BlendedMVS: A Large-scale Dataset for Generalized Multi-view Stereo Networks

Yao Yao[1] Zixin Luo[1] Shiwei Li[2] Jingyang Zhang[1] Yufan Ren[3] Lei Zhou[1]
Tian Fang[2] Long Quan[1]

[1]The Hong Kong University of Science and Technology
{yyaoag, zluoag, jzhangbs, lzhouai, quan}@cse.ust.hk

[2]Everest Innovation Technology
{sli, fangtian}@altizure.com

[3]Zhejiang University
renyufan@zju.edu.cn

Abstract

While deep learning has recently achieved great success on multi-view stereo (MVS), limited training data makes the trained model hard to be generalized to unseen scenarios. Compared with other computer vision tasks, it is rather difficult to collect a large-scale MVS dataset as it requires expensive active scanners and labor-intensive process to obtain ground truth 3D structures. In this paper, we introduce BlendedMVS, a novel large-scale dataset, to provide sufficient training ground truth for learning-based MVS. To create the dataset, we apply a 3D reconstruction pipeline to recover high-quality textured meshes from images of well-selected scenes. Then, we render these mesh models to color images and depth maps. To introduce the ambient lighting information during training, the rendered color images are further blended with the input images to generate the training input. Our dataset contains over 17k high-resolution images covering a variety of scenes, including cities, architectures, sculptures and small objects. Extensive experiments demonstrate that BlendedMVS endows the trained model with significantly better generalization ability compared with other MVS datasets. The dataset and pretrained models are available at https://github.com/YoYo000/BlendedMVS.

1. Introduction

Multi-view stereo (MVS) reconstructs the dense representation of the scene from multi-view images and corresponding camera parameters. While the problem is previously addressed by classical methods, recent studies [30, 31, 10] show that learning-based approaches are also able to produce results comparable to or even better than classical state-of-the-arts. Conceptually, learning-based approaches implicitly take into account global semantics such as specularity, reflection and lighting information during the reconstruction, which would be beneficial for reconstructions of textureless and non-Lambertian areas. It has been reported on the small object DTU dataset [2] that, the best overall quality has been largely improved by recent learning-based approaches [30, 31, 4, 13].

By contrast, leaderboards of Tanks and Temples [14] and ETH3D [23] benchmarks are still dominated by classical MVS methods. In fact, current learning-based methods are all trained on DTU dataset [2], which consists of small objects captured with a fixed camera trajectory. As a result, the trained model cannot generalize very well on other scenes. Moreover, previous MVS benchmarks [24, 26, 2, 14, 23] mainly focus on the point cloud evaluation rather than the network training. Compared with other computer vision tasks (e.g., classification and stereo), the training data for MVS reconstruction is rather limited, and it is desired to establish a new dataset to provide sufficient training ground truth for learning-based MVS.

In this paper, we introduce BlendedMVS, a large-scale synthetic dataset for multi-view stereo training. Instead of using expensive active scanners to obtain ground truth point clouds, we propose to generate training images and depth maps by rendering textured 3D models to different viewpoints. The texture mesh of each scene is first reconstructed from images, which is then rendered into color images and depth maps. To introduce the ambient lighting information during training, we further blend rendered images with input color images to generate the training input. The resulting images inherit detailed visual cues from rendered color images, which makes them consistently align with rendered depth maps. At the same time, the blended images still largely preserve the realistic ambient lighting information from input images, which helps the trained model better

