MVS2D: Efficient Multi-view Stereo via Attention-Driven 2D Convolutions Supplementary Materials

Zhenpei Yang[1,*] Zhile Ren[2] Qi Shan[2] Qixing Huang[1]
[1]The University of Texas at Austin [2]Apple

1. Details of Experiments

1.1. Model Details

We use ResNet-18 (with the fully connected layer and pooling layer removed) as the building block of our networks $\mathcal{G}$ and $\mathcal{F}$. $\mathcal{F}$ additionally includes two up-sampling and convolution layer that output a feature map $F_0 \in \mathcal{R}^{\frac{h}{4}\times\frac{w}{4}\times c}$, where h and w are the input image's height and width respectively. F_0 is further feed into a small network with 3 convolutions and 3 de-convolutions to recover a depth probability volume $F_1 \in \mathcal{R}^{\frac{h}{4}\times\frac{w}{4}\times k'}$. F_1 is finally decoded into a depth map $d_{\frac{1}{4}} \in \mathcal{R}^{\frac{h}{4}\times\frac{w}{4}}$ using the soft-argmax operation as in MVSNet [10]. To supervise the network at full resolution instead of $\frac{1}{4}$ resolution, we further upsample $d_{\frac{1}{4}}$ to the full resolution via nearest interpolation, and add a predicted residual to it to get final prediction $d \in \mathcal{R}^{h\times w}$. Please refer to our released code for more details.

1.2. Training Details

We use 4 NVIDIA DGX-V100 GPUs with 32GB memory each to conduct following experiments.

ScanNet. We train for 30 epochs with a batch size 16, and reduce the learning rate by 10 at epoch 25 and 28. The input image size is 640×480. During training, we sample depth hypothesis uniformly in inverse depth space

$$d_i = 1/((1-\frac{i}{k})\frac{1}{d_{\text{min}}} + \frac{i}{k}\frac{1}{d_{\text{max}}}), \tag{1}$$

where k is the number of depth hypothesis. We set $k = 32$, $d_{\text{min}} = 0.3$, and $d_{\text{max}} = 10.1$.

We trained several baselines on ScanNet dataset. Specifically, we trained MVSNet [10] for 500k iterations. We trained FastMVSNet [11] for a total 30 epochs, with the last 10 epochs optimizing the GaussNewton layer. We trained NAS [5] for 20 epochs for initialization, and 10 epoch including the normal consistency module. We trained PatchmatchNet [9] for 30 epochs. We trained DPSNet [3] for 20 epochs. We trained Bts [6] for 20 epochs. The training for MVSNet/DPSNet costs around 3 days. The training for NAS costs around 5 days. The training time of FastMVSNet/PatchmatchNet costs around 1 day.

SUN3D, RGBD, and Scenes11. We follow similar setup as the training of ScanNet. During training, we sample depth hypothesis uniformly in inverse depth space:

$$d_i = 1/((1-\frac{i}{k})\frac{1}{d_{\text{min}}} + \frac{i}{k}\frac{1}{d_{\text{max}}}), \tag{2}$$

where $d_{\text{min}} = 0.5$, $d_{\text{max}} = 32.0$.

DTU. One modification we made on training DTU dataset is that we add a confidence prediction. The added confidence will be used to filter out unconfident predictions during the final 3D reconstruction. We adopt following loss [4]:

$$\mathcal{L} = \frac{|\hat{d} - d_{\text{gt}}|}{\hat{\sigma}} + \log(\hat{\sigma}), \tag{3}$$

where $\hat{\sigma}$ is the predicted confidence map. We implement this confidence prediction by adding a separate head in the final output. Such confidence prediction is trained along with depth prediction without the need for explicit supervision. During the test time, we set a threshold τ for $\hat{\sigma}$ and prune all predictions $\hat{d}_i$ whose corresponding $\hat{\sigma}_i$ are larger than τ.

