Hybrid Cost Volume Regularization for Memory-efficient Multi-view Stereo Networks

Qingtian Zhu[1]
zqt@stu.pku.edu.cn
Zizhuang Wei[2]
weizizhuang@huawei.com
Zhongtao Wang[1]
wangzhongtao@whu.edu.cn
Yisong Chen[1]
chenyisong@pku.edu.cn
Guoping Wang[1]✉
wgp@pku.edu.cn

[1] School of Computer Science
Peking University, China
[2] Huawei Technologies, China

Abstract

Learning-based multi-view stereo (MVS) has been studied for years. To overcome the problem of massive computational overhead and memory footprint, different regularization schemes have been attempted. For instance, recurrent methods trade time for space and regularize the cost volume as a sequence, with a RNN to interchange the depth-wise context between sliced cost maps. Meanwhile, cascade methods follow a coarse-to-fine regularization fashion, which enables a gradually refined depth range but still requires a large amount of memory. To this end, we present a novel network for multi-view stereo, termed as HR-MVSNet, which adopts a hybrid design of cascade coarse-to-fine and recurrent cost volume regularization. HR-MVSNet benefits not only from the low memory consumption by the recurrent regularization scheme, but also from the fast inference speed brought by cascade methods. Extensive experiments show that our HR-MVSNet achieves a nice balance between performance and efficiency. It is able to conduct satisfactory reconstruction while still keeps the memory footprint at a relatively low level. For the point clouds and comparative experiments with HR-MVSNet, please contact the first author.

1 Introduction

Multi-view stereo (MVS) aims to reconstruct a dense geometric representation of the observed scene given a collection of images with known camera parameters. It is a fundamental problem in computer vision and has various applications, *e.g.*, 3D reconstruction, augmented reality and autonomous driving. Learning-based MVS methods have achieved impressive results in terms of reconstruction quality, since deep features can provide more robust matching clues against weakly textured regions and varying lighting conditions. Typically, a learning-based MVS network firstly extracts image features and then builds a unified

cost volume under the frustum of the reference camera with a series of depth hypotheses. The cost volume is then regularized to obtain the probability distribution of depth values. At last, a depth estimator is adopted to turn the per-pixel distribution into an estimated depth map.

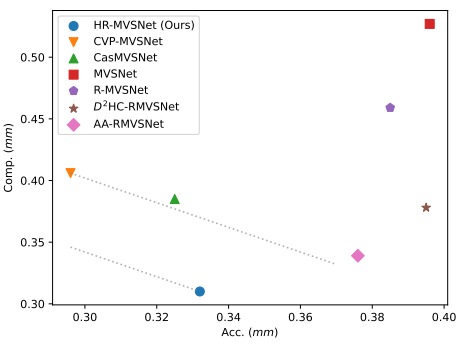
The massive computation overhead and memory occupancy of the 3D U-Net in MVS-Net [20] for cost volume regularization make the network less applicable in practice. To mitigate this problem, previous attempts follow different regularization schemes, namely recurrent regularization and cascade regularization. Proposed in R-MVSNet [21], recurrent regularization models depth-wise matching costs as sequential data and applies a RNN to deliver inter-depth context information. As a well acknowledged representative of cascade regularization, CasMVSNet [7] performs cost volume regularization in a multi-stage manner, where a coarse depth map is estimated first and used to guide subsequent sampling of local finer depth.

MVS methods [4, 13, 16, 18] adopting either regularization scheme have successfully reduced the computational costs of MVSNet but still suffer from the following problems. (a) Recurrent methods trade time for space so that with a relatively low memory demand, the inference time becomes dramatically longer. (b) Cascade methods apply 3D CNNs for stage-wise cost volume regularization which require the whole volume to be constantly maintained in the memory, making the memory occupancy practically unacceptable for a common GPU model. The detailed introduction of both methods is elaborated in Sec. 2. Recently, efficiency-focused methods for MVS reconstruction are emerging. Patchmatch-Net [15], inspired by the traditional PatchMatch algorithm [2, 3], only adopts 2D CNNs for regularization and significantly improves the efficiency of learning-based MVS.

