GC-MVSNet: Multi-View, Multi-Scale, Geometrically-Consistent Multi-View Stereo

Vibhas K. Vats, Sripad Joshi, David J. Crandall
Indiana University Bloomington
{vkvats, joshisri, djcran}@iu.edu

Md. Alimoor Reza
Drake University
md.reza@drake.edu

Soon-heung Jung
ETRI
zeroone@etri.re.kr

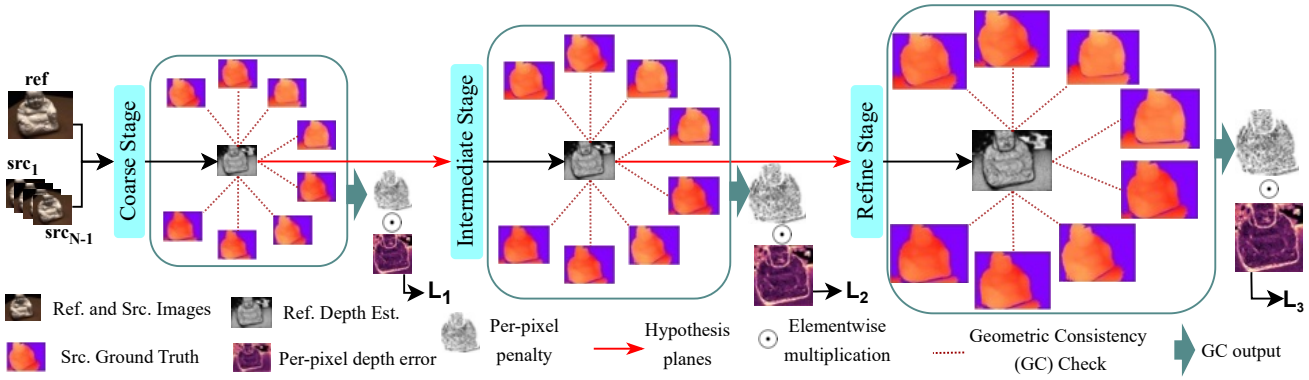

Figure 1: Our multi-view, multi-scale geometric consistency checking process. During training, the geometric consistency of the estimated depth map is explicitly modeled across multiple source views. This allows the model to more quickly and accurately learn about geometric consistency, allowing the trained model to produce better reconstructions during inference.

Abstract

Traditional multi-view stereo (MVS) methods rely heavily on photometric and geometric consistency constraints, but newer machine learning-based MVS methods check geometric consistency across multiple source views only as a post-processing step. In this paper, we present a novel approach that explicitly encourages geometric consistency of reference view depth maps across multiple source views at different scales during learning (see Fig. 1). We find that adding this geometric consistency loss significantly accelerates learning by explicitly penalizing geometrically inconsistent pixels, reducing the training iteration requirements to nearly half that of other MVS methods. Our extensive experiments show that our approach achieves a new state-of-the-art on the DTU and BlendedMVS datasets, and competitive results on the Tanks and Temples benchmark. To the best of our knowledge, GC-MVSNet is the first attempt to enforce multi-view, multi-scale geometric consistency during learning.

1. Introduction

Traditional multi-view stereo (MVS) methods such as Gipuma [10], Furu [9], COLMAP [33], and Tola [36] rely on solving for photometric and geometric consistency constraints across multiple views. Recent machine learning-based MVS methods [2, 3, 6, 12, 25, 30, 41, 45, 47–49, 52] use deep networks to extract feature maps and then construct 3D cost volumes to measure similarity between feature maps [12]. Each paper in this stream of machine learning-based approaches has introduced innovations that have significantly improved the quality of depth estimates and point cloud reconstructions, like multi-level feature extraction, attention-based feature matching, similarity-based cost-volume creation, and improved loss formulations. These modern methods use plane sweep volumes to implicitly encode geometric constraints, and perform multi-view geometric consistency checks as a postprocess after inference to filter the depth maps. However, they do not explicitly model multi-view geometric constraints during learning. Instead, learning about multi-view geometric information thus must happen only implicitly.

In this paper, we show, for the first time, that providing the model with explicit multi-view geometric cues using geometric consistency checks across multiple source views during training (see Fig. 1) significantly improves accuracy while significantly lowering the training iteration requirements. We formulate a multi-stage model called GC-MVSNet which learns geometric cues at three scales. At each scale, we introduce a novel multi-view geometric consistency module that performs geometric consistency checks of reference view depth estimates across multiple source views and generates a per-pixel penalty. This penalty

is then combined with per-pixel depth error (estimated using cross-entropy loss at each stage) to generate the final loss.

This formulation of loss function provides abundant geometric cues to accelerate learning of the model. Our extensive experiments show that GC-MVSNet requires nearly half the training iterations needed by other recent models [6, 12, 30, 41, 48]. Our approach also achieves a new state-of-the-art accuracy on DTU [15] and BlendedMVS [50] datasets, and competitive results on Tanks and Temples [20]. To the best of our knowledge, GC-MVSNet is the first attempt to leverage multi-view, multi-scale geometric consistency checks during the training process. We also perform extensive ablation experiments to demonstrate the effectiveness of the proposed approach.

In summary, in this paper:

- We propose a novel multi-view, multi-scale geometric consistency (GC) module during learning that encourages geometric consistency of reference view depth maps across multiple source views.
- We show that this technique reduces the training iteration requirements to nearly half that of other models, by explicitly providing multi-view geometric cues during learning.
- We show that the module is highly general and can be plugged into different MVS pipelines to enhance geometric cues during training.

2. Related Work

The taxonomy proposed by Furukawa and Ponce [9] classifies MVS methods into four primary scene representations: *volumetric fields* [7,21,34,35], *point clouds* [2,22], *3D meshes* [8], and *depth maps* [1, 3, 6, 10, 12, 30, 33, 45, 48, 52]. Depth map-based methods can further be categorized into either traditional techniques based on feature detection and solving for geometric constraints [1, 9, 10, 33], or learning-based methods [3,6,12,30,45,48,52]. The latter have become very popular in the last few years.

Among the learning-based techniques, MVSNet [48] formulates a single-stage MVS pipeline by encoding camera parameters via differential homography to build 3D cost volumes. It requires a huge amount of memory and computation as it uses 3D U-Nets [32] to regularize the cost volume. Subsequent work has taken two main approaches to alleviate this problem: RNNs [41,44,46,49] and coarse-to-fine multi-stage methods [3,6,12,30,45,52].

Among the RNN-based methods, R-MVSNet [49] sequentially regularizes the 2D cost maps along the depth direction via gated recurrent units. AA-RMVSNet [41] slices the cost volume along D depth hypotheses and regularizes the horizontal and vertical components using CNN and ConvLSTMCells, respectively. Xu et al. [44] use RNNs to model global dependencies with non-local depth interactions. Yan et al. [46] couple LSTM and U-Net architectures to regularize multi-scale information.

Coarse-to-fine multi-stage methods [2,3,6,12,25,30,41, 47, 49] have significantly improved the quality of depth estimates and point cloud reconstructions. They initially predict a low-resolution (coarse) depth map and then progressively refine it. For example, inspired by other coarse-to-fine methods [37,40,51], CasMVSNet [12] presents a multi-stage formulation of single-stage MVSNet [48], TransMVSNet [6] focuses on feature matching to improve performance over CasMVSNet, UniMVSNet [30] uses unified loss formulation to further improve over CasMVSNet. CVP-MVSNet [47] builds a cost volume pyramid in a coarse-to-fine manner. UCS-Net [12] uses an adaptive thin volume module that uses a smaller number of hypothesis planes to efficiently partition the local depth range within learned small intervals. TransMVSnet [6] uses transformer based [18,38] feature matching to promote similarity in the extracted features. UniMVSNet [30] unifies the advantages of regression and classification methods by designing a unified focal loss in a multi-stage framework.

While all these methods improve the performance of the multi-stage MVS pipeline by improving specific portions, none of them explicitly models multi-view geometric cues during the learning process. Consequently, during training these models depend on the limited geometric cues available from multiple source views and the cost function formulation. Xu and Tao [45] present a multi-scale geometric consistency-guided MVS method that uses multi-hypothesis joint view selection to leverage structured region information to sample better candidate hypotheses. They hypothesize that the upsampled depth maps of source images can geometrically constrain these estimates, and use reprojection error [33, 53] to indicate this consistency. In this paper, we use forward-backward reprojection with multiple source views to check the geometric consistency of depth estimates and to generate per-pixel penalties for geometrically inconsistent pixels.