We train for 80 epochs with a batch size 8, and reduce the learning rate by ten at epoch 40 and 70. The input image size is 1536×1152. The training takes around 5 days. We follow the practice of the robust training scheme [9] and use randomly chosen 4 reference images for each source image during training. We use $N = 96$ depth hypothesis uniformly sampled in the inverse depth space ($d_i = 1/((1-\frac{i}{k})\frac{1}{d_{\text{min}}} + \frac{i}{k}\frac{1}{d_{\text{max}}})$. We use $d_{\text{min}} = 425.0$, $d_{\text{max}} = 935.0$. During testing, we use top 4 reference images(ranked by a heuristic criterior introduced in [10]), which is the same as previous approaches [2, 9, 10]. We fuse the depth maps into final 3D model using the fusion code provided in [9]. The qualitative results on DTU test set can be found in Figure 5.

1.3. Speed Comparison

We use one V100 GPU to conduct the speed benchmark for all methods. to We feed each method an input image of size 640x480. For methods that do not produce full-resolution depth maps (i.e. MVSNet [10]), we further upsampled them to the full resolution using nearest-neighbor

Method	AbsRel ↓	AbsDiff ↓	SqRel ↓	RMSE ↓	RMSELog ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
COLMAP	0.384	0.843	1.26	1.480	0.500	0.482	0.663	0.840
DeMoN	0.311	1.330	19.970	2.607	0.247	0.641	0.902	0.967
DeepMVS	0.231	0.663	0.615	1.149	0.302	0.674	0.887	0.941
DPSNet	0.081	0.201	0.097	0.442	0.160	0.885	0.945	0.973
NAS	**0.068**	**0.168**	**0.056**	0.375	0.142	**0.905**	0.964	0.988
Ours-robust	0.100	0.231	0.057	**0.313**	**0.140**	0.895	**0.966**	**0.991**
Ours	0.108	0.271	0.130	0.513	0.184	0.860	0.939	0.973

Table 1. Depth evaluation results on the MVS dataset (trained on RGBD, SUN3D, and Scenes11). Please see Sec. 2.2 for discussion.

Method	AbsRel ↓	SqRel ↓	log10 ↓	RMSE ↓	RMSELog ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
MVSNet	0.154	0.125	0.067	0.478	0.212	0.779	0.927	0.973
NAS	0.134	0.094	0.064	0.434	0.190	0.789	0.932	0.979
Ours	**0.113**	**0.062**	**0.049**	**0.332**	**0.149**	**0.871**	**0.968**	**0.993**
Ours-robust	0.115	0.066	0.052	0.354	0.158	0.862	0.960	0.990

Table 2. Depth evaluation results on the SUN3D dataset (trained on ScanNet). Please see Sec. 2.2 for discussion.

interpolation. The FPS numbers are averaged over 500 random inputs for each method.

1.4. Pose Corruption

We use the following procedure to generate perturbations for input poses. Assume the ground truth relative pose is $T = [R\,|\,t]$ between the source image and a certain reference image. Firstly, we sample N points $\{p_k\}_{k=1}^N$ on corresponding ground-truth 3D point cloud of the source image. We use $N = 10$ in our experiments. Then, we project those N points into reference image using ground truth relative pose T and camera intrinsics $\mathcal{K}$ to get $\{\overline{p}_k\}_{k=1}^N$. We then perturb $\{\overline{p}_k\}_{k=1}^N$ by adding noise from a uniform distribution whose maximum value is 10 pixels. We solve a PnP problem [1] using $\{p_k\}_{k=1}^N$ and the perturbed pixels $\{\overline{p}_k\}_{k=1}^N$ to get the corrupted $\overline{T} = [\overline{R}\,|\,\overline{t}]$. We accept the perturbed $\overline{T}$ if the average pixel offset over the source image is less than 10 pixels. Otherwise, we set $\overline{T} = T$. We pre-compute all perturbations for all image pairs. Figure 1 shows the statistics of pose perturbations. Specifically, we plot the histogram of $\Delta R = \arccos(\frac{\mathrm{Tr}(\overline{R}R^{-1})-1}{2})$, and $\Delta t = \left\|\overline{t} - t\right\|_2$.

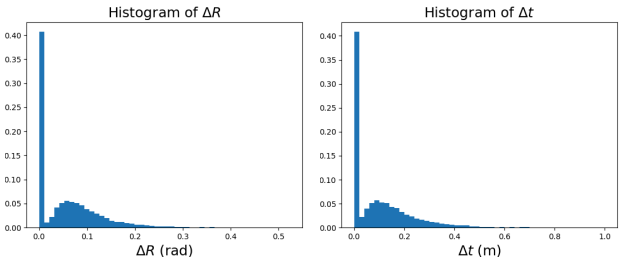

Figure 1. Pose corruption statistics. Left: histogram of rotation perturbation. Right: histogram of translation perturbation.