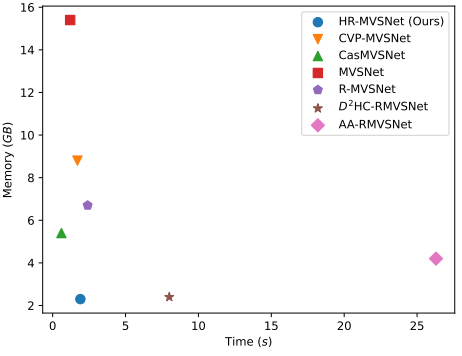
To this end, we propose a novel efficient network for MVS, termed as HR-MVSNet, which benefits from both RNN-based recurrent methods and multi-stage cascade methods. It adopts a hybrid regularization scheme where the overall architecture follows a multi-stage cascade scheme but in each stage, the cost volume is processed in a typically recurrent way. The idea of hybrid regularization makes HR-MVSNet capable of achieving state-of-the-art performance at a low memory cost. Fig. 1 provides a visualized comparison of HR-MVSNet and other state-of-the-art methods in terms of reconstruction quality, inference time and memory consumption.

2 Related Work

Deep learning has been introduced to the task of MVS for better reconstruction quality. MVSNet [20] follows the traditional plane sweep algorithm [5], where the depth range is discretized into finite depth value candidates and a cost volume is built to measure the variance of multi-view image features with each of the depth values given. In this way, MVSNet encodes camera parameters and image features into one cost volume in a differentiable way, enabling an end-to-end training scheme. MVSNet adopts a 3D U-Net for cost volume regularization, which is computationally expensive and brings high memory occupancy. Several solutions have been proposed to alleviate this problem. They can be categorized into RNN-based recurrent methods [16, 18, 21] and multi-stage cascade methods [4, 7, 19, 23]. Fig. 2 provides an illustration of recurrent and cascade regularization methods.



(a) Reconstruction error (accuracy error and completeness error) of different networks.



(b) Runtime efficiency of different networks.

Figure 1: A visualized comparison in performance and efficiency of HR-MVSNet and other state-of-the-art methods on the evaluation set of DTU dataset [1]. In (a), methods lying on the same dotted line share the same overall reconstruction error.

2.1 Recurrent Methods

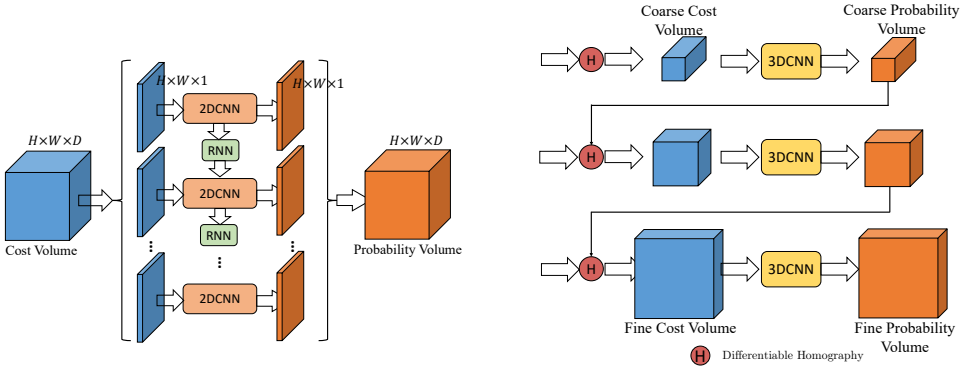
Recurrent methods regularize the 3D cost volumes recurrently, and adopt RNNs to transmit features between adjacent depth hypotheses. The cost volume is divided into slices at the depth dimension and the slices are regularized sequentially by a 2D CNN. As a result, the memory consumption of cost volume regularization becomes invariant to the number of depth hypotheses. R-MVSNet [21] adopts convolutional GRU units for cost volume regularization while D^2HC-RMVSNet [18] and AA-RMVSNet [16] choose LSTM units for better robustness and generalizability. Since recurrent methods trade time for space, they are capable of handling a large number of depth hypotheses, but at the cost of inference speed.