3. Methodology

Our goal is to take N views as input, including a reference image $I_0 \in \mathbb{R}^{H \times W \times 3}$ and its paired N-1 source view images $\{I_i\}_{i=1}^{N-1}$, along with the corresponding camera parameters $c_0, ..., c_N$, and then to estimate the reference view depth map (D_0) as the output.

3.1. Network Overview

Fig. 2 shows the architecture of our approach, which we call Geometric Consistency MVSNet (or GC-MVSNet). We use a deformable convolution-based [4] feature pyramid network (FPN) [23] architecture (Sec. 3.4) to extract features from input images in a coarse-to-fine manner in three

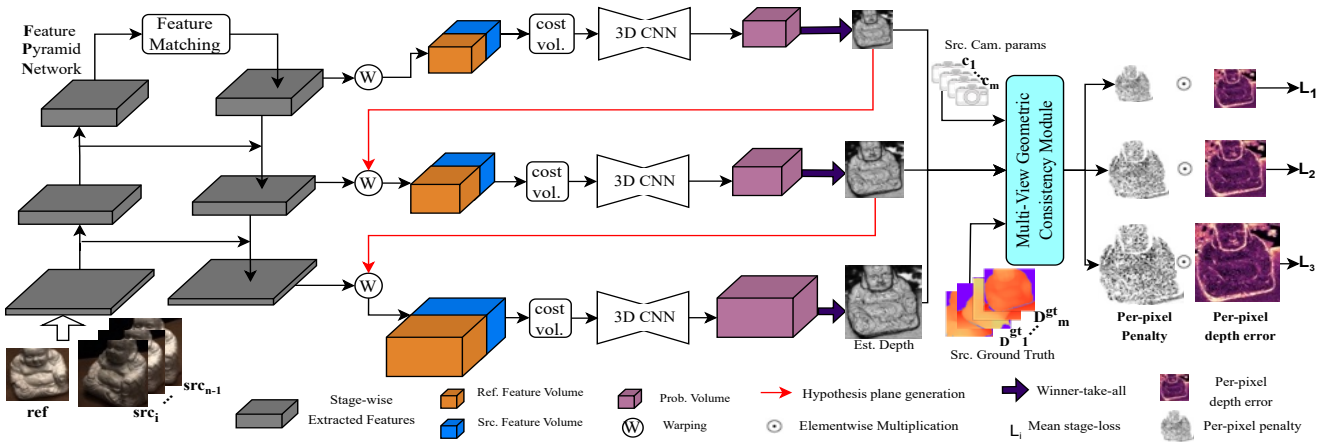

Figure 2. The GC-MVSNet architecture. The GC module is applied at the end of each stage. It takes the estimated reference view depth, M source view ground truths and their camera parameters to perform a multi-view geometric consistency check. It generates a per-pixel penalty (ξ_p) for reference view, which is element-wise multiplied with per-pixel depth error (ξ_d) to generate stage loss L_i. ξ_d is calculated using cross-entropy loss. All stage losses are added to produce the final loss.

stages. At each stage, we build a correlation-based cost volume of shape $H' \times W' \times D'_h \times 1$ using feature maps of shape $N \times H' \times W' \times C$, where H', W', and C denote the height, width, and number of channels of a given stage, and D'_h is the number of depth hypotheses at the corresponding stage. The cost volume is regularized with a cost regularization network. We use a winner-takes-all strategy to estimate the depth map D_0 at each stage. At only the coarse stage, we apply feature matching [6] with linear attention [6, 18] to leverage global context information within and between reference and source view features.

We employ the GC module at each stage. The GC module checks the geometric consistency of each pixel in D_0 across M source views and generates ξ_p (Sec. 3.2), a pixel-wise factor that is multiplied with the per-pixel depth error (ξ_d), calculated using a cross-entropy function. It penalizes each pixel in D_0 for its inconsistency across M source views to accelerate geometric cues learning during training. TransMVSNet [6] trained with cross-entropy loss (TransMVSNet-B) is our baseline; see Table 7 for different stages of GC-MVSNet.

Algorithm 1 Geometric Consistency Check Algorithm

Inputs: $D_0, c_0, D_i^{gt}, c_i^{gt}, D_{pixel}, D_{depth}$
Output: $per\_pixel\_penalty$
Require $M \geq N$
$mask\_sum \leftarrow 0$
$D \leftarrow D_1^{gt}, ...D_M^{gt}$
$c \leftarrow c_1^{gt}, ...c_M^{gt}$
for D_i^{gt}, c_i^{gt} in $zip(D, c)$ **do**
  $D''_{P''_0}, P''_0 \leftarrow FBR(D_0, c_0, D_i^{gt}, c_i^{gt})$ ▷ Alg. 2
  $PDE \leftarrow ||P_0 - P''_0||_2$
  $RDD \leftarrow {}^1/_{D_0} ||D''_{P''_0} - D_0||_1$
  $PDE_{mask} \leftarrow PDE > D_{pixel}$
  $RDD_{mask} \leftarrow RDD > D_{depth}$
  $mask \leftarrow PDE_{mask} \vee RDD_{mask}$
  if $mask > 0$ **then**
    $mask \leftarrow 1$
  else
    $mask \leftarrow 0$
  end if
  $mask\_sum \leftarrow mask\_sum + mask$
end for
$per\_pixel\_penalty \leftarrow 1 + mask\_sum/M$

3.2. Multi-View Geometric Consistency Module

GC-MVSNet estimates reference depth maps at three stages with different resolutions. At each stage, the GC module takes D_0, M source view ground truths $D_1^{gt}, ...D_M^{gt}$, and their camera parameters $c_0, ...c_M$ as input (see Alg. 1). The GC module is then initialized with a *geometric inconsistency mask sum* (or $mask\_sum$) of zero at each stage. This mask sum accumulates the inconsistency of each pixel across the M source views. For each source view, the GC module performs *forward-backward reprojection* of D_0 to generate the penalty and then adds it to the mask sum.

Forward-backward reprojection (FBR), as shown in Alg. 2, is a crucial three-step process. First, we project each pixel P_0 of D_0 to its i^{th} neighboring source view using intrinsic (K_R, K_S) and extrinsic (E_R, E_S) camera parameters to obtain corresponding pixel P'_i, and denote the corresponding depth map as $D_{(R \to S)}$. Second, we similarly remap D_i^{gt} to obtain $D_{S_{remap}}$. Finally, we back project $D_{S_{remap}}$ to the reference view using intrinsic and extrinsic camera parameters to obtain $D''_{P''_0}$ (see Alg. 2). D_0 and $D''_{P''_0}$ represent the depth values of pixels P_0 and P''_0 [13]. With P''_0 and $D''_{P''_0}$, we calculate the pixel displacement error (PDE) and relative depth difference (RDD). PDE is the L_2 norm between P_0 and P''_0 and RDD is the absolute value difference between $D''_{P''_0}$ and D_0 relative to D_0 as shown in Alg. 1.

For each stage, we generate two binary masks of in-

Algorithm 2 Forward Backward Reprojection (FBR)

Inputs: $D_0, c_0, D_i^{gt}, c_i^{gt}$
Output: $D''_{P''_0}, P''_0$

$K_R, E_R \leftarrow c_0$; $K_S, E_S \leftarrow c_i^{gt}$
$D_{(R\to S)} \leftarrow K_S \cdot E_S \cdot E_R^{-1} \cdot K_R^{-1} \cdot D_0$ ▷ Project
$X_{D_{(R\to S)}}, Y_{D_{(R\to S)}} \leftarrow D_{(R\to S)}$
$D_{S_{remap}} \leftarrow REMAP(D_i^{gt}, X_{D_{(R\to S)}}, Y_{D_{(R\to S)}})$ ▷ Remap
$D''_{P''_0} \leftarrow K_R \cdot E_R \cdot E_S^{-1} \cdot K_S^{-1} \cdot D_{S_{remap}}$ ▷ Back project
$P''_0 \leftarrow (X_{D''_{P''_0}}, Y_{D''_{P''_0}})$

consistent pixels, PDE_{mask} and RDD_{mask}, by applying thresholds D_{pixel} and D_{depth}, and then take a logical-OR of the two to preduce a single mask of inconsistent pixels. These inconsistent pixels are assigned a value 1 and all other pixels, including the consistent and the out-of-scope pixels, are assigned 0 to form a penalty mask. This penalty mask is then added to the mask sum (Alg. 1), which accumulates the penalty mask for each of the M source views to generate a final mask sum with values $\in [0, M]$. Each pixel value indicates the number of inconsistencies of the pixel across the M source views.