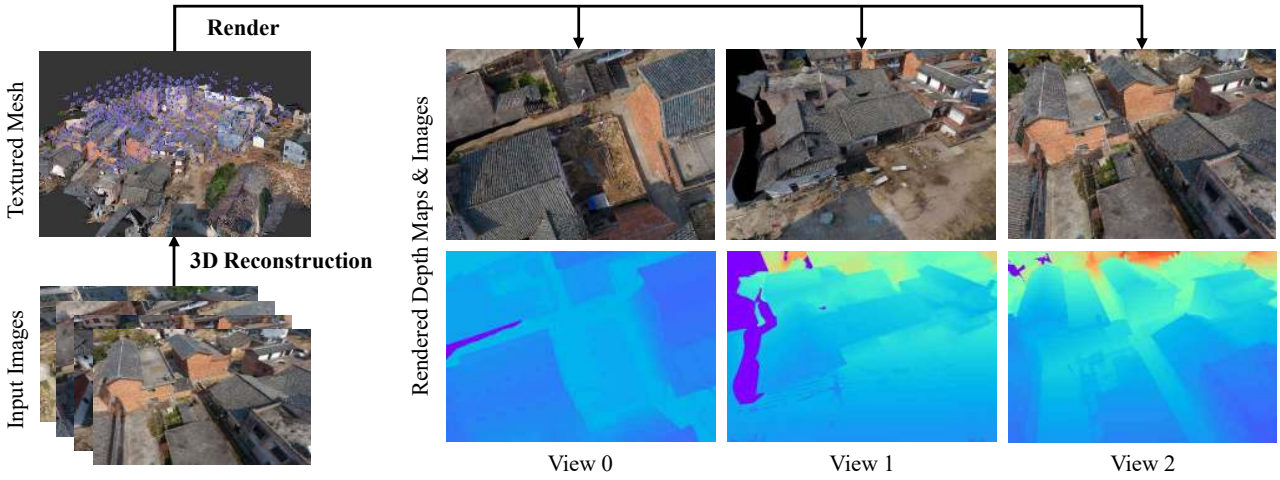

Figure 1: Pipeline of rendered data generation. We reconstruct the textured 3D model from input images, and then rendered the model into different view point to generate rendered images and depth maps.

generalize to real-world scenarios.

Our dataset contains 113 well selected and reconstructed 3D models. These textured models cover a variety of different scenes, including cities, architectures, sculptures and small objects. Each of the scene contains 20 to 1,000 input images and there are more than 17,000 images in total. We train recent MVSNet [30], R-MVSNet [31] and Point-MVSNet [4] on several MVS datasets. Extensive experiments on different validation sets demonstrate that models trained on BlendedMVS achieve better generalization ability compared with models trained on other MVS datasets.

Our main contributions can be summarized as:

- We propose a low-cost data generation pipeline with a novel fusion approach to automatically generate training ground truth for learning-based MVS.
- We establish the large-scale BlendedMVS dataset. All models in the dataset are well selected and cover a variety of diversified reconstruction scenarios.
- We report on several benchmarks that BlendedMVS endows the trained model with significantly better generalization ability compared with other MVS datasets.

2. Related Works

2.1. Learning-based MVS

Learning-based approaches for MVS reconstruction have recently shown great potentials. Learned multi-patch similarity [7] first applies deep neural networks for MVS cost metrics learning. SurfaceNet [10] and DeepMVS [8] unproject images to the 3D voxel space, and use 3D CNNs to classify if a voxel belongs to the object surface. LSM [11] and RayNet [18] encoded the camera projection to the network, and utilized 3D CNNs or Markov Random Field to predict surface label. To overcome the precision deficiency in volume presentation, MVSNet [30] applies differentiable homography to build the cost volume upon the camera frustum. The network applies 3D CNNs for the cost volume regularization and regress the per-view depth map as output. The follow-up R-MVSNet [31] is designed for high-resolution MVS, by replacing the memory-consuming the 3D CNNs with the recurrent regularization, and significantly reduce the peak memory size. More recently, Point-MVSNet [4] presents an point-based depth map refinement network, while MVS-CRF [29] introduces the conditional random field for the depth map refinement.

2.2. MVS Datasets

Middlebury MVS [24] is the earliest MVS dataset for MVS evaluation. It contains two indoor objects with low-resolution (640 $\times$ 480) images and calibrated cameras. Later, the EPFL benchmark [26] captures ground truth models of building facades and provides high-resolution images (6.2 MP) and ground truth point clouds for MVS evaluation. To evaluate algorithms under different lighting conditions, DTU dataset [2] captures images and point clouds for more than 100 indoor objects with a fixed camera trajectory. The point clouds are further triangulated into mesh models and rendered into different view point to generate ground truth depth maps [30]. Current learning-based MVS networks [30, 31, 4, 13] usually apply DTU dataset as their training data. Recent Tanks and Temples benchmark [14] captures indoor and outdoor scenes using high-speed video cameras, however, their training set only contains 7 scenes with ground truth point clouds. ETH3D benchmark [23] contains one low-resolution set and one high-resolution set. But similar to Tanks and Temples, ETH3D only provides a small number of ground truth scans for the network training. The available training data in these datasets is rather

limited, and a larger scale dataset is required to further exploit the potentials of learning-based MVS. In contrast, the proposed dataset will provide more than 17,000 images with ground truth depth maps, which covers a variety of diversified scenes and can greatly improve the generalization ability of the trained model.

2.3. Synthetic Datasets

Generating synthetic datasets for training is a common practice in many computer vision tasks, as a large amount of ground truth can be generated at very low cost. Thanks to recent advances in computer graphics, the rendering effect becomes increasingly photo-realistic, making the usage of synthetic datasets more plausible. For example, synthetic rendered images are used in stereo matching [3, 16, 32], optical flow [3, 16, 5], object detection [6, 27] and semantic segmentation [6, 19, 5, 20, 25]. Similar to these datasets, we consider incorporating the lighting effects in rendering synthetic datasets for 3D reconstruction. However, since it is difficult to generate correct material properties in different parts of the model, we resort to a blending approach with original images to recover the lighting effects.