2. Additional Studies

2.1. Comparison with Video-based Method

We further compare our method with ESTD [7] which use a memory unit to accumulate information from the past frames. As shown in Table 3, although ESTD uses memory units that accumulate information from the past and 2 additional frames(5 v.s. 3), MVS2D still performs considerably better in both accuracy and speed.

Method	AbsRel	$\delta < 1.25$	FPS^A	FPS^B	Param(M)
ESTD (CVPR '21)	0.081	0.931	14.1	-	36.2
ESTD*	0.076	0.939	10.1	2.8	36.2
MVS2D (Ours)	**0.059**	**0.963**	**81.4**	**42.9**	**13.0**

Table 3. Evaluations on on ScanNet. The top row are numbers directly obtained from ESTD [8] paper, which use different testing split to ours thus is not directly comparable. ESTD* is tested on the same test split as ours. We report speed at ESTD's setting (FPS^A at input resolution 320x256) and our setting (FPS^B at 640x480, same as our main paper Tab. 1). The speed between ESTD and ESTD* is not comparable due to hardware differences. MVS2D surpasses ESTD both on speed and accuracy.

Metric	None	Layer1	Layer2*	Layer3	Layer4
AbsRel ↓	0.144	0.061	**0.055**	0.059	0.817
$\delta < 1.25 \uparrow$	0.631	0.874	**0.893**	0.886	0.814
RMSE ↓	0.269	0.146	**0.134**	0.139	0.180

Table 4. Ablation study on which layer to inject multi-view cues. The results are computed on ScanNet validation set. We can see inject on the second layer, which is used in our design, leads to the best results. The results on ScanNet test set are similar.

2.2. Generalization Ability

We did two experiments to evaluate the generalization ability of all methods to unseen datasets. Following the experimental setups of DeMoN and NAS, we use the model trained on SUN3D/RGBD/Scenes11 and test on the MVS dataset, which is an outdoor dataset and the data distributions are very different from the training set. The results can be found in Table 1. Our methods perform better than COLMAP, DeMoN and DeepMVS, although they still fall behind NAS under some metrics such as $AbsRel$. Such a result is reasonable since our approach is better adapted to the training distribution, which will lead to performance drop on heavily out-of-distribution test data.

To further examine each method's performance on unseen datasets whose input data statistics are similar to those in the training sets, we further test models trained using ScanNet on SUN3D test sets (see Table 2). The input data of SUN3D ScanNet are all indoor scenes. We can see that our methods still perform favorably among other methods.

2.3. Ablation Study on Mask Encoding

To study the benefits of mask encoding, we further experiment with **Ours-nomask** which removes the mask encoding in Eq 5. The results on ScanNet can be found in Table 5. Remove mask encoding (**Ours-nomask**) leads to worse results than **Ours**. Such behavior is reasonable since mask encoding provides an easy way for the network to distinguish the valid and invalid interpolation, thus facilitate the training.

Method	AbsRel ↓	$\delta < 1.25 \uparrow$	$thre@0.2 \uparrow$	$thre@0.5 \uparrow$
Ours-nomask	0.0605	0.9635	0.8543	0.9719
Ours	**0.0597**	**0.9640**	**0.8585**	**0.9735**

Table 5. Ablation study on mask encoding. $thre@0.2$/$thre@0.5$ measures the percentage of pixel that has absolute depth error less than 0.2m/0.5m respectively. Removing mask encoding hurts the performance, especially for *AbsRel*

2.4. More Qualitative Results

We show additional visualizations of depth predictions in Figure 2 & 3. Our method produces higher quality depth estimations compared to other MVS methods and performs better than one of the state-of-the-art single-view depth estimation method Bts. Additionally, we show more qualitative comparisons on 3D reconstruction on ScanNet in Figure 4. Our method generates comparable or even better visual results with other methods that require expensive 3D convolutions. We also show the reconstruction result on DTU test set in Figure 5.