From this mask sum, we then generate the inconsistency penalty ξ_p for each pixel. Our initial approach generated ξ_p by dividing the mask sum by M to normalize within the $[0, 1]$. However, we found that using ξ_p itself for element-wise multiplication reduces the contribution of perfectly consistent (zero inconsistency) pixels to zero, preventing further improvement of such pixels. To avoid this, we add 1 so that elements of ξ_p are in $[1, 2]$. A reference view binary mask is applied on initial ξ_p to generate the final ξ_p, as shown in Fig. 3.

Occlusion and its impact. Occluded pixels naturally arise in multi-view stereo, since 3D points are often not visible in all views. These occluded pixels have a major impact on geometric constraints, since the reference view pixels whose corresponding 3D points are occluded are penalized as inconsistent. It is thus important to prevent occluded pixels from dominating the geometric consistency losses. While occlusion is sometimes modeled explicitly [17, 29], we found that our approach is naturally robust to occlusion because of the following three considerations. First, we select the closest M source views as defined in MVSNet [48] to minimize the number of occluded pixels in different source views. Then, during FBR, we remap D_i^{gt} to obtain $D_{S_{remap}}$ and back project it as shown in Alg. 2. Remapping and back projection largely handles extreme cases of occlusion (see Appendix A in Supplemental Material). Finally, we apply reference view binary mask on ξ_p, Fig. 3, to restrict penalties only to valid reference view pixels. The combination of these steps helps us deal with occluded pixels and loss explosion.

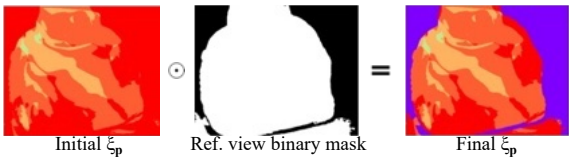

Figure 3. The final ξ_p is the outcome of elementwise multiplication ($\odot$) of initial ξ_p and reference view mask. It restricts the penalties within the reference view mask.

3.3. Cost Function

Most learning-based MVS methods [12, 47, 54] treat depth estimation as a regression problem and use an L_1 loss between prediction and ground truth. Following AA-RMVSNet [41] and UniMVSNet [30], we treat depth estimation as a classification problem and adopt a cross-entropy loss formulation from AA-RMVSNet [41] (see [30] for relative advantages of regression and classification approaches.) The pixelwise penalty ξ_d is calculated at each stage,

$$\xi_d = \sum_{p\in\{P_v\}} \log(P(\tilde{d})_{(P)}), \tag{1}$$

where $\{P_v\}$ is the subset of pixels with valid ground truth, $P(\tilde{d})_{(P)}$ denotes the estimated probability of the depth hypothesis d at pixel P, and $\tilde{d}$ denotes the depth value closest to the ground truth among all hypotheses. We further enhance the one-hot supervision by penalizing each pixel for its inconsistency across different source views. This is implemented using element-wise multiplication ($\odot$) between ξ_d and ξ_p at each stage. The mean stage loss, L_i, is calculated as,

$$\begin{aligned} L_{i(stage)} &= mean(\xi_p \odot \xi_d) \\ \mathcal{L}_{total} &= \alpha.L_1 + \beta.L_2 + \gamma.L_3 \end{aligned} \tag{2}$$

where $L_{i(stage)}$ is the mean stage loss and $\mathcal{L}_{total}$ is the total loss. α, β and γ are the stage-wise weights. This formulation of cost function with pixel-level inconsistency penalty explicitly forces the model to learn to produce multi-view geometrically-consistent depth maps.

3.4. Other Modifications

Besides the geometric consistency module, we made two other major modifications to the MVS pipeline. First, while keeping the feature extraction network architecture as FPN, we replaced its regular convolutional layers with deformable layers [4, 55]. Deformable layers are known to adjust their sampling locations based on model requirements [4, 55]. This helps extract better features for accelerated learning.

Second, most MVS methods [6, 12, 30, 41, 42, 54] use batch normalization [14] and batch synchronisation during training. As observed in [14], batch normalization provides more consistent and stable training with large batch sizes, but it is inconsistent and has a degrading effect on training with smaller batches. MVS methods are restricted to very

	Method	Acc ↓	Comp ↓	Overall ↓
Traditional	Furu [9]	0.613	0.941	0.777
	Tola [36]	0.342	1.190	0.766
	Gipuma [10]	**0.283**	0.873	0.578
	COLMAP [33]	0.400	0.664	0.532
Learning-based	SurfaceNet [16]	0.450	1.040	0.745
	MVSNet [48]	0.396	0.527	0.462
	P-MVSNet [25]	0.406	0.434	0.420
	R-MVSNet [49]	0.383	0.452	0.417
	Point-MVSNet [2]	0.342	0.411	0.376
	CasMVSNet [12]	0.325	0.385	0.355
	CVP-MVSNet [47]	<u>0.296</u>	0.406	0.351
	UCS-Net [3]	0.338	0.349	0.344
	AA-RMVSNet [41]	0.376	0.339	0.357
	UniMVSNet [30]	0.352	0.278	0.315
	TransMVSNet [6]	0.321	0.289	0.305
	GBi-Net* [28]	0.312	0.293	<u>0.303</u>
	MVSTER [39]	0.350	<u>0.276</u>	0.313
	GC-MVSNet (ours)	0.330	**0.260**	**0.295**
	GBi-Net [28]	0.315	0.262	0.289
	GC-MVSNet (ours)	0.323	**0.255**	0.289

Table 1. Quantitative results on DTU evaluation set at 864 × 1152 resolution. Accuracy (Acc), completeness (comp) and overall are in mm. $*$ means that GBiNet is re-tested with the same post-processing threshold to all scans for fair comparison with other methods. Gray font shows the methods that use scan-specific thresholds for evaluation. **Bold** and <u>underline</u> represents first and second place, respectively.

small batch sizes, often 1, due to large memory requirements. Thus, we replaced batch normalization with group normalization layers [43] of group size 4 across the network. Group normalization performs normalization across a number of channels that is independent of the number of examples in a batch [43]. We also implement weight standardization [31] for all layers in the network. With these modifications, we achieve stable and reproducible training (see Appendix D in Supplemental Material).

4. Experiments

We evaluate on three datasets with different complexities. **DTU** [15] is an indoor dataset that contains 128 scenes with 49 or 64 views under 7 lighting conditions and predefined camera trajectories. We follow MVSNet [48] for training, validation, and test splits. **BlendedMVS** [50] is a large-scale synthetic dataset with 113 indoor and outdoor scenes. It has 106 training scenes and 7 validation scenes. **Tanks and Temples** [20] is collected from a more complicated and realistic scene, and contains 8 intermediate and 6 advanced scenes. DTU and TnT evaluate using point clouds while BLD evaluates on depth maps.

4.1. Implementation Details

Following the general practice [6], we first train and evaluate our model on DTU. Then, we finetune on BlendedMVS to evaluate on Tanks and Temples. For training on DTU, we set the number of input images $N = 5$ and image resolution as 512×640. The depth hypotheses are sampled from $425mm$ to $935mm$ for coarse-to-fine regularization with the number of plane sweeping depth hypotheses for the three stages set to 48, 32, and 8. The corresponding depth interval ratio (DIR) is set as 2.0, 0.8, and 0.4. The model is trained with Adam [19] for 9 epochs with an initial learning rate (LR_{DTU}) of 0.001, which decays by a factor of 0.5 once after 8^{th} epoch. For the Geometric Consistency (GC) module, we use M=8 and set the stage-wise thresholds D_{pixel} as 1, 0.5, 0.25 and D_{depth} as 0.01, 0.005, 0.0025. We use α=β=1 and $\gamma = 2$ for all experiments. We train our model with a batch size of 3 on 8 NVIDIA RTX A6000 GPUs for about 9 hours.