3. Dataset Generation

The proposed data generation pipeline is shown in Fig. 1. We first apply a full 3D reconstruction pipeline to produce the 3D textured mesh from input images (Sec. 3.1). Next, the mesh is rendered to each camera view point to obtain the rendered image and the corresponding depth map. The final training image input is generated by blending the rendered image and input image in our proposed manner (Sec. 3.2).

3.1. Textured Mesh Generation

The first step to build a synthetic MVS dataset is generating sufficient high-quality textured mesh models. Given input images, we use Altizure online platform [1] for the textured mesh reconstruction. The software will perform the full 3D reconstruction pipeline and return the textured mesh and camera poses as final output.

With the textured mesh model and camera positions of all input images, we then render the mesh model to each camera view point to generate the rendered images and rendered depth maps. One example is shown in Fig. 1. The rendered depth maps will be used as the ground truth depth maps during training.

3.2. Blended Image Generation

Intuitively, rendered images and depth maps can be directly used for the network training. However, one potential problem is that rendered images do not contain view-dependent lightings. In fact, a desired training sample to multi-view stereo network should satisfy:

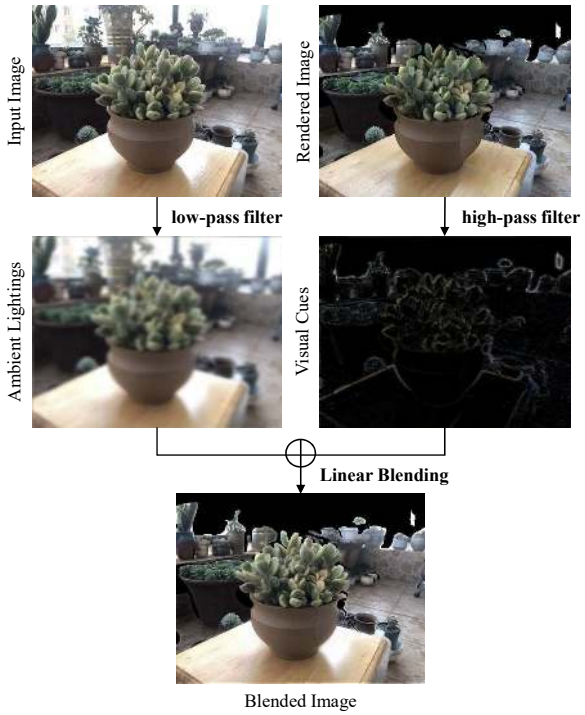

Figure 2: The blending process of the data generation pipeline. The high-pass filter is applied to extract image visual cues from the rendered image, while the low-pass filter is applied to extract ambient lightings from the input.

- Images and depth maps should be consistently aligned. The training sample should provide reliable mappings from input images to ground truth depth maps.

- Images should reflect view-dependent lightings. The realistic ambient lighting could strengthen model's generalization ability to real-world scenarios.

To introduce lightings to rendered images, one solution is to manually assign mesh materials and set up lighting sources during the rendering process. However, this is extremely labor-intensive, which makes it rather difficult to build a large-scale dataset.

On the other hand, the original input images have already contained the natural lighting information. The lighting could be automatically overlaid to rendered images if we can directly extract such information from input images. Specifically, we notice that ambient lightings are mostly **low-frequency** signals in images, while visual cues for establishing multi-view dense correspondences (e.g., rich textures) are mostly **high-frequency** signals in images. Following the observation, we propose to extract visual cues from the rendered image $\mathbf{I}_r$ using a high-pass filter $\mathbf{H}$, and extract the view-dependent lighting from the input image $\mathbf{I}$ using the low-pass filters $\mathbf{L}$. The visual cues and lightings

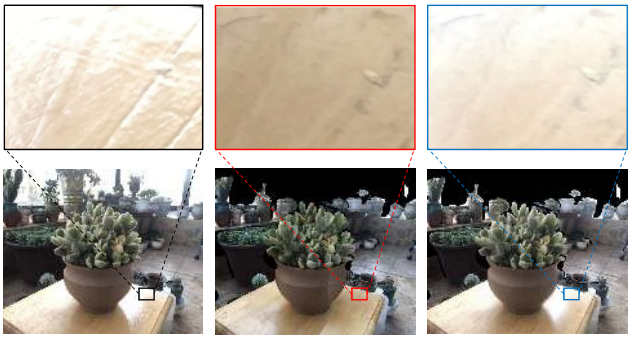

Figure 3: Detailed textures of input, rendered and blended images. The blended image has similar background lightings to the input image, while inherits texture details from the rendered image.