2.2 Cascade Methods

Led by CasMVSNet [7], multi-stage cascade methods follow a coarse-to-fine pattern for making depth hypotheses [4, 19] (with normally three stages). In the first stage, a low-resolution depth map is estimated with a rough sampling of depth values. In the following stage, the image resolution is lifted and the plane sweeping is guided by the interpolated coarse depth, where the depth hypotheses are sampled around the coarse depth values. Larger resolution depth maps are estimated as the candidate depth division becomes finer. Benefiting from the gradually narrowed depth range, cascade methods are known to obtain accurate depth maps with a much faster inference speed compared to recurrent methods. Cascade methods generally adopt non-shared 3D CNNs for cost volume regularization at different stages, which leads to substantial or even unacceptable memory consumption.

3 Method

We follow the common pipeline of depth-based MVS reconstruction where per-view depth maps of the image set are estimated first and then fused to obtain the dense 3D point cloud. Specifically, for a reference image $\mathbf{I}_0$, the proposed network estimates a corresponding depth



(a) Recurrent Regularization (b) Cascade Regularization

Figure 2: An illustration of recurrent regularization scheme and cascade regularization scheme. (a) In recurrent regularization, the cost volume is divided from the dimension of D. Each cost slice is regularized sequentially and a RNN is applied to thread all intermediate outputs. (b) In cascade regularization, cost volumes are regularized stage by stage in a multi-scale manner, where early estimated depth guides subsequent differentiable homography. Each stage is regularized by a 3D CNN.

map by aid of its $N-1$ neighboring source images $\{\mathbf{I}_i | i = 1, \ldots, N-1\}$, as well as the camera parameters of all N images.

We present the overall architecture of the network in Fig. 3. The network first applies a feature extraction network which extracts multi-scale deep image features. It then constructs cost volumes with extracted features and aggregates multiple pairwise cost volumes into one. The cost volume encodes featuremetric similarity between all $N-1$ sources image and the reference image as well as the respective camera parameters. Afterwards, a cost volume regularization network with hybrid design is adopted to leverage 3D context in an efficient manner. At last, the depth map of the reference view is outputted.

3.1 Feature Extraction

The consistency across multi-view images in a typical learning-based MVS network is measured by featuremetric similarity. To extract multi-scale deep image features for coarse-to-fine cascade regularization, we apply a FPN [10] for multi-scale feature extraction.

3.2 Cost Volume Construction

The goal of cost volume construction and aggregation is to encode image features as well as camera parameters into a canonical space so the network is well adapted to any arbitrary value of N. Concretely, plane sweep algorithm [5] defines the canonical space as the fronto-parallel planes of the reference camera frustum.

Since our hybrid regularization scheme is three-staged following the conventions of cascade regularization [7, 23], the determination of depth hypotheses is coarse-to-fine, where early estimated depth guides subsequent stages. Please refer to the Supplemental Material for detailed information. Note that the hybrid regularization enables a more flexible choice of sampling strategies.

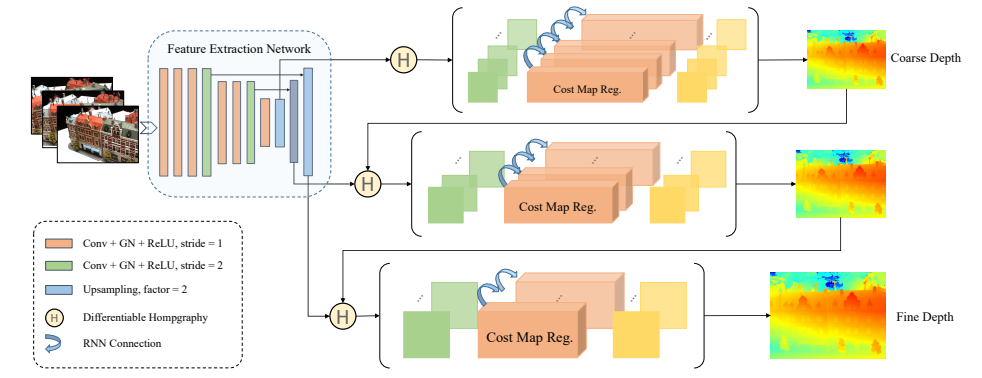


Figure 3: An overview of the network architecture of HR-MVSNet. Multi-scale deep image features are extracted and cost volumes are build in a coarse-to-fine manner. For each of the three stages, RNN-based recurrent regularization is applied.