Method	EPE ↓	e_1 ↓	e_3 ↓
MVSNet [48]	1.49	21.98	8.32
CasMVSNet [12]	1.43	19.01	9.77
CVP-MVSNet [47]	1.90	19.73	10.24
Vis-MVSNet [54]	1.47	15.14	5.13
EPP-MVSNet [26]	1.17	12.66	6.20
TransMVSNet [6]	<u>0.73</u>	<u>8.32</u>	<u>3.62</u>
GC-MVSNet (ours)	**0.48**	**0.89**	**0.97**

Table 2. Quantitative comparison on BlendedMVS evaluation set. We follow evaluation steps described in [5]. **Bold** and <u>underline</u> represents first and second place, respectively.

4.2. Experimental Performance

Evaluation on DTU. On DTU, we generate depth maps with N=5 at an input resolution of 864×1152. We slightly adjust the depth interval ratio (DIR) to $1.6, 0.7, 0.3$ to accommodate the resolution change (more on DIR in Appendix C in Supplemental Material) and use the Fusibile algorithm [10] for depth fusion. Table 1 shows quantitative evaluations, where accuracy is the mean absolute distance in mm from the reconstructed point cloud to the ground truth point cloud, completeness measures the opposite (see Appendix F in Supplemental Material), and overall is the average of these metrics, indicating the overall performance of the models. We find that GC-MVSNet achieves the best overall score as well as the best completeness score, when compared to nearly two dozen previous and state-of-the-art techniques. A qualitative evaluation is presented in Fig. 4 on a few sample MVS problems. We find that our model generates denser and more complete point clouds.

Evaluation on BlendedMVS. Unlike DTU and Tanks and Temples, evaluation on Blended MVS is usually measured as the quality of depth maps, not the quality of point clouds. We set N=5, M=8, image resolution as 576×768, and number of depth planes D=128, and finetune for 10 epochs with one-tenth the learning rate we used for DTU ($\frac{1}{10} LR_{DTU}$). We follow [5] for evaluation process.

Table 2 presents the results of our quantitative evaluation, using three metrics: Endpoint error (EPE) is the average L_1 distance between the estimated and the ground truth

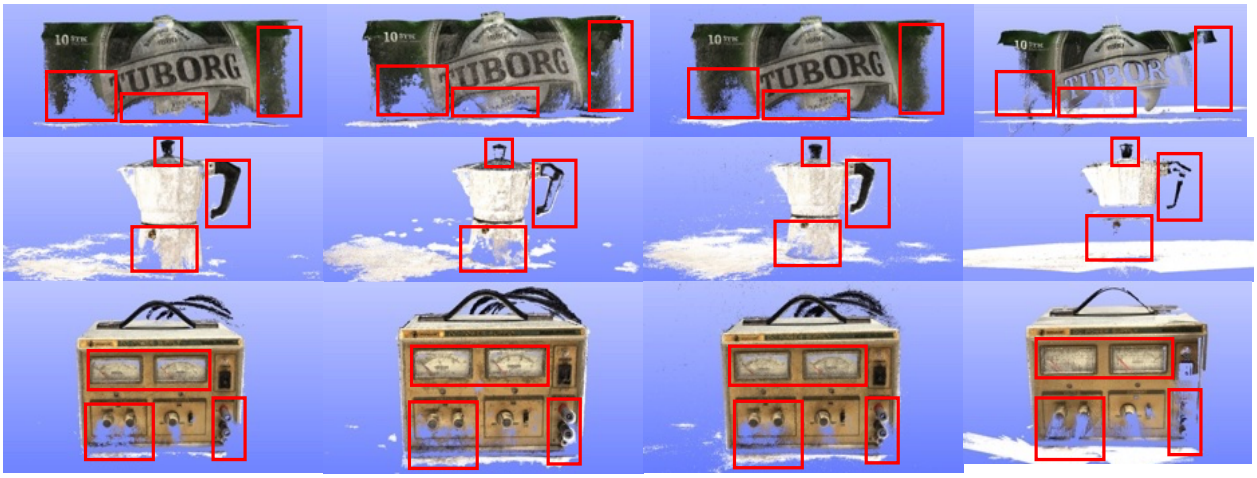

Figure 4. Visual comparison of reconstructed point clouds of GC-MVSNet with CasMVSNet [12], R-MVSNet [49] and Ground truths. Our method obtains a more complete point cloud. See Appendix H in Supplemental Material for all point clouds.

	Intermediate set									Advanced set						
Method	**Mean** ↑	Fam.	Fra.	Hor.	Lig.	M60	Pan.	Pla.	Tra.	**Mean** ↑	Aud.	Bal.	Cour.	Mus.	Pal.	Tem.
COLMAP [33]	42.14	50.41	22.25	26.63	56.53	44.83	46.97	48.53	42.04	27.24	16.02	25.23	34.70	41.51	18.05	27.94
P-MVSNet [25]	55.62	70.04	44.64	40.22	**65.20**	55.08	55.17	60.37	54.29	-	-	-	-	-	-	-
R-MVSNet [49]	50.55	73.01	54.56	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
Point-MVSNet [2]	48.27	61.79	41.15	34.20	50.79	51.97	50.85	52.38	43.06	-	-	-	-	-	-	-
CasMVSNet [12]	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
CVP-MVSNet [47]	54.03	76.50	47.74	36.34	55.12	57.28	54.28	57.43	47.54	-	-	-	-	-	-	-
UCS-Net [3]	54.83	76.09	53.16	43.03	54.00	55.60	51.49	57.38	47.89	-	-	-	-	-	-	-
AA-RMVSNet [41]	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
UniMVSNet [30]	**64.36**	**81.20**	66.43	53.11	63.46	**66.09**	**64.84**	**62.23**	57.53	**38.96**	28.33	44.36	**39.74**	**52.89**	33.80	34.63
TransMVSNet [6]	63.52	80.92	65.83	**56.94**	62.54	63.06	60.00	60.20	**58.67**	37.00	24.84	44.59	34.77	46.49	34.69	36.62
GBi-Net [28]	61.42	79.77	**67.69**	51.81	61.25	60.37	55.87	60.67	53.89	37.32	**29.77**	42.12	36.30	47.69	31.11	36.39
MVSTER [39]	60.92	80.21	63.51	52.30	61.38	61.47	58.16	58.98	51.38	37.53	26.68	42.14	35.65	49.37	32.16	**39.19**
GC-MVSNet(ours)	62.74	80.87	67.13	53.82	61.05	62.60	59.64	58.68	58.48	38.74	25.37	**46.50**	36.65	49.97	**35.81**	38.11

Table 3. Quantitative results on intermediate and advanced sets of Tanks and Temples [20]. **Bold**, single-underline, double-underline represent first, second and third places, respectively.

depth values, and e_1 and e_2 are the ratio of number of pixels with L_1 error larger than 1*mm* and 3*mm*, respectively. The significant improvement in depth map estimates corroborates that providing explicit geometric cues during training helps the model learn about multi-view geometric consistency while requiring much less training iteration. See Appendix H of the Supplemental Material for point clouds.

Evaluation on Tanks and Temples. We also test the performance of our model on an outdoor dataset with the Tanks and Temples benchmark. To adapt to this change, we first finetune our model on BlendedMVS and then evaluate on the intermediate and advanced test sets of Tanks and Temples. We use an image resolution of 576×768, N=7, M=10, one-tenth the learning rate of DTU ($\frac{1}{10}LR_{DTU}$), and D=192 for finetuning. We finetune the model for 12 epochs. The camera parameters and neighboring view selection are used as in R-MVSNet [49] and follow evaluation steps described in CDS-MVSNet [11].

Table 3 presents our quantitative comparison of different methods. GC-MVSNet achieves the third highest spot on the intermediate set and the second highest spot on advanced set evaluation. Fig. 5 shows point clouds visualizing precision and recall comparisons with other MVS methods. See Appendix H in Supplemental Material for point clouds.

4.3. Ablation Study

Having demonstrated the efficacy of our proposed approach relative to the state of the art, we now conduct ablation studies to evaluate the importance of the various components of our model.