References

[1] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001. 2

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

[3] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: End-to-end deep plane sweep stereo. In *International Conference on Learning Representations (ICLR)*, 2019. 1

[4] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1

[5] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[6] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 1

[7] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8258–8267, 2021. 2

[8] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[9] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14194–14203, 2021. 1, 7

[10] 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

[11] Zehao Yu and Shenghua Gao. Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1949–1958, 2020. 1

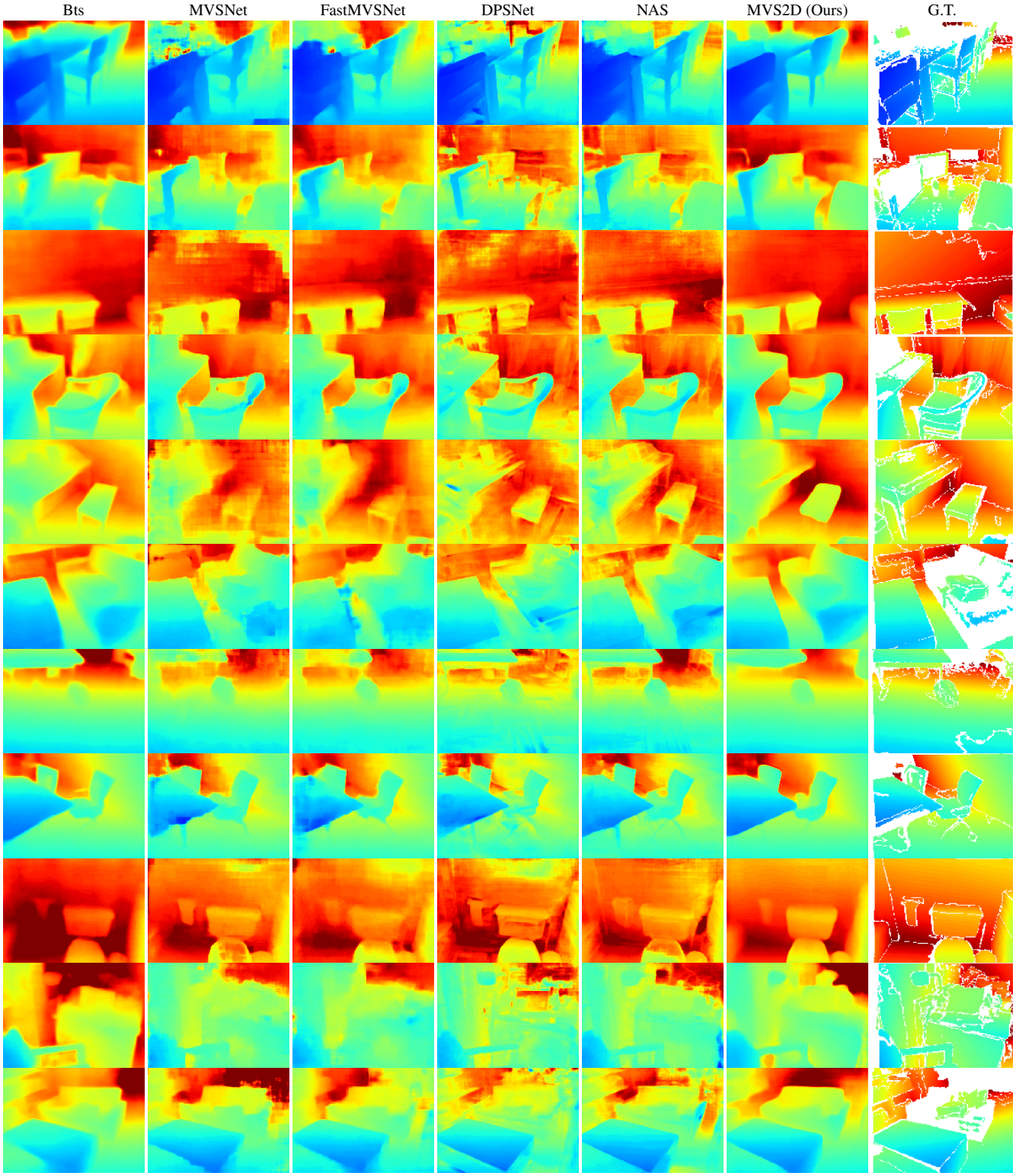

Figure 2. [1/2] Qualitative results on depth prediction. Each row corresponds to one test example. The region without ground truth depth labels is colored white in the last column. Our prediction outperforms both the single-view depth estimation method and other multi-view methods.