The warping process from a pixel of the reference view $\mathbf{p} \in \mathbb{R}^2$, with a specific depth hypothesis d, to the i-th source view, is formulated as

$$\hat{\mathbf{p}} = \pi_i(\mathbf{R}_i \pi_0^{-1}(\mathbf{p}; d) + \mathbf{t}_i), \tag{1}$$

where $\pi : \mathbb{R}^3 \to \mathbb{R}^2$ denotes the perspective projection parameterized by known camera intrinsics. $\mathbf{R}_i$ and $\mathbf{t}_i$ stand for the relative rotation and translation from the reference view to the i-th source view. We denote the warped feature map of the i-th source image with a global depth hypothesis d as $\hat{\mathbf{F}}_i^{(d)}$. The pairwise matching cost between the source view and the reference view is

$$\mathbf{c}_i^{(d)} = \|\hat{\mathbf{F}}_i^{(d)} - \mathbf{F}_0\|_2^2. \tag{2}$$

The squared ℓ-2 norm measures the featuremetric similarity between one source image and the reference image. To reduce the dimension of N and aggregate all pairwise cost volumes into one, we follow the adaptive inter-view aggregation scheme in [16] to estimate the aggregated cost volume at d as

$$\mathbf{C}^{(d)} = \frac{1}{N-1} \sum_i [1 + \omega_\theta(\mathbf{c}_i^{(d)})] \odot \mathbf{c}_i^{(d)}, \tag{3}$$

where $\odot$ represents Hadamard multiplication and $\omega_\theta : \mathbb{R}^{H \times W \times F} \to \mathbb{R}^{H \times W \times 1}$ is a gated convolutional network parameterized by θ. With the depth hypothesis d assigned as each possible sampled values, we obtain the aggregated cost volume for regularization.

3.3 Hybrid Cost Volume Regularization

The constructed and aggregated 3D cost volume provides an elementary measure of featuremetric similarity. The procedure of cost volume regularization aims to leverage spatial constraints to denoise the volume as well as to enforce the piece-wise smoothness of depth maps. It is also acknowledged as the most computationally expensive module in a learning-based MVS network.

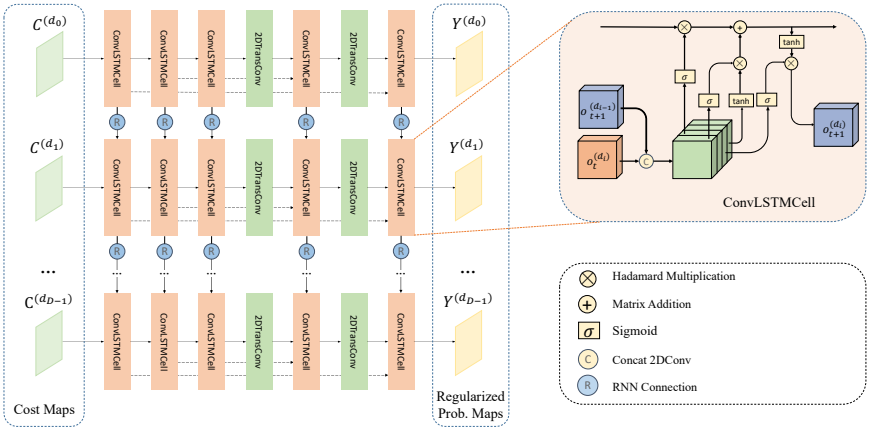


Figure 4: Illustration of the hybrid network for stage-wise cost volume regularization. It contains 5 RNN connections with LSTM architecture and 1 2D encoder-decoder CNN.

The overall design of the proposed hybrid regularization is illustrated in Fig. 4. It follows a cascade pattern, where three coarse-to-fine regularization stages are applied. In each regularization stage, we instead apply the principle of recurrent methods and adopt a hybrid network with convolution layers (for cost map regularization) and LSTM (Long Short-Term Memory) units (for inter-depth context). Since larger resolution and D lead to heavier computation, we can balance the computational overhead of the three stages, *i.e.*, using a large D at a coarse resolution and a smaller D when the resolution is finer. As a result, HR-MVSNet inferences faster than recurrent methods while remains memory-efficient.