Range of ξ_p. ξ_p is generated using the mask sum ($mask\_sum$ in Alg. 1). It is the sum of penalties accumulated across the M source views during multi-view geometric consistency check. At this stage, its elements take a discrete value between 0 and M. Using mask sum as it is leads to very high penalty per-pixel and consequently, very high loss value. Such a high loss value destabilizes the learning process. We control the magnitude of penalty

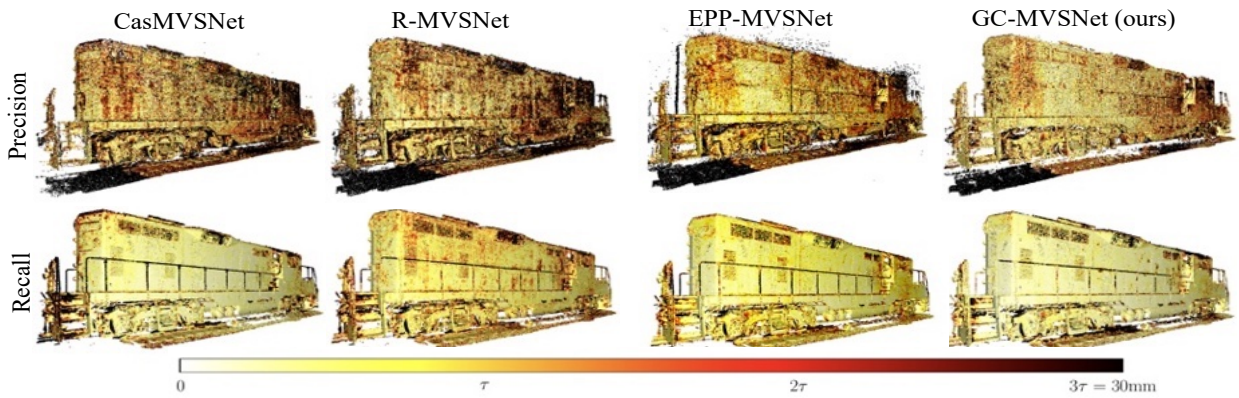

Figure 5. Precision and recall comparison with other methods [12, 27, 49] for Train on Tanks and Temples benchmark. τ is the scene-relevant distance threshold. Darker regions indicate larger error encountered with regard to τ. GC-MVSNet shows visual improvements with brighter regions for precision as well as recall metric.

by controlling the range of per-pixel penalty.

We explore two different ranges to control the magnitude of ξ_p, $[1, 2]$ and $[1, 3]$. To generate $\xi_p \in [1, 2]$, we divide the mask sum by M and then add 1. To generate $\xi_p \in [1, 3]$, we divide the mask by $M/2$ and then add 1. Table 4 shows the impact of these two ranges of penalty for M=8. Since $\xi_p \in [1, 2]$ produces best results, we use it for all other experiments.

Hyperparameters of GC module. The GC module has two types of hyperparameters, global and local. In this section, we investigate the effect of these hyperparameters on our results.

The global hyperparameter M is the number of source views across which the geometric consistency is checked, and is the same for all three stages (coarse, intermediate, and refinement stages). For training on DTU, we vary the value of M while keeping N=5, i.e. while the MVS method uses only 4 source views to estimate D_0, the GC module checks the geometrical consistency of D_0 across M source views. It is important to note that the first N-1 out of M source views are exactly the same used by GC-MVSNet to estimate D_0. We always keep $M \geq N - 1$.

Table 5 presents a quantitative comparison for different values of M and the amount of training iterations required for optimal performance of the model. At M=N-1= 4, i.e. checking geometric consistency across the same number of source views as used by GC-MVSNet to estimate D_0, the model performance significantly improves with sharp decrease in training iteration requirements, as compared to our baseline TransMVSNet-B (Table 7). As we increase the value of M from 4 to 10, the training iteration required by our model further decreases. We find that at $M = 8$, which is twice the number of source views used by GC-MVSNet, it achieves its best performance.

The two local hyperparameters, D_{pixel} and D_{depth}, are the stage-wise thresholds applied to generate PDE_{mask} and RDD_{mask} in Alg. 1. These values are set to smaller values in the later (finer) stages, providing a stricter penalty

ξ_p Range	Acc↓	Comp↓	Overall↓
$[1, 3]$	0.331	0.270	0.3005
$[1, 2]$	**0.330**	**0.260**	**0.295**

Table 4. Impact of range of ξ_p during training on DTU with M=8, N=5. Numbers are generated on DTU evaluation set.

M	Acc↓	Comp↓	Overall↓	Opt. Epoch
4	0.343	0.264	0.3035	12
5	0.342	0.271	0.3065	13
6	**0.326**	0.271	0.298	9
7	0.332	0.270	0.301	10
8	0.330	**0.260**	**0.295**	9
9	0.328	0.280	0.304	9
10	0.329	0.268	0.298	10

Table 5. Quantitative results on DTU evaluation set [15]. M is the number of source views used by the GC module for checking geometric consistency of reference view depth map. Training iteration requirement of the model decreases as M increases.

to geometrically inconsistent pixels at finer resolutions. Table 6 shows the overall performance of GC-MVSNet with a range of different D_{pixel} and D_{depth} thresholds. GC-MVSNet performance remains fairly consistent and it achieves its best performance with D_{pixel}= 1, 0.5, 0.25 and D_{depth}= 0.01, 0.005, 0.0025. We use these threshold values for all datasets throughout the paper.

GC module as a plug-in. Our Geometric Consistency module is generic can be integrated into many different MVS pipelines. To demonstrate this, we tested it with two very different MVS pipeline, CasMVSNet [12] and TransMVSNet [6]. CasMVSNet treats depth estimation as a regression problem, while TransMVSNet treats it as a classification problem and uses winner-take-all to estimate the final depth map. We purposefully choose different methods to show that the GC module can perform well for both types of formulation. We compare the architectures of GC-MVSNet with TransMVSNet and CasMVSNet in Sec. 5.1.

Table 7 presents the results, showing the impact of adding the GC module as well as the *other* modifications (deformable convolution-based FPN with group-norm and

D_{depth}			D_{pixel}			Overall↓
C	I	R	C	I	R	
0.04	0.03	0.02	4	3	2	0.302
0.03	0.0225	0.015	3	2.25	1.5	0.302
0.02	0.015	0.01	2	1.5	1.0	0.298
0.01	0.008	0.006	1	0.8	0.6	0.303
0.01	**0.005**	**0.0025**	**1**	**0.5**	**0.25**	**0.295**
0.008	0.003	0.002	0.8	0.3	0.2	0.303
0.005	0.002	0.001	0.5	0.2	0.1	0.3015

Table 6. Overall score on the evaluation set of DTU [15] for different values of D_{depth} and D_{pixel}. M is fixed at 8. C, I, and R means Coarse, Intermediate and Refine stages.

	Methods	Loss	Other	GC	Overall↓	Epoch
GC as a plug-in	CasMVSNet	L_1	×	×	0.355	16
		L_1	✓	×	0.357	16
		L_1	×	✓	**0.335**	**11**
	TransMVSNet	FL	×	×	0.305	16
		FL	✓	×	0.322	16
		FL	×	✓	**0.303**	**8**
Stages	TransMVSNet-B	CE	×	×	0.332	16
	TransMVSNet-B	CE	✓	×	0.328	16
	TransMVSNet-B	CE	×	✓	0.298	**8**
	GC-MVSNet	CE	✓	✓	**0.295**	9

Table 7. Performance comparison of different MVS methods with different modifications on DTU [15]. L_1, FL, CE and Others indicate L_1 loss, Focal loss [24], Cross-entropy loss [41] and other modifications from Sec. 3.4, respectively.

weight-standardization) in the original pipeline. To observe the absolute impact of adding these modifications, we do not change anything else in the original pipelines. We observe in the table that applying only the *other* modification leads to degradation in performance. It indicates that the *other* modification helps in stabilizing the training process and promoting reproducibility, but has no significant impact on the performance of the model on its own. We also observe a sharp increase in model performance and decrease in training iteration requirements after integrating our GC module into the original pipeline. With GC, training the CasMVSNet pipeline requires only 11 epochs instead of 16 epochs, while TransMVSNet (with GC module) requires only 8 epochs instead of 16 epochs. This corroborates our hypothesis that multi-view geometric consistency significantly reduces training computation because it accelerates learning of geometric cues.

