. Learning-based multi-view stereo. Recently, some works have leveraged deep neural networks to learn deep similarity features and deep regularization for multi-view stereo (Hartmann et al. 2017; Ji et al. 2017; Kar, Häne, and Malik 2017). MVSNet (Yao et al. 2018) and DeepMVS (Huang et al. 2018) propose to construct cost volumes to learn depth maps for each input image. This makes learning-based MVS methods more scalable and applicable to scene reconstruction. To infer depth maps, MVSNet and many following works (Xue et al. 2019; Luo et al. 2019; Xu and Tao 2020a) utilize multi-scale 3D convolutions to regularize cost volumes. However, due to the limitation of cost volume resolution, these methods still cannot tackle high-resolution images. To predict high-resolution depth maps, cascade methods (Cheng et al. 2020; Gu et al. 2020; Yang et al. 2020) employ the coarse-to-fine framework to infer high-resolution estimation via thin cost volumes. However, these cascade methods heavily rely on coarse-scale estimation which often lacks important detail information. Consequently, this may result in blurred effects in high-resolution estimation. To achieve high-resolution prediction, R-MVSNet (Yao et al. 2019) recurrently regularizes cost maps through GRU cells. This dramatically reduces memory consumption. However, the generic GRU cells do not allow the regularization to consider enough context information. D<sup>2</sup>HC-RMVSNet (Yan et al. 2020) absorbs both the merits of LSTM and 2D U-Nets to present a 2D U-Net architecture with convolutional LSTM cells. This architecture considers more context information in image domain and can directly operate on highresolution image features. In contrast to our work, existing recurrent methods only explicitly consider interactions between adjacent depth planes and thus lack global context along the depth dimension.

#### Method

The network architecture of our method, NR2-Net, is depicted in Fig. 2. The input to our network is a reference image  $I_0$  and N-1 source images  $\{I_i\}_{i=1}^{N-1}$ , where N is the total number of input images. Their camera intrin-

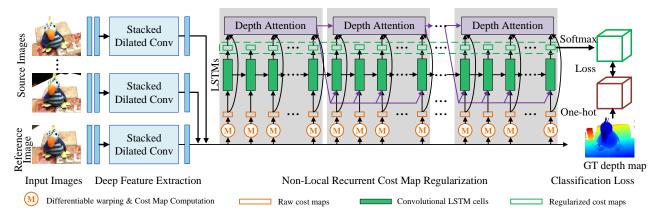


Figure 2: The NR2-Net architecture. Our network first extracts high-resolution deep features for all input images. Then cost maps at different depth planes are computed by differentiable homography warping and regularized by convolutional LSTM cells. To do non-local recurrent cost map regularization, the whole depth planes are divided into different blocks. The non-local interactions within each block are modeled by a depth attention module to distill latent high-level features. These high-level features are further interacted in a gated recurrent manner to capture global context information, which is used to help regularize the hidden states of LSTM cells in the next depth block.

sic parameters  $\{\mathbf{K}_i\}_{i=0}^{N-1}$  and relative extrinsic parameters  $\{\mathbf{R}_i,\mathbf{t}_i\}_{i=0}^{N-1}$  are known. Our NR2-Net first extracts highresolution deep features for all input images. The cost maps at different depth planes are computed by homography warping. Our non-local recurrent regularization framework is built upon convolutional LSTM cells to regularize 2D cost maps. To achieve non-local interactions for non-adjacent depth planes, we divide the whole depth interval into different blocks. The different depth planes within each block are interacted by a depth attention module to distill latent high-level features. The global context information is captured by further interacting the high-level features in a gated recurrent manner. Then, the global context information is in turn used to regularize the hidden states of different LSTM cells in the next block. This helps each LSTM cell perceive more global context information in the depth direction. Finally, we convert these hidden states into regularized cost maps to predict the depth map of the reference image.