The convolutional LSTM contains 1 encoder-decoder network with skip connection and 5 LSTM connections iterating at different intermediate layers of the encoder-decoder network. The encoder-decoder is a 2D CNN which regularizes a cost map (a sliced cost volume at D) at a time. For example, to obtain the intermediate output $o_{t+1}^{(d_i)}$, where $t+1$ and d_i denote the index of 2DConv layers and index of cost maps respectively, we need both $o_t^{(d_i)}$ and $o_{t+1}^{(d_{i-1})}$. The upper-right part of Fig. 4 shows one single cell of the convolutional LSTM, where former outputs are concatenated to current inputs and get processed by a convolution layer and then the tensor is split into four branches for different gates.

3.4 Depth Estimator & Loss Function

As is mentioned in Sec. 3.2, for a more convenient choice of sampling strategies at inference phase, we apply the winner-take-all strategy to obtain the depth estimation from the regularized probability volume.

Accordingly, we train the network end-to-end with classification-based cross entropy loss, where each depth hypothesis is considered as a pre-defined class. The loss function of the s-th stage is defined as

$$\mathcal{L}_s = -\frac{1}{|\mathbf{M}_s|} \sum_{\mathbf{p} \in \mathbf{M}_s} \log \mathbf{P}(\mathbf{p}; d_{\text{gt}}), \tag{4}$$

where d_{gt} is the corresponding ground truth depth value and $\mathbf{M}$ is the binary mask indicating

Method	Acc.(*mm*)↓	Comp.(*mm*)↓	Overall(*mm*)↓	Time(*s*)	Mem.(*GB*)
Gipuma [6]	**0.283**	0.873	0.578	-	-
COLMAP [24]	0.400	0.664	0.532	-	-
MVSNet [20]	0.396	0.527	0.462	1.2	15.4
R-MVSNet [21]	0.385	0.459	0.422	2.4	6.7
D^2HC-RMVSNet [18]	0.395	0.378	0.386	8.0	2.4
AA-RMVSNet [16]	0.376	0.339	0.357	26.3	4.2
Vis-MVSNet [23]	0.369	0.361	0.365	-	-
CasMVSNet [8]	0.325	0.385	0.355	0.6	5.4
CVP-MVSNet [19]	0.296	0.406	0.351	1.7	8.8
PatchmatchNet [13]	0.427	**0.277**	0.352	**0.3**	3.6
HR-MVSNet	0.332	0.310	**0.321**	1.9	**2.3**

Table 1: Evaluation results on the evaluation set of DTU dataset [1]. The metrics are reconstruction errors defined in [1]. The table includes previous non-learning traditional methods, recurrent methods and cascade methods. The inference time and memory footprint are tested according to reported experiment settings.

the valid subset of pixels. The final loss is obtained by summing the loss of all three stages with respective weights of 0.5, 1.0 and 2.0 empirically.

4 Experiments

4.1 Datasets

We apply the following three datasets for experiments. DTU dataset [1] is captured under well-controlled laboratory conditions with a fixed camera rig, containing 128 scans. Following the practice of MVSNet [20], we split DTU dataset into 79 training scans, 18 validation scans, and 22 evaluation scans. BlendedMVS dataset [22] is a large-scale dataset for multi-view stereo and contains objects and scenes of varying complexity and scale. There are 106 training scans and 7 validation scans. Tanks and Temples benchmark [8] is a public benchmark acquired under realistic conditions, which contains 8 scenes for the intermediate subset and 6 for the advanced.

4.2 Implementation Details

4.2.1 Training

We train HR-MVSNet on the training set of DTU dataset [1] and set $N = 5$ and image resolution $H \times W = 512 \times 640$. The number of depth hypotheses, namely D, of each stage is respectively 48, 32, and 8; the corresponding depth interval decays by 0.5 after each stage. The network is implemented by PyTorch and trained with Adam for 10 epochs with an initial learning rate of 0.001 and a cosine annealing schedule [11]. The batch size is 4 on 4 NVIDIA RTX TITAN GPUs. The training phase takes about 16 hours and occupies 10 *GB* memory of each GPU. Note that the training of RNN-based networks requires much more memory than inference to store all intermediate outputs and gradients.