Table 7 also shows different stages of development of GC-MVSNet. TransMVSNet-B uses TransMVSNet pipeline with cross-entropy loss, performs much worse than original TransMVSNet [6] which uses focal loss. With only *other* modifications, it slightly improves the overall performance of the model but does not impact the training iteration requirements. Only after applying the GC module, independently and with *other* modifications, we see significant reduction in training iteration requirements as well as a significant improvement in the overall accuracy metric. This clearly shows the significance of multi-view multi-scale geometric consistency check in the GC-MVSNet pipeline.

5. Discussion

5.1. Comparison to Related Work

GC-MVSNet vs. TransMVSNet. TransMVSNet [6] uses regular 2D convolution-based FPN (with batch-norm) for feature extraction and employs adaptive receptive field (ARF) modules with deformable layers after feature extraction. It trains using focal loss [24]. GC-MVSNet replaces the combination of regular FPN and ARF modules with deformable FPN (with group-norm and weight-standardization) for feature extraction. It trains with cross-entropy loss and GC module for accelerated learning.

GC-MVSNet vs. CasMVSNet: CasMVSNet [12] proposes a coarse-to-fine regularization technique. It uses regular 2D convolutions-based FPN for feature extraction, generates variance-based cost volume and employ depth regression to estimate D_0. The only similarity with our model is that we also use coarse-to-fine regularization.

5.2. Limitations

Like any other MVS method, GC-MVSNet require hyper-parameter tuning during learning. Hyperparameters like, depth interval ratio, number of stage-wise depth hypothesis, number of initial depth hypothesis, depth interval decay factor, etc. impacts model performance. The GC module hyperparameters, D_{pixel}, D_{depth} and M, also require tuning to achieve its best performance. Along with the GC module hyperparameters, the quality of ground truth has also a direct impact on its performance as it uses source view ground truth depth maps for multi-view geometric consistency check.

6. Conclusion

In this paper, we present a novel learning-based MVS pipeline, GC-MVSNet, which explicitly models geometric consistency of reference depth maps across multiple source views during training. To the best of our knowledge, this is the first attempt to leverage multi-view multi-scale geometric consistency check during the training process. We show that the GC module is generic and can be plugged into other MVS methods to accelerate their learning as well. We perform extensive experiments and ablation study to show the advantages of GC-MVSNet. We hope that our work will bring some insights about including explicit geometric reasoning during learning.

Acknowledgement: This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [23ZH1200, The research of the fundamental media·contents technologies for hyper-realistic media space].

References

[1] Neill D. F. Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*, 2008. 2

[2] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1538–1547, 2019. 1, 2, 5, 6

[3] 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. 1, 2, 5, 6

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2, 4

[5] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Deep multi-view stereo gone wild. In *2021 International Conference on 3D Vision (3DV)*, pages 484–493. IEEE, 2021. 5

[6] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8585–8594, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[7] O. Faugeras and R. Keriven. Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344, 1998. 2

[8] Pascal Fua and Yvan G Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *International Journal of Computer Vision*, 16(ARTICLE):35–56, 1995. 2

[9] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1362–1376, 2010. 1, 2, 5

[10] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 873–881, 2015. 1, 2, 5

[11] Khang Truong Giang, Soohwan Song, and Sungho Jo. CURVATURE-GUIDED DYNAMIC SCALE NETWORKS FOR MULTI-VIEW STEREO. In *International Conference on Learning Representations*, 2022. 6

[12] 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. 1, 2, 4, 5, 6, 7, 8

[13] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 3

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 4

[15] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 2, 5, 7, 8

[16] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2326–2334, 2017. 5

[17] Sing Bing Kang, R. Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001. 4

[18] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 2, 3

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 5

[20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 2, 5, 6

[21] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 307–314 vol.1, 1999. 2

[22] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):418–433, 2005. 2

[23] 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. 2

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 8

[25] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-mvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10451–10460, 2019. 1, 2, 5, 6

[26] 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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5712–5720, 2021. 5

[27] 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. 7

[28] Zhenxing Mi, Chang Di, and Dan Xu. Generalized binary search network for highly-efficient multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12991–13000, 2022. 5, 6

[29] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo-occlusion patterns in camera matrix. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 371–378, 1996. 4

[30] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 4, 5, 6

[31] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019. 5

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[33] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 501–518, Cham, 2016. Springer International Publishing. 1, 2, 5, 6

[34] S.M. Seitz and C.R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1067–1073, 1997. 2

[35] Sudipta N. Sinha, Philippos Mordohai, and Marc Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. 2

[36] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large scale multi-view stereo for ultra high resolution image sets. *Machine Vision and Applications*, 23, 09 2011. 1, 5

[37] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 195–204, 2019. 2

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[39] Xiaofeng Wang, Zheng Zhu, Guan Huang, Fangbo Qin, Yun Ye, Yijia He, Xu Chi, and Xingang Wang. Mvster: Epipolar transformer for efficient multi-view stereo. In *European Conference on Computer Vision*, pages 573–591. Springer, 2022. 5, 6

[40] Yan Wang, Zihang Lai, Gao Huang, Brian H Wang, Laurens Van Der Maaten, Mark Campbell, and Kilian Q Weinberger. Anytime stereo image depth estimation on mobile devices. In *2019 international conference on robotics and automation (ICRA)*, pages 5893–5900. IEEE, 2019. 2

[41] 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. 1, 2, 4, 5, 6, 8

[42] Rafael Weilharter and Friedrich Fraundorfer. Highres-mvsnet: A fast multi-view stereo network for dense 3d reconstruction from high-resolution images. *IEEE Access*, 9:11306–11315, 2021. 4

[43] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 5

[44] Qingshan Xu, Martin R. Oswald, Wenbing Tao, Marc Pollefeys, and Zhaopeng Cui. Non-local recurrent regularization networks for multi-view stereo. *CoRR*, abs/2110.06436, 2021. 2

[45] 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. 1, 2

[46] 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. 2

[47] Jiayu Yang, Wei Mao, Jose M. Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 4, 5, 6

[48] 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. 1, 2, 4, 5

[49] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. *Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 6, 7

[50] 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. 2, 5

[51] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6044–6053, 2019. 2

[52] Anzhu Yu, Wenyue Guo, Bing Liu, Xin Chen, Xin Wang, Xuefeng Cao, and Bingchuan Jiang. Attention aware cost volume pyramid based multi-view stereo network for 3d re-

construction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:448–460, 2021. 1, 2

[53] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Recovering consistent video depth maps via bundle optimization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[54] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *British Machine Vision Conference (BMVC)*, 2020. 4, 5

[55] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019. 4

Multi-View, Multi-Scale, Geometrically-Consistent Multi-View Stereo (Supplementary Materials)

Vibhas K. Vats, Sripad Joshi, David J. Crandall
Indiana University Bloomington
{vkvats, joshisri, djcran}@iu.edu

Md. Alimoor Reza
Drake University
md.reza@drake.edu

Soon-heung Jung
ETRI
zeroone@etri.re.kr

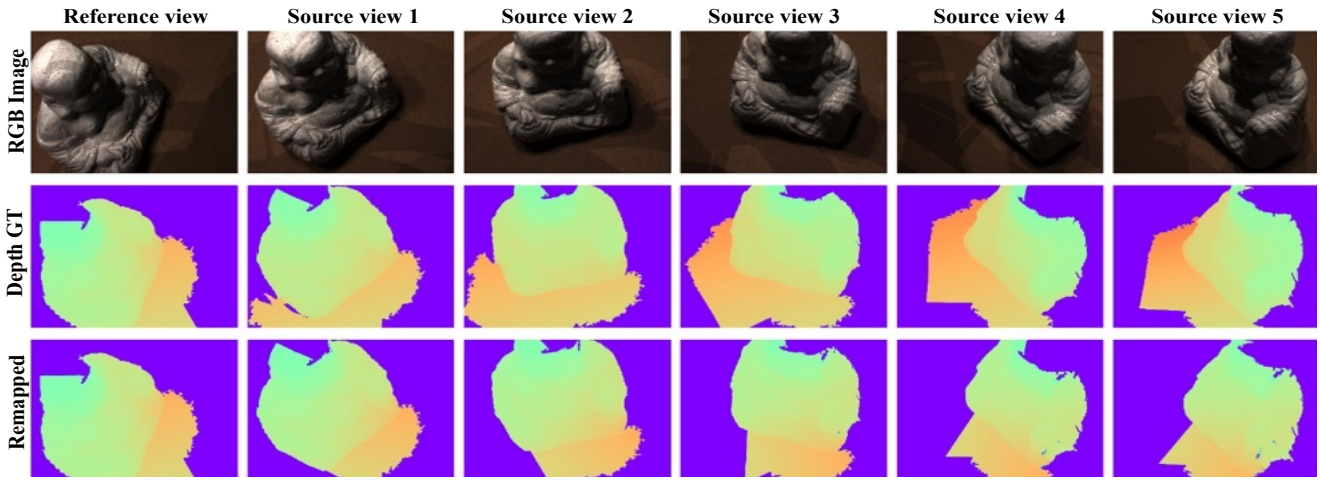

Figure 1: First row shows the selection of M closest source images for a given reference image. Middle row shows the corresponding ground truth depth maps and last row shows the remapped source ground truth depth maps using x-y coordinates of reference view projection to the source view. During remapping, all additional pixels from the source views are ignored. The remapped depths are then back-projected to source view to generate mask. Finally, reference view mask is applied on per-pixel penalty to restrict the penalties. Corresponding final ξ_p is shown in Fig. 3 of the paper. All depth maps are shown within respective view mask.

