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

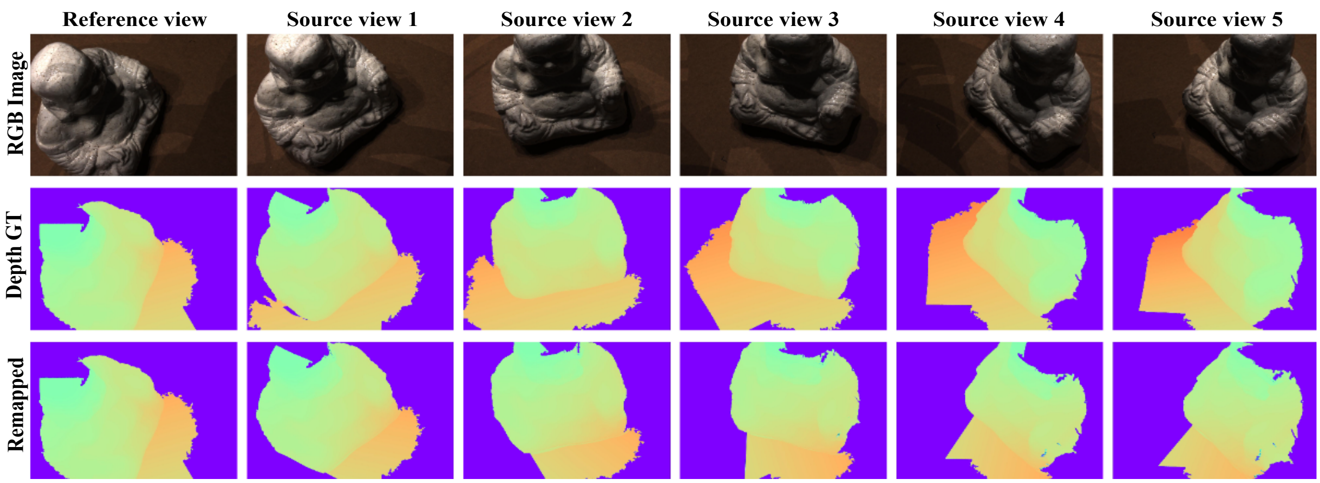

Figure 1: First row shows the selection of M closest source images for a given reference image. Middle row shows the corresponding ground truth depth maps and last row shows the remapped source ground truth depth maps using x-y coordinates of reference view projection to the source view. During remapping, 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

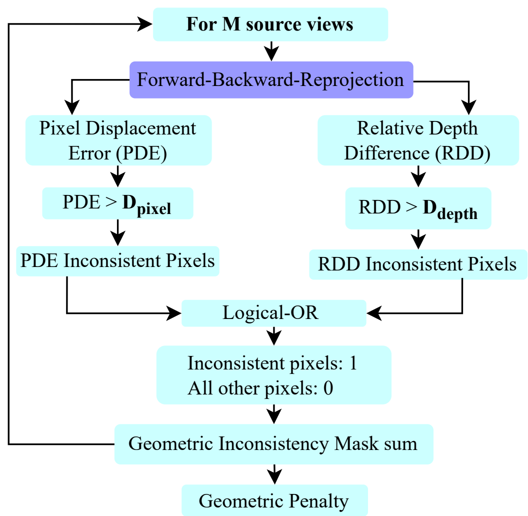

Figure 2. GC module flow-chart for consistency check.

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

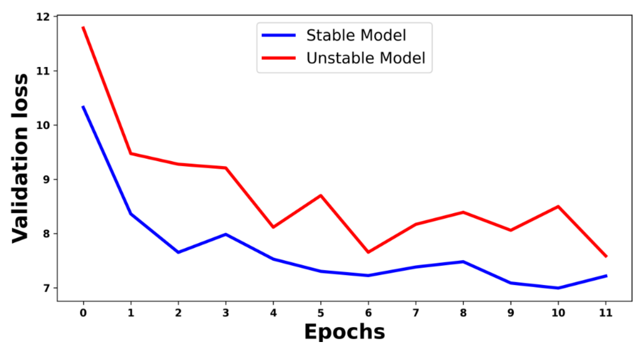

Figure 3. Validation loss on DTU [8] dataset during training. The red line shows the unstable model training, validation loss change in zig-zag manner. Blue line shows stable training with smooth change in validation loss.