4.2.2 Evaluation

For the evaluation on DTU dataset [1], we set $N = 5$, $H \times W = 864 \times 1152$, D of each stage as 96, 16, 8, and the factor of interval decay as 0.5. We apply forward depth sampling for all

Method	**Int.Mean**	Family	Francis	Horse	L.H.	M60	Panther	P.G.	Train	**Adv.Mean**	Audi.	B.R.	C.R.	Museum	Palace	Temple
COLMAP [23]	42.14	50.41	22.25	26.63	56.43	44.83	46.97	48.53	42.04	27.24	16.02	25.23	34.70	41.51	18.05	27.94
ACMM [24]	57.27	69.24	51.45	46.97	63.20	55.07	57.64	60.08	54.48	34.02	23.41	32.91	41.17	48.13	23.87	34.60
DeepC-MVS [9]	59.79	71.91	54.08	42.29	**66.54**	55.77	**67.47**	60.47	**59.83**	**34.54**	**26.30**	34.66	**43.50**	45.66	23.09	34.00
AttMVS [25]	60.05	73.90	62.58	44.08	64.88	56.08	59.39	**63.42**	56.06	31.93	15.96	27.71	37.99	**52.01**	29.07	28.84
CasMVSNet [8]	56.84	76.37	58.45	46.26	55.81	56.11	54.06	58.18	49.51	31.12	19.81	38.46	29.10	43.87	27.36	28.11
Vis-MVSNet [26]	60.03	77.40	60.23	47.07	63.44	62.21	57.28	60.54	52.07	33.78	20.79	38.77	32.45	44.20	28.73	37.70
PatchmatchNet [15]	53.15	66.99	52.64	43.24	54.87	52.87	49.54	54.21	50.81	32.31	23.69	37.73	30.04	41.80	28.31	32.29
R-MVSNet [27]	50.55	73.01	54.46	43.42	43.88	46.80	46.69	50.87	45.25	29.55	19.49	31.45	29.99	42.31	22.94	31.10
D^2HC-RMVSNet [18]	59.20	74.69	56.04	49.42	60.08	59.81	59.61	60.04	53.92	-	-	-	-	-	-	-
AA-RMVSNet [16]	61.51	77.77	59.53	51.53	64.02	64.05	59.47	60.85	54.90	33.53	20.96	**40.15**	32.05	46.01	29.28	32.71
HR-MVSNet	**63.12**	**80.55**	**65.27**	**52.85**	64.05	**64.24**	62.18	60.17	55.66	34.27	19.29	39.31	36.22	45.46	**30.53**	**34.80**

Table 2: Benchmarking results on the Tanks and Temples [8]. The evaluation metric is mean F-score (**higher is better**). **Bold** figures indicate the best scores.

three stages. For filtering and fusion of depth maps, we adopt the dynamic consistency check proposed in [18], where both confidence-based thresholding and geometric consistency are enforced. The detailed ablation study of the hyperparameters is in Sec. 4.4.

Before benchmarking on Tanks and Temples benchmark [8], we further finetune the network with the training set of BlendedMVS dataset [22]. Following the practice in [16], we resize input images to the size of 544×1024 and 544×960. Different from the evaluation on DTU dataset, we adopt inverse depth sampling at the first stage. Other hyperparameters are consistent with the evaluation on DTU dataset.

4.3 Results

In Tab. 1 we show the comparison between HR-MVSNet and previous well-known methods in terms of reconstruction error and runtime overhead. HR-MVSNet obtains the lowest overall reconstruction error while it achieves a better balance between inference time and memory occupation. Being as memory-efficient as RNN-based methods, *e.g.*, D^2HC-RMVSNet [18] and AA-RMVSNet [16], the time required for inference is shortened. Compared to cascade methods [4, 7, 23] which mainly adopt 3D CNNs for cost volume regularization, HR-MVSNet lowers the memory footprint, making it more applicable on low-end devices.