A. Occlusion and its impact

Modeling occlusion of pixels in multi-view setting is a difficult problem. It is difficult to reason about a pixel in a view whose corresponding 3D points are occluded in other view. The problem becomes significant if a penalty is being attached to all such pixels, like in the proposed multi-view geometric consistency checking module. The GC module checks geometric consistency of each pixel across multiple source views and awards a penalty for inconsistency. Assigning penalties to occluded pixels and multiplying it with depth error adversely impacts the training process. Early in our experiments, we observe that the loss started to explode with training, i.e. as the model trains the loss values starts to increase.

Our investigation suggests that the wrongful penalties of occluded pixels dominated loss during training. We find that our method becomes robust to this problem with a series of steps taken. First, we use the closest source view images as defined by MVSNet [16]. The first row of Fig. 1 shows the source view selection for the given reference view. Choosing closest view to the reference view reduces the number of possible occluded pixels. Second, during forward-backward-reprojection, we remap the source view depth map as per the x-y coordinate projections of the reference view to the source view and then, the remapped values are back-projected to the reference view (see Alg. 2 in the paper). The last row in Fig. 1 shows the remapped version of the source view depth maps. During remapping, all the occluded as well as the additional pixels of the source view is dropped and then this remapped version is back-projected. This handles the extreme cases of occlusion or additional visible pixels. At the end, once the per-pixel penalty is generated, we apply the reference view binary mask on it to do away with any such pixel which is not part of the scene in consideration (see Fig. 3 in the paper).

The combination of these steps help us control the impact of wrongful penalties and stabilize the training process.

B. Geometric Consistency Module

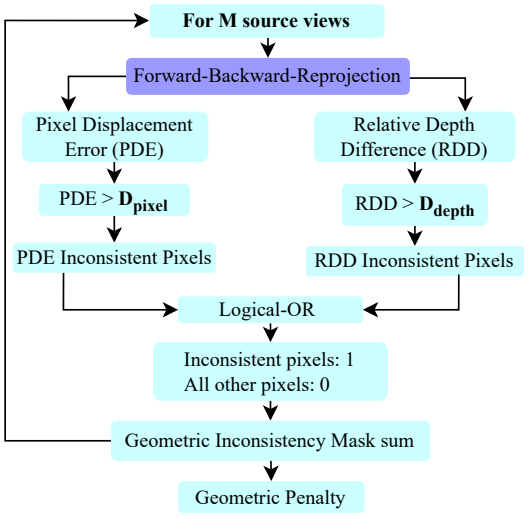

Figure 2. GC module flow-chart for consistency check.

We describe the steps of geometric consistency (GC) module in Fig. 2. At each stage, the geometric consistency of estimated depth map is checked across M source views. For each source view, we perform the forward-backward-reprojection of estimated depth map to reason about geometric inconsistency of pixels (described in Alg. 2). In this three-step process, first, we warp each pixel P_0 of a reference view depth map D_0 to its i^{th} neighboring source view to obtain corresponding pixel $P_i^{'}$. Then, we back-project $P_i^{'}$ into 3D space and finally, we reproject it to the reference view as $P_0^{"}$ using c_0. D_0, $D_{P_i^{'}}^{'}$ and $D_{P_0^{"}}^{"}$ represents depth value of pixels associated with P_0, $P_i^{'}$ and $P_0^{"}$ [6]. With $P_0^{"}$ and $D_{P_0^{"}}^{"}$, we calculate pixel displacement error (PDE) and relative depth difference (RDD). After taking logical-OR between PDE and RDD, we assign value 1 to all inconsistent pixel and zero to all other pixels. The geometric inconsistency mask sum is generated over M source views and averaged to generate per-pixel penalty ξ_p.

C. Depth Interval Ratio (DIR)

ξ_p Range	Stage-wise DIR	Acc↓	Comp↓	Overall↓
[1, 3]	2.0, 0.8, 0.40	0.338	0.269	0.3035
[1, 3]	2.0, 0.7, 0.35	0.343	**0.264**	0.3035
[1, 3]	2.0, 0.7, 0.30	0.331	0.27	0.3005
[1, 3]	1.6, 0.7, 0.30	**0.329**	0.271	**0.300**

Table 1. The performance of GC-MVSNet on evaluation set of DTU [8] with change in stage-wise DIR (depth interval ratio).

DIR directly impacts the separation of two hypothesis planes at pixel level. For a given stage, the pixel-level depth interval is calculated as product of DIR_{stage} and *depth interval* (DI). The value of DI is calculated using *interval scale* and a constant value provided in DTU camera parameter files.

Following the trend of modern learning-based methods [1, 3, 5, 10, 15, 16, 19], we train our model on 512×640 resolution and test on 864×1152 resolution. To adjust for the pixel-level depth interval caused by the increase in resolution, we explore different DIR values for testing on DTU. We train our model with stage-wise DIR $2.0, 0.8, 0.4$ (DIR_{train}). such that the refine stage pixel-level depth interval is same as the provided *interval scale* value of 1.06. Table 1 shows DIR values for evaluation on DTU, we only explore smaller values than DIR_{train} to compensate for the increase in resolution. GC-MVSNet achieves its optimal performance at DIR $1.6, 0.7, 0.3$ with $\xi_p \in [1, 3]$, DIR_{test}. We use the same DIR_{train} and DIR_{test} with $\xi_p \in [1, 2]$.

D. Stabilizing the Training Process

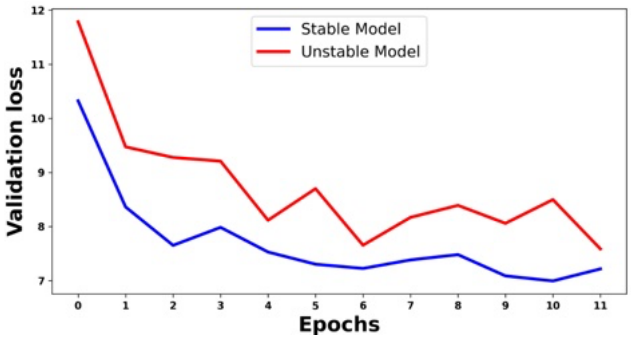

Figure 3. Validation loss on DTU [8] dataset during training. The red line shows the unstable model training, validation loss change in zig-zag manner. Blue line shows stable training with smooth change in validation loss.

Most of the modern learning-based MVS methods [3, 5, 10, 12, 13, 20] use BatchNorm [7] along with Apex (Nvidia) for batch synchronization. BatchNorm is most useful with large batch size. For smaller batch size, like 1 or 3, it degrades the training process [7] by poor estimation of population mean (μ) and std. (σ) over small batch size.

GroupNorm [14] alleviates this problem by estimating μ and σ along the channels instead of batch. Weight-standardization [11] further stabilizes the training and evaluation steps. We refer to the original papers for further understanding of these concepts. GC-MVSNet replaces BatchNorm with GroupNorm and Weight-standardization techniques to stabilize the training process. Fig. 3 shows the difference between model trained with (red line) and without (blue line) BatchNorm. With the use of GroupNorm along with Weight-standardization, the evaluation

loss curve become smooth and stable.

E. Depth Map Fusion Methods

The quality of point clouds depends heavily on depth fusion methods and their hyperparameters. Following the recent learning-based methods [3, 5, 10], we also use different fusion method for DTU and Tanks and Temples dataset. For DTU, we use Fusibile [4] and for Tanks and Temples, we use Dynamic method [3, 12].