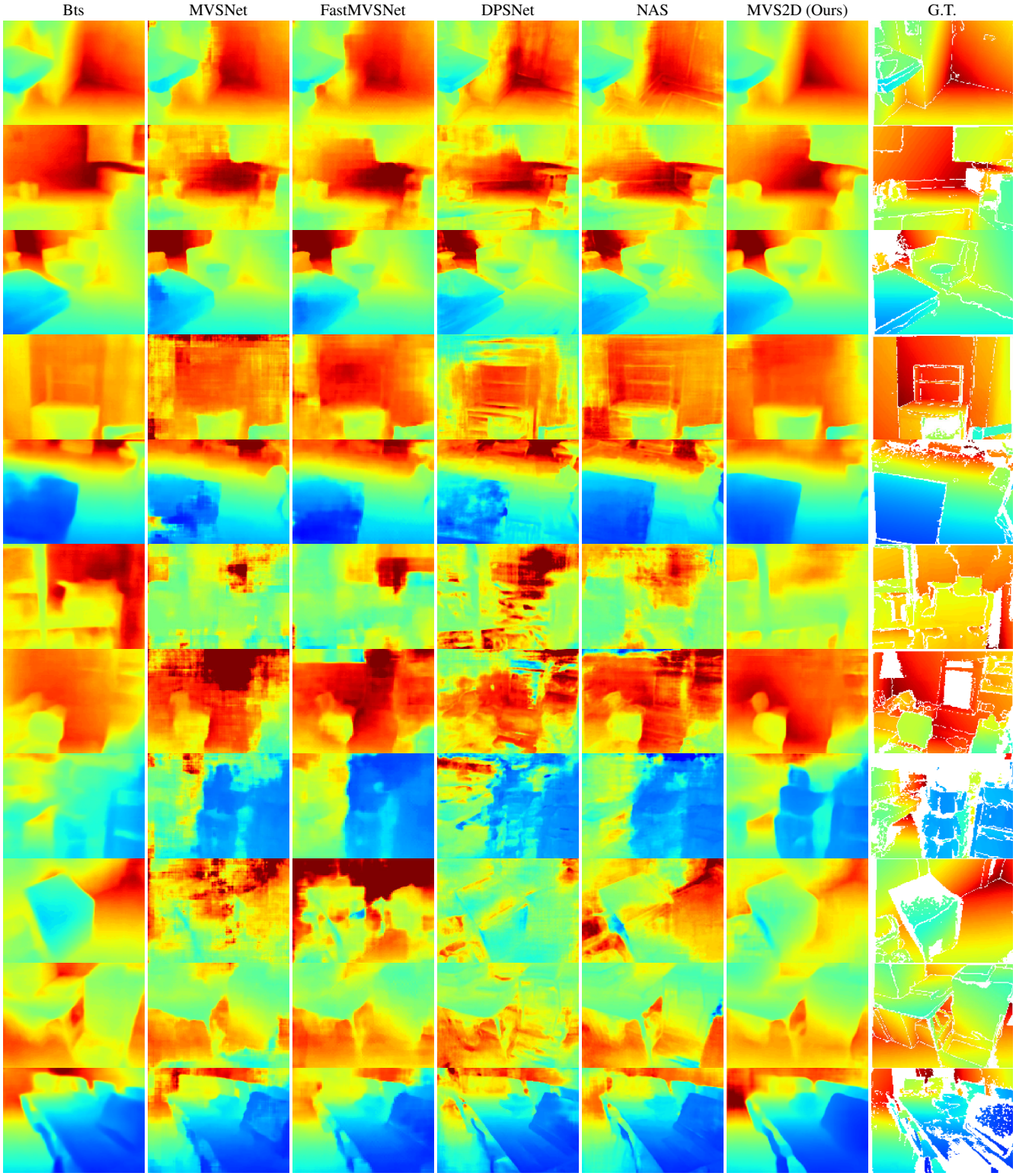

Figure 3. [2/2] Qualitative results on depth prediction. Each row corresponds to one test example. The region without ground truth depth labels is colored white in the last column. Our prediction outperforms both the single-view depth estimation method and other multi-view methods.