For benchmarking on Tanks and Temples online benchmark [8], we demonstrate the quantitative comparison in Tab. 2 as well as the qualitative comparison in Fig. 5. Our HR-MVSNet outperforms previous cascade methods and recurrent methods on several reconstruction cases.

4.4 Ablation Study

We here study the influence of different numbers of input views N, different image resolutions $H \times W$, and different numbers of depth candidates D, on the evaluation set of DTU dataset [1]. As is shown in Tab. 3, the network achieves allround superior results when $N = 5$. It is worth noting that the performance is no longer better when N is increased from 5 to 7. It is probably due to the sparse camera distribution of the dataset. Similar results are observed in [15]. As is shown in Tab. 4, the optimal resolution tends to be 864×1152. From Tab. 5, we empirically set the numbers of depth hypotheses as 96, 16, and 8.

In addition, to further demonstrate the inference efficiency of HR-MVSNet, we conduct ablation experiments towards the regularization method applied on different input image sizes, whose results are shown in Fig. 6. The depth maps are half the resolution of input

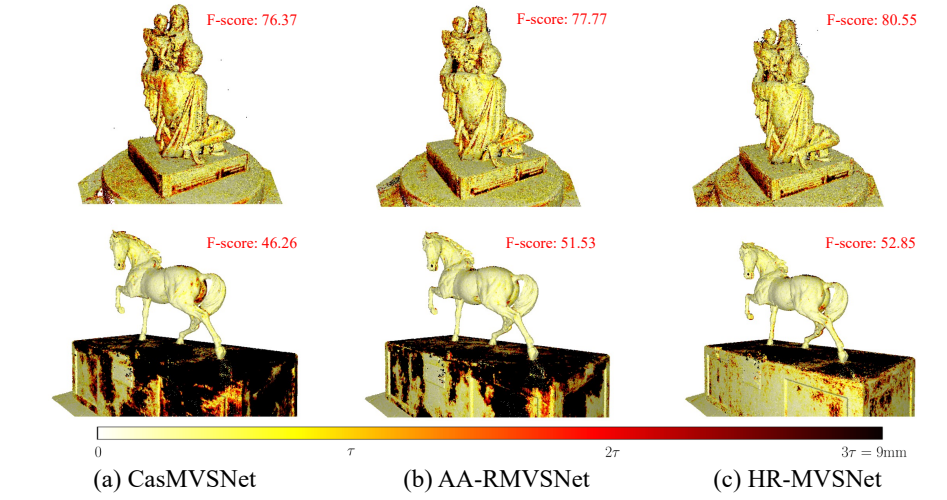
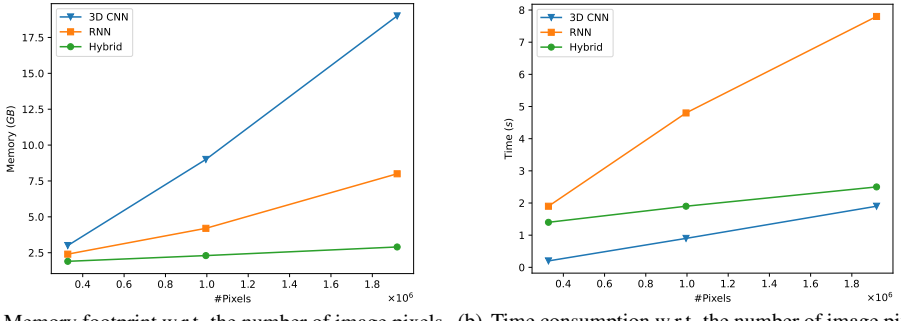


Figure 5: Comparison of reconstructed results with a cascade method [7], and a recurrent method [16] on Tanks and Temples benchmark [8]. τ is the scene-relevant distance threshold and darker regions indicate larger error encountered.



(a) Memory footprint w.r.t. the number of image pixels. (b) Time consumption w.r.t. the number of image pixels.

Figure 6: Ablation study of different regularization settings.

images. For RNN-based regularization, the scheme follows [21] and for the one with 3D CNN, we adopt [20].