Fusibile fusion method uses three hyperparameters, disparity threshold, probability confidence threshold, and consistency threshold. Disparity threshold defines the upper limit of disparity for points to be eligible for fusion. Probability confidence threshold defines the lower limit of confidence above which points are eligible for fusion. The consistency threshold mandates that the eligible points be geometrically consistent across as many source views. During the fusion process, only those points that satisfy all three conditions are fused into point cloud.

Dynamic fusion method uses only two hyperparameters, probability confidence threshold and consistency threshold. Both these hyperparameters have exact same function as in Fusibile method. The disparity threshold is not provided by the user, it is dynamically adjusted during the fusion process.

F. Accuracy and Completeness Metrics

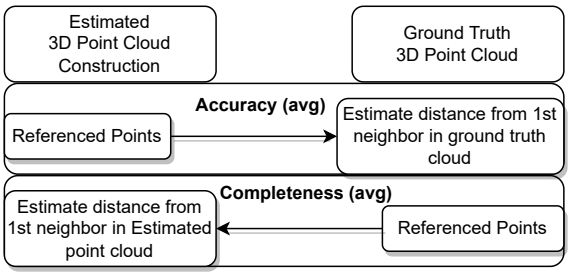

Figure 4. The process of calculating accuracy and completeness for DTU [8] point cloud evaluation.

Accuracy and completeness are two metrics used with DTU [8] dataset. Fig. 4 shows the process of calculation. Accuracy is the average of the distance of the first neighbor from predicted point cloud to ground truth point cloud. It only considers the points which are below the maximum threshold for the distance. For completeness, same process is repeated but with ground truth as referenced point cloud, i.e. the average of the distance of the first neighbor from the ground truth point cloud to the predicted point cloud.

G. Use of Existing Assets

We use PyTorch to implement GC-MVSNet. It is based on CasMVSNet [5] and TransMVSNet [3]. These two methods heavily borrow code from the PyTorch implementation of MVSNet [16].

We use preprocessed images and camera parameters of DTU [8] dataset from official repository of MVSNet [16] and R-MVSNet [17]. We follow [2] for training and testing on BlendedMVS [18]. For Tanks and Temples [9] evaluation, we use images and camera parameters as used in R-MVSNet [17].

H. Point Clouds

In this section, we show all evaluation set points clouds reconstructed using GC-MVSNet on DTU [8], Tanks and Temples [9] and BlendedMVS [18] datasets. Fig. 5, 6 and 7 show all evaluation set point clouds from DTU, Tanks and Temples and BlendedMVS, respectively.

Figure 5. Point clouds reconstructed using GC-MVSNet for all scenes from DTU [8] evaluation set.

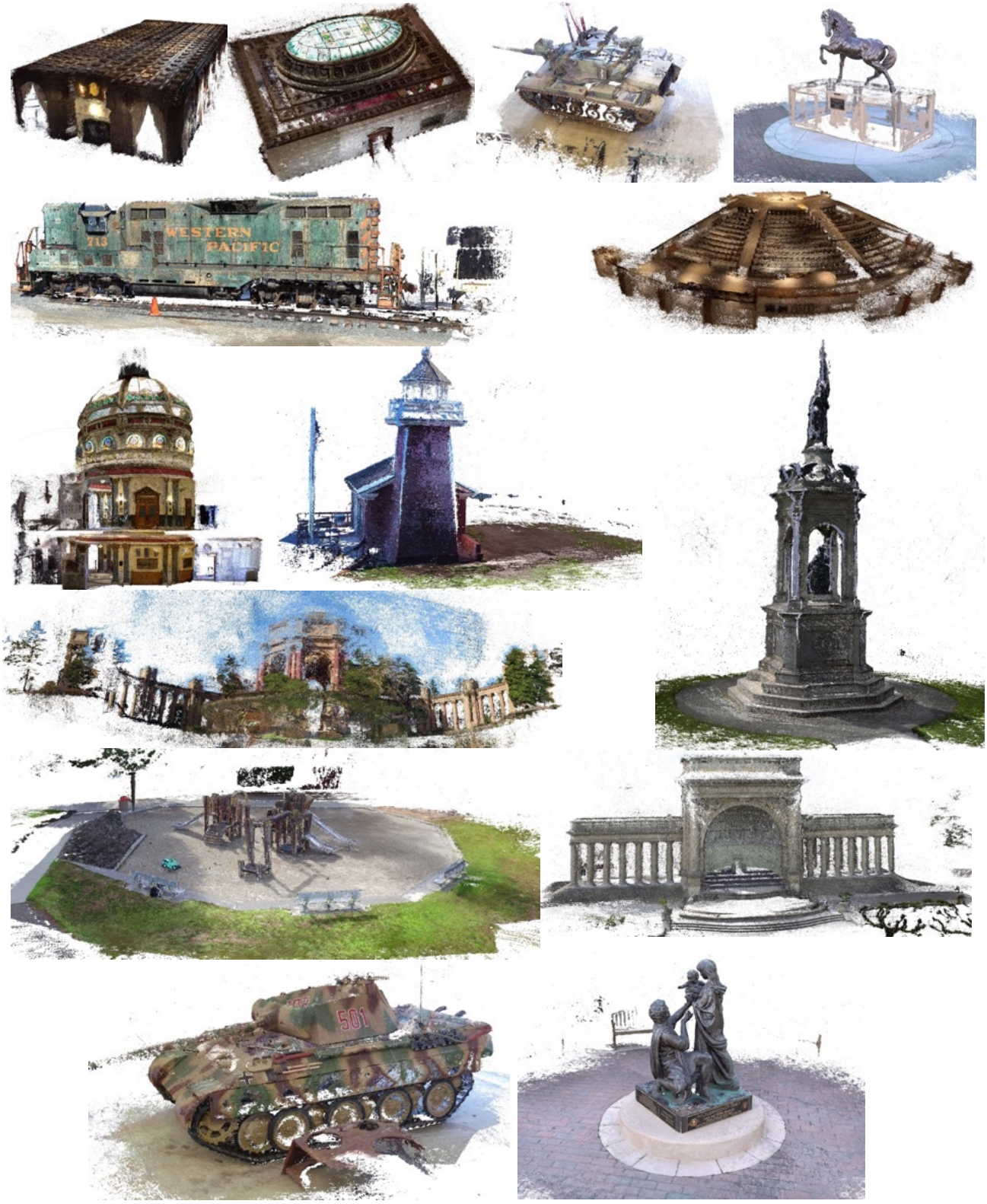

Figure 6. Point clouds reconstructed using GC-MVSNet for all scenes from Tanks and Temples [9] intermediate and advanced set.

Figure 7. Point clouds reconstructed using GC-MVSNet for all scenes from BlendedMVS [18] evaluation set.

References

[1] 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. 2

[2] François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Deep multi-view stereo gone wild. In *2021 International Conference on 3D Vision (3DV)*, pages 484–493. IEEE, 2021. 3

[3] Yikang Ding, Wentao Yuan, Qingtian Zhu, Haotian Zhang, Xiangyue Liu, Yuanjiang Wang, and Xiao Liu. Transmvsnet: Global context-aware multi-view stereo network with

transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8585–8594, 2022. 2, 3

[4] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 873–881, 2015. 3

[5] 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. 2, 3

[6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003. 2

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 2

[8] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 2, 3, 4

[9] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 3, 5

[10] Rui Peng, Rongjie Wang, Zhenyu Wang, Yawen Lai, and Ronggang Wang. Rethinking depth estimation for multi-view stereo: A unified representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[11] Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019. 2

[12] 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. 2, 3

[13] Rafael Weilharter and Friedrich Fraundorfer. Highres-mvsnet: A fast multi-view stereo network for dense 3d reconstruction from high-resolution images. *IEEE Access*, 9:11306–11315, 2021. 2

[14] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[15] 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. 2

[16] 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. 1, 2, 3

[17] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. *Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[18] 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. 3, 6

[19] Anzhu Yu, Wenyue Guo, Bing Liu, Xin Chen, Xin Wang, Xuefeng Cao, and Bingchuan Jiang. Attention aware cost volume pyramid based multi-view stereo network for 3d reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:448–460, 2021. 2

[20] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *British Machine Vision Conference (BMVC)*, 2020. 2