# **Cost Map Construction**

Similar to previous works, we use the plane-sweeping algorithm (Collins 1996) to construct cost maps at different depth planes. Specifically, after extracting high-resolution deep image features  $\{F_i\}_{i=0}^{N-1}$  for all input images by stacked dilated convolutions (Yan et al. 2020), we compute the warped image features  $\{\tilde{F}_i(d)\}_{i=1}^{N-1}$  for N-1 source images at depth value d by differentiable homography warping. Then the 2D cost map at depth value d is calculated by:

$$\boldsymbol{C}(d) = \frac{\sum_{i=1}^{N-1} (1 + \boldsymbol{w}_i(d)) \odot (\tilde{\boldsymbol{F}}_i(d) - \boldsymbol{F}_0)^2}{N-1}, \quad (1)$$

where  $\boldsymbol{w}_i(d)$  is the view weight of the *i*-th source image and ' $\odot$ ' represents element-wise multiplication. Herein,  $\boldsymbol{w}_i(d)$  is computed by applying one convolution layer followed by group normalization and rectified linear units, a residual

block (He et al. 2016) and one convolution layer followed by a sigmoid function on the feature difference,  $\tilde{\boldsymbol{F}}_i(d) - \boldsymbol{F}_0$ . In this way, a cost volume  $\{\boldsymbol{C}(d)\}_{d=0}^{D-1}$  can be obtained by concatenating 2D cost maps of all depth planes in the depth direction, where D is the total number of depth planes.

# **Non-local Recurrent Regularization**

As the core of learning-based MVS methods, cost volume regularization is important to aggregate context information in both spatial and depth domain. Existing recurrent 2D cost map regularization methods cast the whole cost volume as a series of 2D cost maps in the depth direction and sequentially regularize 2D cost maps via stacked recurrent neural networks. Although such methods are memory-friendly, they only explicitly consider the local interactions between adjacent depth values.

Inspired by non-local methods (Wang et al. 2018; Fu et al. 2019), we design a non-local recurrent regularization network to model long-range dependencies along the depth dimension, which helps to regularize cost volumes by aggregating more context information. As illustrated in Fig. 2, we divide the whole cost volume into T blocks, each block contains s cost maps. Each cost map is regularized by stacked convolutional LSTM cells to incorporate context information in both the spatial and the depth domain. As depicted in Fig. 3(a), our stacked convolutional LSTM cells are built upon 2D U-Net architecture to aggregate multi-scale context information in the spatial domain. Moreover, these LSTM cells contain one non-local LSTM cell and four vanilla LSTM cells. This non-local LSTM cell is beneficial to incorporate more context information in the depth direction.

Next we elaborate on the mechanism of our non-local LSTM cell. As illustrated in Fig. 3(b), besides using the current cost map C(d), the previous regularized cost map  $C_r(d-1)$  and the previous cell state map C(d-1) as input like the vanilla LSTM cells, the non-local LSTM cell

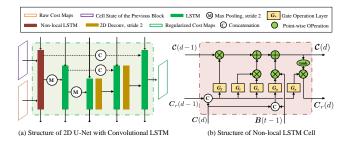


Figure 3: Illustrations of (a) structure of 2D U-Net with convolutional LSTM and (b) structure of non-local LSTM cell.

also incorporates the cell state map of the previous block,  $\boldsymbol{B}(t-1)$ , to consider long-range dependencies in the depth direction. How to obtain  $\boldsymbol{B}(t-1)$  through the depth attention module will be detailed in the next section. With these inputs, the forget gate map  $\mathbf{F}(d)$ , candidate state map  $\hat{\boldsymbol{C}}(d)$ , input gate map  $\mathbf{I}(d)$  and output gate map  $\mathbf{O}(d)$  are respectively modeled as:

$$\mathbf{F}(d) = \operatorname{sigmoid}(\mathbf{W}_f * [\mathbf{C}(d), \mathbf{C}_r(d-1)] + \mathbf{b}_f), (2)$$

$$\mathbf{I}(d) = \operatorname{sigmoid}(\mathbf{W}_i * [\mathbf{C}(d), \mathbf{C}_r(d-1)] + \mathbf{b}_i), \quad (3)$$

$$\hat{\boldsymbol{\mathcal{C}}}(d) = \tanh(\mathbf{W}_c * [\boldsymbol{C}(d), \boldsymbol{C}_r(d-1)] + \mathbf{b}_c), \quad (4)$$

$$\mathbf{O}(d) = \operatorname{sigmoid}(\mathbf{W}_o * [\mathbf{C}(d), \mathbf{C}_r(d-1)] + \mathbf{b}_o), \quad (5)$$

where '\*' means convolution, '[]' means concatenation, W is a transformation matrix and b is a bias term. Then, the current cell state map C(d) and the current regularized cost map  $C_r(d)$  are computed as,

$$\mathbf{C}(d) = \mathbf{F}(d) \odot \mathbf{C}(d-1) + \mathbf{I}(d) \odot \hat{\mathbf{C}}(d) + \mathbf{A}(d) \odot \mathbf{B}(t-1),$$
(6)

$$C_r(d) = \mathbf{O}(d) \odot \tanh(\mathcal{C}(d)),$$
 (7)

where A(d) is the depth attention gate map given by

$$\mathbf{A}(d) = \operatorname{sigmoid}(\mathbf{W}_a * [\mathbf{C}(d), \mathbf{B}(t-1)] + \mathbf{b}_a). \tag{8}$$