5 Discussions

The motivation of the hybrid design of cost volume regularization is simple and straightforward. It enables HR-MVSNet to keep the memory consumption at a low level and inference faster than pure recurrent methods. There is actually an apparent trade-off between space and time in learning-based MVS networks. Methods using 3D CNNs for cost volume regularization are faster at inference phase for its good parallelizability of computation. Recurrent methods trade time for space and occupies less memory for inference. The hybrid regularization pattern inherently achieves a good balance between these two sides.

We frankly list the known limitations of HR-MVSNet as follows. (a) Though the infer-

N	Acc.	Comp.	Overall	Mem.(GB)
3	0.357	0.318	0.337	**1.9**
5	**0.332**	**0.310**	**0.321**	2.3
7	0.343	0.334	0.338	2.9

Table 3: Ablation study of the number of input views (N) on the evaluation set of DTU dataset [11].

$H \times W$	Acc.	Comp.	Overall
1200×1600	0.362	0.327	0.345
864×1152	0.332	**0.310**	**0.321**
512×640	**0.315**	0.405	0.360

Table 4: Ablation study of the resolution of input images ($H \times W$) on the evaluation set of DTU dataset [11].

D_0, D_1, D_2	Overall(mm)	Time(s)
96, 16, 8	**0.321**	1.9
48, 16, 4	0.352	**1.3**
48, 16, 8	0.349	1.7

Table 5: Ablation study of the number of depth hypotheses (D) on the evaluation set of DTU dataset [11].

ence speed has been accelerated compared to recurrent methods, there is still a significant gap between the inference speed of HR-MVSNet and real-time applications. (b) As HR-MVSNet adopts coarse-to-fine cascade plane sweep, its performance heavily depends on the estimation quality of the first stage. (c) Though HR-MVSNet is memory-efficient at inference phase, it still consumes massive GPU memory at training stage since all intermediate layers are stored for back-propagation. (d) Similar to recurrent methods, the memory consumption is reduced by deleting intermediate tensors which will be no more referred from GPU. It in fact increases the difficulty of implementation since normally deep learning frameworks have complicated mechanism of data caching.

6 Conclusion

In this paper, we present HR-MVSNet, which adopts a hybrid design of cost volume regularization that benefits from both RNN-based regularization and cascade multi-stage regularization. It enables memory-efficient recurrent regularization at inference phase and keeps the memory consumption at a low level, making HR-MVSNet applicable under varying scenes and on diverse devices. Extensive experiments on a public dataset and an online benchmark show that our HR-MVSNet is able to achieve satisfactory results compared to existing methods which adopt either recurrent regularization or cascade regularization.

Acknowledgements

This research is funded by National Key Technology Research and Development Program of China (2021YFF0500901) and Independent Research Project of Guangdong Laboratory (Zhuhai) of Southern Marine Science and Engineering (SML2021SP101).

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[3] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, 2011.

[4] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2524–2534, 2020.

[5] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 358–363. IEEE, 1996.

[6] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015.

[7] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping 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*, pages 2495–2504, 2020.

[8] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

[9] Andreas Kuhn, Christian Sormann, Mattia Rossi, Oliver Erdler, and Friedrich Fraundorfer. Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 404–413. IEEE, 2020.

[10] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[11] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representation (ICLR)*, 2017.

[12] Keyang Luo, Tao Guan, Lili Ju, Yuesong Wang, Zhuo Chen, and Yawei Luo. Attention-aware multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1590–1599, 2020.

[13] Xinjun Ma, Yue Gong, Qirui Wang, Jingwei Huang, Lei Chen, and Fan Yu. Epp-mvsnet: Epipolar-assembling based depth prediction for multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5732–5740, 2021.

[14] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016.

[15] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14194–14203, 2021.

[16] Zizhuang Wei, Qingtian Zhu, Chen Min, Yisong Chen, and Guoping Wang. Aa-rmvsnet: Adaptive aggregation recurrent multi-view stereo network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6187–6196, 2021.

[17] Qingshan Xu and Wenbing Tao. Multi-scale geometric consistency guided multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5483–5492, 2019.

[18] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *European Conference on Computer Vision*, pages 674–689. Springer, 2020.

[19] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020.

[20] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[21] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

[22] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long 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*, pages 1790–1799, 2020.

[23] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. In *British Machine Vision Conference (BMVC)*, 2020.