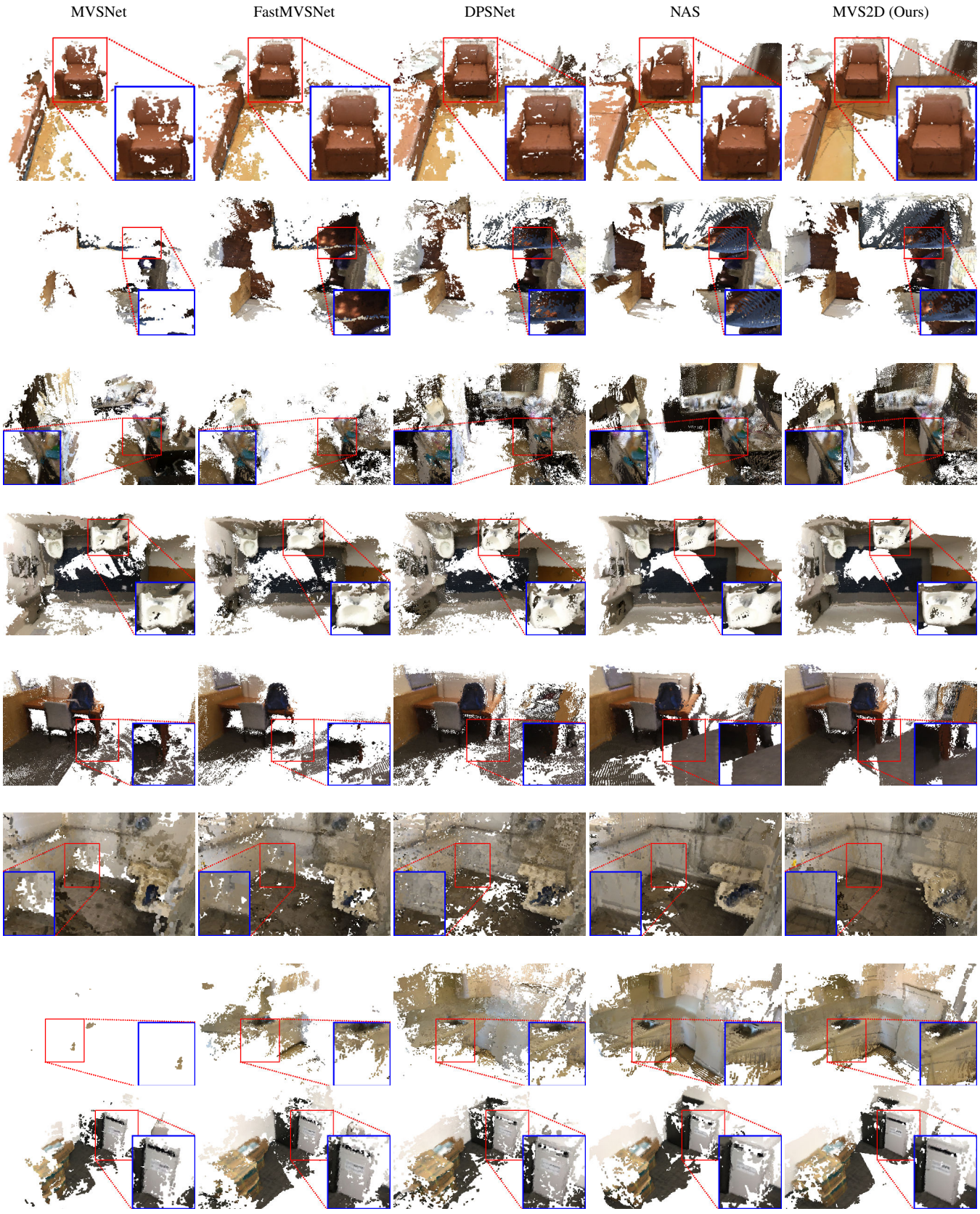

Figure 4. Qualitative scene reconstruction results on ScanNet. Our method yields smoother outputs than other baselines. We zoom in parts of a scene (red box) and show at the corner (blue box) to highlight the differences. Best viewed in PDF.

Figure 5. Qualitative 3D reconstruction results on the DTU test set. We use 4 reference views during the evaluation. The results are obtained from fusing multi-view depth prediction using the tool provided by PatchMatchNet [9].