Note that, the last term in the right side of Eq. (6) reflects the non-local interactions in the depth direction. This is the difference between our non-local LSTM cell and the vanilla LSTM cell. Finally, the regularized cost maps  $\{C_r(d)\}_{d=0}^{D-1}$  pass a softmax layer to produce the probability volume P. In this way, our regularization not only incorporates multiscale context information in the spatial domain, but also aggregates long-range dependencies in the depth direction.

### **Depth Attention Module**

In the previous section, the depth attention module plays an important role in distilling latent high-level features within each block. It captures discriminative depth context information for each block. Then, the high-level features between blocks are interacted in a gated recurrent manner to capture global depth context information. Finally, we can use this information to impose explicit long-range regularization constraints for non-adjacent depth values as described above.

The structure of the depth attention module is illustrated in Fig. 4. For each block, we only use every other

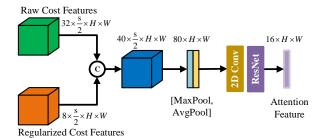


Figure 4: Structure of the depth attention module.

depth value in the block to sample raw cost features  $\boldsymbol{F}_{\text{raw}} \in \mathbb{R}^{32 imes \frac{s}{2} imes H imes W}$  and regularized cost features  $\boldsymbol{F}_{\text{reg}} \in$  $\mathbb{R}^{8 \times \frac{s}{2} \times H \times W}$ , where H and W denote the image height and width respectively. This enables the larger receptive fields in the depth direction. We first concatenate these two kinds of cost features into complex cost features  $F_{ ext{comp}} \in$  $\mathbb{R}^{40 \times \frac{s}{2} \times H \times W}.$  Such dense connection is good to explore their potential interactions and learn non-local information (Huang et al. 2017). Then we use max-pooling and average-pooling along the depth dimension to aggregate discriminative depth information, generating two different depth context features,  $F_{\rm max}$  and  $F_{\rm avg}$ . As pointed out in (Woo et al. 2018), we think that max-pooling is good to gather distinctive clues while average-pooling is beneficial to learn the extent of depth space. Using both can highlight the informative depth values as much as possible. These two features are further concatenated and forwarded into one convolution layer and a residual block to produce attention features  $\boldsymbol{F}_{\mathrm{att}} \in \mathbb{R}^{16 \times H \times W}$ 

To further model the global context information in the depth direction, the attention features between blocks are interacted by another recurrent neural network. Specifically, we reshape raw cost features  $\boldsymbol{F}_{\text{raw}} \in \mathbb{R}^{32 \times \frac{s}{2} \times H \times W}$  to  $\mathbb{R}^{C \times H \times W}$ , where  $C = 32 \times \frac{s}{2}$ . The input gate map  $\mathbf{G}_i(t)$  and the forget gate map  $\mathbf{G}_f(t)$  are modeled as

$$\mathbf{G}_{i}(t) = \operatorname{sigmoid}(\mathbf{W}_{ia} * [\mathbf{F}_{raw}(t), \mathbf{B}(t-1)] + \mathbf{b}_{ia}), \quad (9)$$

$$\mathbf{G}_f(t) = \operatorname{sigmoid}(\mathbf{W}_{fa} * [\mathbf{F}_{raw}(t), \mathbf{B}(t-1)] + \mathbf{b}_{fa}), (10)$$

where **W** is a transformation matrix and **b** is a bias term. Finally, we update the cell state map of the current block

$$\mathbf{B}(t) = \mathbf{G}_i(t) \odot \tanh(\mathbf{F}_{\text{att}}(t)) + \mathbf{G}_f(t) \odot \mathbf{B}(t-1).$$
 (11)

By updating the cell states of different blocks in a gated recurrent manner, our method is able to capture global context information in the depth direction. These cell states of blocks are in turn used to update the cell states of the LSTM cells.