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

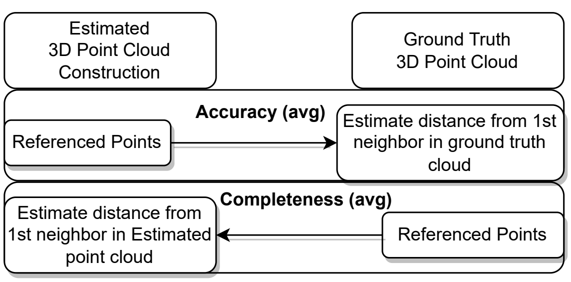

Figure 4. The process of calculating accuracy and completeness for DTU [8] point cloud evaluation.

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.

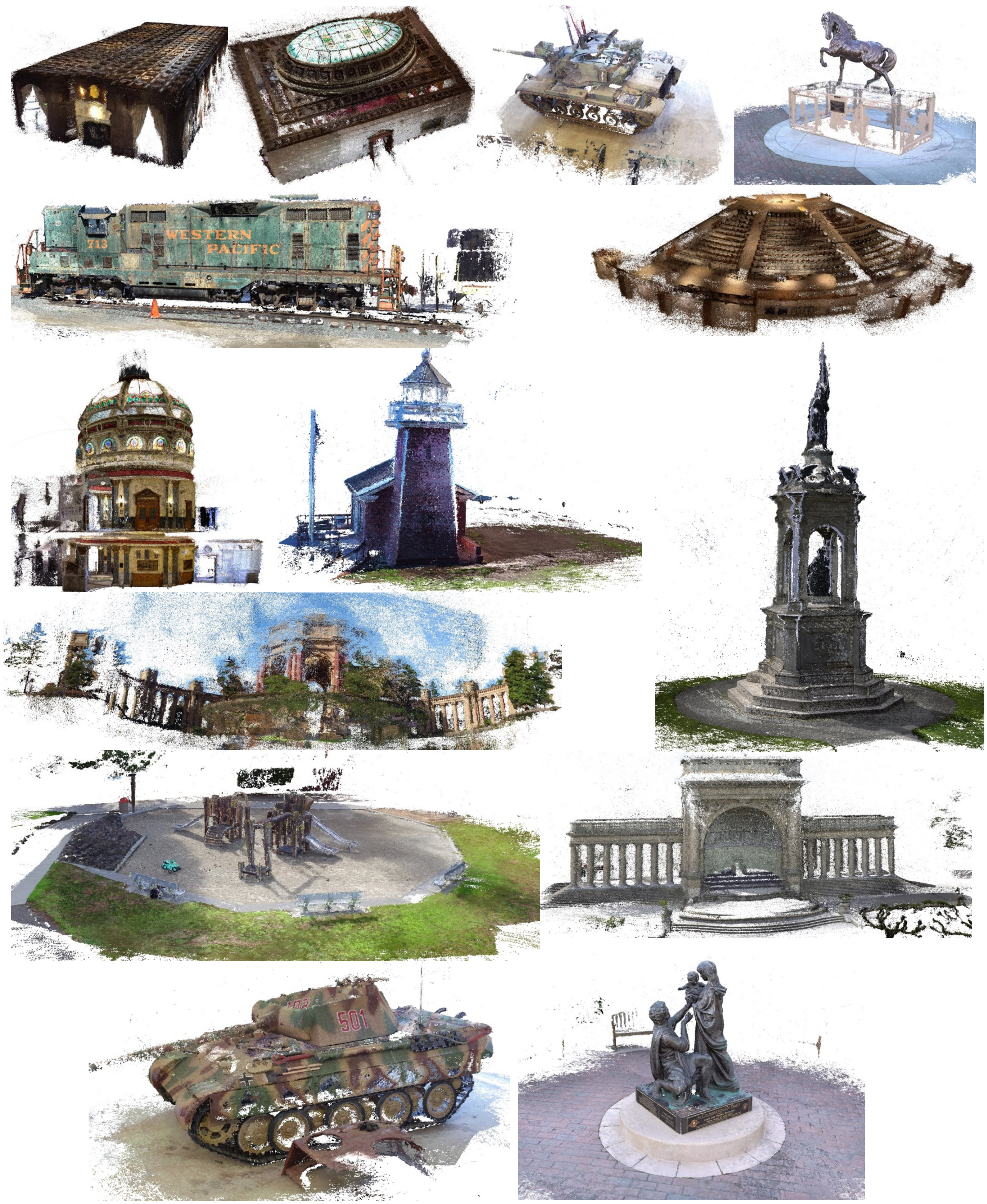

Figure 6. Point clouds reconstructed using GC-MVSNet for all scenes from Tanks and Temples [9] intermediate and advanced set.

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