#### **Loss Function**

Following previous practices (Yao et al. 2019; Yan et al. 2020), we cast the depth inference task as a multi-class classification problem. We use the following cross-entropy loss to train our network:

$$\mathcal{L} = \sum_{\boldsymbol{p} \in \Phi} \sum_{d=0}^{D-1} -\boldsymbol{G}(d, \boldsymbol{p}) \cdot \log(\boldsymbol{P}(d, \boldsymbol{p})), \quad (12)$$

where  $\Phi$  denotes the set of valid ground truth pixels,  $G(d, \mathbf{p})$  is the one-hot vector generated according to the ground truth depth map at pixel  $\mathbf{p}$ , and  $P(d, \mathbf{p})$  is the predicted depth probability at pixel  $\mathbf{p}$ . Note that, we do not need to store the whole classification probability volume during testing. Moreover, since the cell states of blocks are also sequentially updated, our designed depth attention module will not occupy too much memory.

### **Dynamic Depth Map Fusion**

After estimating depth maps for all input images, it is necessary to filter out wrong depth estimates for each depth map and then fuse them into a consistent point cloud representation. There are two critical factors for filtering wrong depth predictions: depth probability and depth consistency. The depth probability is usually measured by the corresponding probability of the selected depth, *i.e.*,  $\theta$ . The depth consistency is usually measured by the geometric constraint: reprojection errors and relative depth errors (Yao et al. 2018). For a pixel  $\mathbf{p}$  in the reference image  $I_0$ , we denote its reprojection error w.r.t the source image  $I_i$  by  $\psi_i$  and its relative depth error w.r.t.  $I_i$  by  $\phi_i$ . Then, its consistency view set is defined as:

$$S = \{ I_i | \psi_i < \epsilon, \phi_i < \eta \}, \tag{13}$$

where  $\epsilon$  and  $\eta$  are related thresholds. If  $|\mathcal{S}| > \mu$  and  $\theta > \tau$ , the depth estimate of  $\mathbf{p}$  is reliable, where  $\mu$  is the threshold of consistent view number and  $\tau$  is the probability threshold. Previous methods (Yao et al. 2018, 2019; Gu et al. 2020) usually set fixed threshold parameters to remove unreliable depth estimates. However, these intuitively preset parameters make their fusion methods not robust for different scenes. Thus,  $\mathrm{D}^2\mathrm{HC}\text{-RMVSNet}$  (Yan et al. 2020) proposes a dynamic consistency checking algorithm to measure depth consistency. Its dynamic consistency view set is defined as:

$$S_d = \{ \mathbf{I}_i | \psi_i < \epsilon(\mu), \phi_i < \eta(\mu), \epsilon(\mu) = \frac{\mu}{4}, \eta(\mu) = \frac{\mu}{1300} \},$$
(14)

The above definition makes the thresholds of geometric constraint be adaptively adjusted based on the consistent view number  $\mu$ . This means that the estimated depth value is accurate and reliable when satisfying strict depth consistency in a small number of views, or relaxed depth consistency in the majority of views. However, this method still adopts a fixed depth probability threshold to filter depths with different depth consistency. Due to the uncertainty of depth prediction networks (Kendall and Gal 2017), the depth probability will sometimes filter credible depth values that meet the strict depth consistency, making reconstructed point clouds less complete. Thus, the depth probability threshold should also be adjusted according to the level of depth consistency. Based on these observations and our experiments, we define the dynamic probability threshold as:

$$\tau(\mu) = 0.6 \cdot \exp\left[\frac{(\mu - 10)}{8}\right]. \tag{15}$$

By traversing different  $\mu$  values, the estimated depth of  ${\bf p}$  will be deemed accurate and reliable if  $|{\cal S}_d| > \mu$  and  $\theta > \tau(\mu)$ . That is, when the reprojection error and relative depth error are strict, the number of consistent view and

	Method	Acc.↓	Comp.↓	Overall↓
Geometric	Furu	0.613	0.941	0.777
	Tola	0.342	1.190	0.766
	Gipuma	0.283	0.873	0.578
	COLMAP	0.400	0.664	0.532
	SurfaceNet	0.450	1.040	0.745
Learning-based	MVSNet	0.396	0.527	0.462
	R-MVSNet	0.383	0.452	0.417
	CasMVSNet	0.325	0.385	0.355
	CVP-MVSNet	0.296	0.406	0.351
	UCSNet	0.338	0.349	0.344
	AttMVS	0.391	0.345	0.368
	D <sup>2</sup> HC-RMVSNet	0.395	0.378	0.386
	Ours	0.370	0.332	<u>0.351</u>

Table 1: Quantitative results on the DTU evaluation set using the distance metric [mm] (lower is better). Our method achieves the best completeness and the second best overall score. The best and second best results are in **bold** and <u>underlined</u>, respectively.



Figure 5: Qualitative results of reconstructed point clouds for the model scan 9 from the DTU evaluation set.

depth probability threshold are relaxed to judge if the depth prediction is reliable. Finally, the reliable depth estimates will be projected into 3D space to generate 3D point clouds.

# **Experiments**

We first describe datasets and implementation details. Then, we show benchmark results on the DTU dataset (Aanæs et al. 2016) as well as the Tanks and Temples dataset (Knapitsch et al. 2017). Finally, we conduct ablation studies to analyze our proposed core components.

# **Datasets and Evaluation Metrics**

**DTU** dataset. This dataset is captured from 49 or 64 views under a controlled indoor environment. 124 scenes are included with 7 different lighting conditions. The ground truth camera parameters and 3D point clouds are both provided. Following (Ji et al. 2017; Yao et al. 2018), we divide this dataset into training, validation and evaluation sets.

Tanks and Temples dataset. This dataset contains both indoor and outdoor scenes, which are captured in the realistic environment. Moreover, unlike the object-centric scenes in DTU, these scenes are larger and more complex. This dataset is divided into Intermediate set and Advanced set. The latter one is more challenging due to scale, textureless regions and complicated visibility.

**Evaluation metrics.** To evaluate the performance of our

	Method	Acc.[%] $\uparrow$	Comp.[%] $\uparrow$	$F_1[\%] \uparrow$
Geo.	COLMAP	43.16	44.48	42.14
	ACMM	49.19	70.85	57.27
	ACMP	49.06	<u>73.58</u>	58.41
	MVSNet	40.23	49.70	43.48
Learning-based	R-MVSNet	43.74	57.90	48.40
	CVP-MVSNet	51.41	60.19	54.03
	UCSNet	46.66	70.34	54.83
	CasMVSNet	53.71	63.88	56.84
	D <sup>2</sup> HC-RMVSNet	49.88	74.08	59.20
	AttMVS	61.89	58.93	60.05
	Ours	<u>55.72</u>	70.28	60.49

Table 2: Quantitative results on the Tanks and Temples Intermediate set. Our method achieves the best  $F_1$ -score. The best and second best results are in **bold** and underlined.



Figure 6: Qualitative results of reconstructed point clouds on Tanks and Temples dataset.

method on different datasets, distance metric and percentage metric are used for the DTU and Tanks and Temples dataset, respectively. For the distance metric, the overall score defined as the mean of accuracy and completeness measures the overall performance of reconstructed point clouds. For the percentage metric, the  $F_1$  score defined as the harmonic mean of accuracy and completeness is adopted.

# **Implementation Details**

**Training.** We train our network with ground truth depth maps on the DTU training set. The ground truth depth maps are generated through Poisson Surface Reconstruction (Kazhdan and Hoppe 2013). Due to the limitation of GPU memory, the input images are resized to  $H \times W = 128 \times 160$ . The input image number is N = 7. The depth values are sampled from 425mm to 905mm with D = 192 in an inverse depth manner. The block size s is set to s. Our network is implemented using Pytorch (Paszke et al. 2019) and trained with Adam (Kingma and Ba 2015) optimizer end-to-end for s0 epochs on two NVIDIA RTX 2080Ti GPU cards. The initial learning rate is set to s0.001 and the batch size is set to s2. Note that, to prevent estimated depth maps from being biased on the recurrent regularization order, we randomly use forward or backward pass to train samples.

**Evaluation.** During the evaluation, we only use the forward pass to infer depth maps. We sample D=512 depth planes in an inverse depth manner (Yao et al. 2019; Xu and Tao 2020a) and set the input image number N as 7. For the DTU dataset, the input image resolution is set to

 $H \times W = 600 \times 800$ . For the Tanks and Temples dataset, we use the camera parameters provided by R-MVSNet (Yao et al. 2019) to evaluate our method with the image resolution  $H \times W = 544 \times 960$ . Following previous practices (Yao et al. 2019; Yan et al. 2020), we first use the maximum classification probability as the depth probability of estimated depths. Then, we apply the designed dynamic depth map fusion method to generate the final point clouds.

#### **Benchmarking Results**

In this section, we directly use our model trained on the DTU training set without any fine-tuning for benchmarking evaluations. We compare our method with other state-of-the-art MVS methods, including geometric and learning-based methods. The geometric methods include Furu (Furukawa and Ponce 2010), Tola (Tola, Strecha, and Fua 2012), Gipuma (Galliani, Lasinger, and Schindler 2015), COLMAP (Schönberger et al. 2016), ACMM (Xu and Tao 2019) and ACMP (Xu and Tao 2020b). For the learning-based methods, SurfaceNet (Ji et al. 2017), MVSNet (Yao et al. 2018), R-MVSNet (Yao et al. 2019), CasMVSNet (Gu et al. 2020), CVP-MVSNet (Yang et al. 2020), UC-SNet (Cheng et al. 2020), AttMVS (Luo et al. 2020) and D<sup>2</sup>HC-RMVSNet (Yan et al. 2020) are compared.

**Results on DTU.** We evaluate our method on the DTU evaluation set. The comparison results are shown in Table 1. Our method achieves the best completeness and competitive overall performance among compared methods. In particular, our method outperforms the previous recurrent regularization methods, including R-MVSNet and D<sup>2</sup>HC-RMVSNet. Moreover, our method is very competitive with the 3D filtering methods, *e.g.*, CVP-MVSNet and UCSNet. The qualitative comparisons in Fig. 5 show that our reconstructed point clouds are more complete than UCSNet and D<sup>2</sup>HC-RMVSNet.

Results on Tanks and Temples. We evaluate the generalization ability of NR2-Net on the Tanks and Temples dataset. Table 2 summarizes the evaluation results. As can be seen, our method yields the best mean  $F_1$  score, 60.49%, over all published methods, demonstrating that our method generalizes better than other methods. In addition, our method achieves much better performance than other MVS methods based on 3D cost volume filtering on this dataset. We believe this is because the scenes in Tanks and Temples are larger than the scenes in DTU, while the 3D cost volume filtering methods can only sample limited depth planes due to the GPU memory limitation. In contrast, our method can sample sufficient depth values and better considers the longrange dependencies in the depth direction, making it more feasible in practice. Fig. 6 shows the qualitative results of our reconstructed point clouds.

#### **Ablation Study**

**Effect of different components.** In this section, we conduct an ablation study on the DTU evaluation set to verify the effectiveness of non-local recurrent regularization and dynamic depth map fusion module in the proposed NR2-Net. The baseline model is created by removing the non-local recurrent regularization module and replacing the dy-

Model	NLR	DF	Acc.↓	Comp.↓	Overall↓
Baseline			0.350	0.473	0.411
Model-A		$\checkmark$	0.378	0.405	0.391
Model-B	✓		0.344	0.364	0.354
Full	✓	$\checkmark$	0.370	0.332	0.351

Table 3: Ablation study of different components in our NR2-Net. NLR and DF mean non-local recurrent regularization and dynamic depth map fusion respectively.

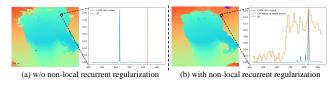


Figure 7: Visual comparison of predicted probability distribution at one pixel between Baseline (a) and Full model (b). The cell states of depth blocks indicate the circle region belongs to the background and constrain the probability distribution of cost map regularization to produce true estimates.

namic depth map fusion with dynamic consistency checking, which means that Eq. (15) becomes  $\tau(\mu) = 0.35$  as (Yan et al. 2020) suggested. The baseline model is also trained with the procedure as described in the previous section. Based on the baseline model, we add the non-local recurrent regularization module and use the dynamic depth map fusion step by step to construct model-A, Model-B and Full model. The comparison results are listed in Table 3. This clearly demonstrates the effectiveness of our proposed two core modules in NR2-Net. In particular, by comparing Baseline and Model-B, Model-A and Full model, we observe that our proposed non-local regularization improves accuracy and greatly boosts completeness, leading to significant overall performance improvement. By comparing Baseline and Model-A, Model-B and Full model, we see that our proposed dynamic fusion can produce more complete point clouds, resulting in better overall performance. This also demonstrates the generalization of our proposed fusion for different depth inference networks.

To further demonstrate the effect of the proposed nonlocal recurrent regularization, we try to visualize the cell states of different depth blocks. In fact, it is hard to give comprehensive visualization about these cell states. Instead, we show the probability distribution of these cell states to see whether they indicate the global scene context. To this end, we average the features of B(t) along the channel dimension and apply the softmax operation along the depth dimension for all depth blocks to obtain the probability distribution. Then, this probability distribution is aligned with the probability distribution of all depth planes by copying the value of each block to its covering s depth planes. The visualization results are shown in Fig. 7. The circle region in Fig. 7 is challenging as it is close to the central object in the spatial domain but in fact it belongs to the background areas. Without non-local recurrent regularization, it is easy

Method	Input Size	Output Size	Mem.	Time
MVSNet	800×576	200×144	8.9	0.78
R-MVSNet	800×576	200×144	1.2	0.89
D <sup>2</sup> HC-RMVSNet	200×144	200×144	0.9	2.67
Ours	200×144	200×144	1.0	1.90
CVP-MVSNet	800×576	800×576	2.2	0.49
D <sup>2</sup> HC-RMVSNet	800×576	800×576	2.4	20.94
Ours	800×576	800×576	3.4	16.65

Table 4: Evaluation of GPU memory [GB] and runtime [s] for different methods.

for the baseline model to propagate the depth information of the central object to the circle region, resulting in incorrect estimation for this region (cf. Fig. 7(a)). With our proposed non-local recurrent regularization, the cell states of depth blocks indicate the circle region belongs to the background areas and constrain the regularization of LSTM cells (cf. Fig. 7(b)). This demonstrates to some extent that the cell states of depth blocks capture the global scene context to help cost map regularization.

Runtime and GPU memory. We evaluate the runtime and GPU memory usage of our method to generate each depth map on the DTU evaluation set. Table 4 shows the comparison results with other methods. For pair comparison with MVSNet and R-MVSNet, we run different methods with the same depth sample number D=256. We see that 3D filtering methods take up a lot of memory and cannot tackle highres input. Recurrent regularization methods greatly reduce memory consumption at the cost of runtime and can tackle high-res input. Although cascade 3D filtering methods such as CVP-MVSNet employ the coarse-to-fine strategy to fit in high-res input, they can only sample limited depth planes in each stage. This prevents them from perceiving full-space global context information in the depth direction, hindering their performance in practical large-scale scenes, e.g., Tanks and Temples dataset.

# Conclusion

We presented a novel non-local recurrent regularization network for multi-view stereo. A novel depth attention module is proposed to model non-local interactions within depth blocks. These non-local interactions are updated in a gated recurrent way to model global scene context. In this way, long-range dependencies along the depth direction are captured and in turn utilized to help regularize cost maps. In addition, we design a dynamic depth map fusion strategy to adaptively take both the depth probability and depth consistency into account. This further improves the robustness of the point cloud reconstruction. Ablation studies demonstrate the effectiveness of our proposed core components. Experimental results show that our method achieves competitive results on the indoor DTU dataset and exhibits excellent performance on the large-scale Tanks and Temples dataset.

# References

- Aanæs, H.; Jensen, R. R.; Vogiatzis, G.; Tola, E.; and Dahl, A. B. 2016. Large-Scale Data for Multiple-View Stereopsis. *International Journal of Computer Vision*, 120(2): 153–168.
- Barnes, C.; Shechtman, E.; Finkelstein, A.; and Goldman, D. B. 2009. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. In *ACM SIG-GRAPH*, 24:1–24:11.
- Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848.
- Chen, R.; Han, S.; Xu, J.; and Su, H. 2019. Point-based multi-view stereo network. In *Proceedings of the IEEE International Conference on Computer Vision*, 1538–1547.
- Cheng, S.; Xu, Z.; Zhu, S.; Li, Z.; Li, L. E.; Ramamoorthi, R.; and Su, H. 2020. Deep Stereo using Adaptive Thin Volume Representation with Uncertainty Awareness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Collins, R. T. 1996. A space-sweep approach to true multiimage matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 358–363.
- Fu, C.; Pei, W.; Cao, Q.; Zhang, C.; Zhao, Y.; Shen, X.; and Tai, Y.-W. 2019. Non-local recurrent neural memory for supervised sequence modeling. In *Proceedings of the IEEE International Conference on Computer Vision*, 6311–6320.
- Furukawa, Y.; and Ponce, J. 2010. Accurate, Dense, and Robust Multiview Stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8): 1362–1376.
- Galliani, S.; Lasinger, K.; and Schindler, K. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, 873–881.
- Gallup, D.; Frahm, J.; Mordohai, P.; Yang, Q.; and Pollefeys, M. 2007. Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.
- Gu, X.; Fan, Z.; Zhu, S.; Dai, Z.; Tan, F.; and Tan, P. 2020. Cascade Cost Volume for High-Resolution Multi-View Stereo and Stereo Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*
- Hartmann, W.; Galliani, S.; Havlena, M.; Gool, L. V.; and Schindler, K. 2017. Learned Multi-patch Similarity. In *Proceedings of the IEEE International Conference on Computer Vision*, 1595–1603.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.
- Hosni, A.; Rhemann, C.; Bleyer, M.; Rother, C.; and Gelautz, M. 2013. Fast Cost-Volume Filtering for Visual Correspondence and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2): 504–511.

- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708.
- Huang, P.; Matzen, K.; Kopf, J.; Ahuja, N.; and Huang, J. 2018. DeepMVS: Learning Multi-view Stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2821–2830.
- Ji, M.; Gall, J.; Zheng, H.; Liu, Y.; and Fang, L. 2017. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, 2307–2315.
- Kar, A.; Häne, C.; and Malik, J. 2017. Learning a Multi-View Stereo Machine. In *Advances in Neural Information Processing Systems*, 365–376.
- Kazhdan, M.; and Hoppe, H. 2013. Screened Poisson Surface Reconstruction. *ACM Trans. Graph.*, 32(3): 29:1–29:13.
- Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*.
- Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representation*.
- Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and Temples: Benchmarking Large-scale Scene Reconstruction. *ACM Trans. Graph.*, 36(4): 78:1–78:13.
- Kolmogorov, V.; and Zabih, R. 2002. Multi-camera Scene Reconstruction via Graph Cuts. In *Proceedings of the European Conference on Computer Vision*, 82–96. ISBN 978-3-540-47977-2.
- Luo, K.; Guan, T.; Ju, L.; Huang, H.; and Luo, Y. 2019. Pmvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, 10452–10461.
- Luo, K.; Guan, T.; Ju, L.; Wang, Y.; Chen, Z.; and Luo, Y. 2020. Attention-Aware Multi-View Stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1590–1599.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32, 8026–8037.
- Schönberger, J. L.; Zheng, E.; Frahm, J.-M.; and Pollefeys, M. 2016. Pixelwise View Selection for Unstructured Multi-View Stereo. In *Proceedings of the European Conference on Computer Vision*, 501–518.
- Tola, E.; Strecha, C.; and Fua, P. 2012. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23: 903–920.
- Wang, F.; Galliani, S.; Vogel, C.; Speciale, P.; and Pollefeys, M. 2021. PatchmatchNet: Learned Multi-View Patchmatch Stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 14194–14203.

- Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.
- Woo, S.; Park, J.; Lee, J.-Y.; and Kweon, I. S. 2018. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision*, 3–19.
- Xu, Q.; and Tao, W. 2019. Multi-Scale Geometric Consistency Guided Multi-View Stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5483–5492.
- Xu, Q.; and Tao, W. 2020a. Learning Inverse Depth Regression for Multi-View Stereo with Correlation Cost Volume. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Xu, Q.; and Tao, W. 2020b. Planar Prior Assisted Patch-Match Multi-View Stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Xue, Y.; Chen, J.; Wan, W.; Huang, Y.; Yu, C.; Li, T.; and Bao, J. 2019. Myscrf: Learning multi-view stereo with conditional random fields. In *Proceedings of the IEEE International Conference on Computer Vision*, 4312–4321.
- Yan, J.; Wei, Z.; Yi, H.; Ding, M.; Zhang, R.; Chen, Y.; Wang, G.; and Tai, Y.-W. 2020. Dense Hybrid Recurrent Multi-view Stereo Net with Dynamic Consistency Checking. In *Proceedings of the European Conference on Computer Vision*.
- Yang, J.; Mao, W.; Alvarez, J. M.; and Liu, M. 2020. Cost Volume Pyramid Based Depth Inference for Multi-View Stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Yao, Y.; Luo, Z.; Li, S.; Fang, T.; and Quan, L. 2018. MVS-Net: Depth Inference for Unstructured Multi-view Stereo. In *Proceedings of the European Conference on Computer Vision*, 767–783.
- Yao, Y.; Luo, Z.; Li, S.; Shen, T.; Fang, T.; and Quan, L. 2019. Recurrent mysnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5525–5534.
- Zheng, E.; Dunn, E.; Jojic, V.; and Frahm, J. M. 2014. Patch-Match Based Joint View Selection and Depthmap Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1510–1517.