are fused to generate the blended image $\mathbf{I}_b$ (Fig. 2):

$$\begin{aligned}\mathbf{I}_b &= \mathbf{I}_r * \mathbf{H} + \mathbf{I} * \mathbf{L} \\ &= \mathcal{F}^{-1}\big(\mathcal{F}(\mathbf{I}_r) \cdot \mathbf{H}_f\big) + \mathcal{F}^{-1}\big(\mathcal{F}(\mathbf{I}) \cdot \mathbf{L}_f\big)\end{aligned} \tag{1}$$

where '$*$' denotes the convolution operation, '$\cdot$' the element-wise multiplication. The symbols $\mathcal{F}$ and $\mathcal{F}^{-1}$ are 2D Fast Fourier Transformation (FFT) and inverse FFT respectively. In our implementation, the filtering process is performed in the frequency domain. $\mathbf{L}_f$ and $\mathbf{H}_f$ are approached by 2D Gaussian low-pass and high-pass filters:

$$\mathbf{L}_f(u, v) = \exp\big(-\frac{(u^2 + v^2)}{2 \cdot D_0}\big) \tag{2}$$

$$\mathbf{H}_f(u, v) = 1 - \mathbf{L}_f(u, v) \tag{3}$$

The Gaussian kernel factor is empirically set to $D_0 = 5,000$ in our experiments. The blended image inherits detailed visual cues from the rendered image, while at the same time largely preserves realistic environmental lightings from the input image. Fig. 3 illustrates the differences between these three images. We will demonstrate in Sec. 5.2 that models trained with blended images have better generalization abilities to different scenes.

4. Scenes and Networks

4.1. Scenes

For the content of the proposed dataset, we manually select 113 well-reconstructed models publicly available in the Altizure.com online platform. These models cover a variety of different scenes, including architectures, street-views, sculptures and small objects. Each of the scene contains 20 to 1,000 input images, and totally there are 17,818 images in the whole dataset. It is also noteworthy that unlike DTU dataset [2] where all scenes are captured by a fixed robot arm, scenes in BlendedMVS contain a variety of different camera trajectories. The unstructured camera trajectories can better model different image capturing styles, and is able to make the network more generalizable to real-world reconstructions. Fig. 4 shows 7 scenes in BlendedMVS dataset with camera positions.

The dataset also provides training images and ground truth depth maps with a unified image resolution of $H \times W = 1536 \times 2048$. As input images are usually with different resolutions, we first resize all blended images and rendered depth maps to a minimum image size $H_s \times W_s$ such that $H_s >= 1536$ and $W_s >= 2048$. Then, we crop image patches of size $H \times W = 1536 \times 2048$ from the resized image centers to build training samples for BlendedMVS dataset. The corresponding camera parameters are changed accordingly. Also, the depth range is provided for each image as this information is usually required by depth map estimation algorithms.

Online Augmentation We also augment the training data during the training process. The following photometric augmentations are considered in our training: 1) Random brightness: we change the brightness of each image by adding a random value b such that $-50 < b < 50$, and then clip the image intensity value to the standard range of $0 - 255$. 2) Random contrast: we change the contrast of each image with a random contrast factor c such that $0.3 < c < 1.5$, and then clip the image to the standard range of $0 - 255$. 3) Random motion blur: we add the Gaussian motion blur to each input image. We consider a random motion direction and a random motion kernel size of $m = 1$ or 3 during the augmentation. The above mentioned augmentations will be imposed to each training image in a random order. In the ablation study section 5.2, we will demonstrate the improvement brought by the online augmentation.

4.2. Networks

To verify the effectiveness of the proposed dataset, we train and evaluate recent MVSNet [30], R-MVSNet [31] and Point-MVSNet [4] on BlendedMVS dataset.

MVSNet [30] is an end-to-end deep learning architecture for depth map estimation from multiple images. Given a reference image $\mathbf{I}_1$ and several source images $\{\mathbf{I}_i\}_{i=2}^N$, MVSNet first extract deep image features $\{\mathbf{F}_i\}_{i=1}^N$ for all images. Next, image features are warped into the reference camera frustum to build the 3D feature volumes $\{\mathbf{V}_i\}_{i=1}^N$ through the differentiable homographies. The network applies a variance-based cost metric to build the cost volume $\mathbf{C}$ and applies a multi-scale 3D CNNs for the cost volume regularization. The depth map $\mathbf{D}$ is regressed from the volume through the soft argmin [12] operation. The network is trained with the stander L1 loss function.

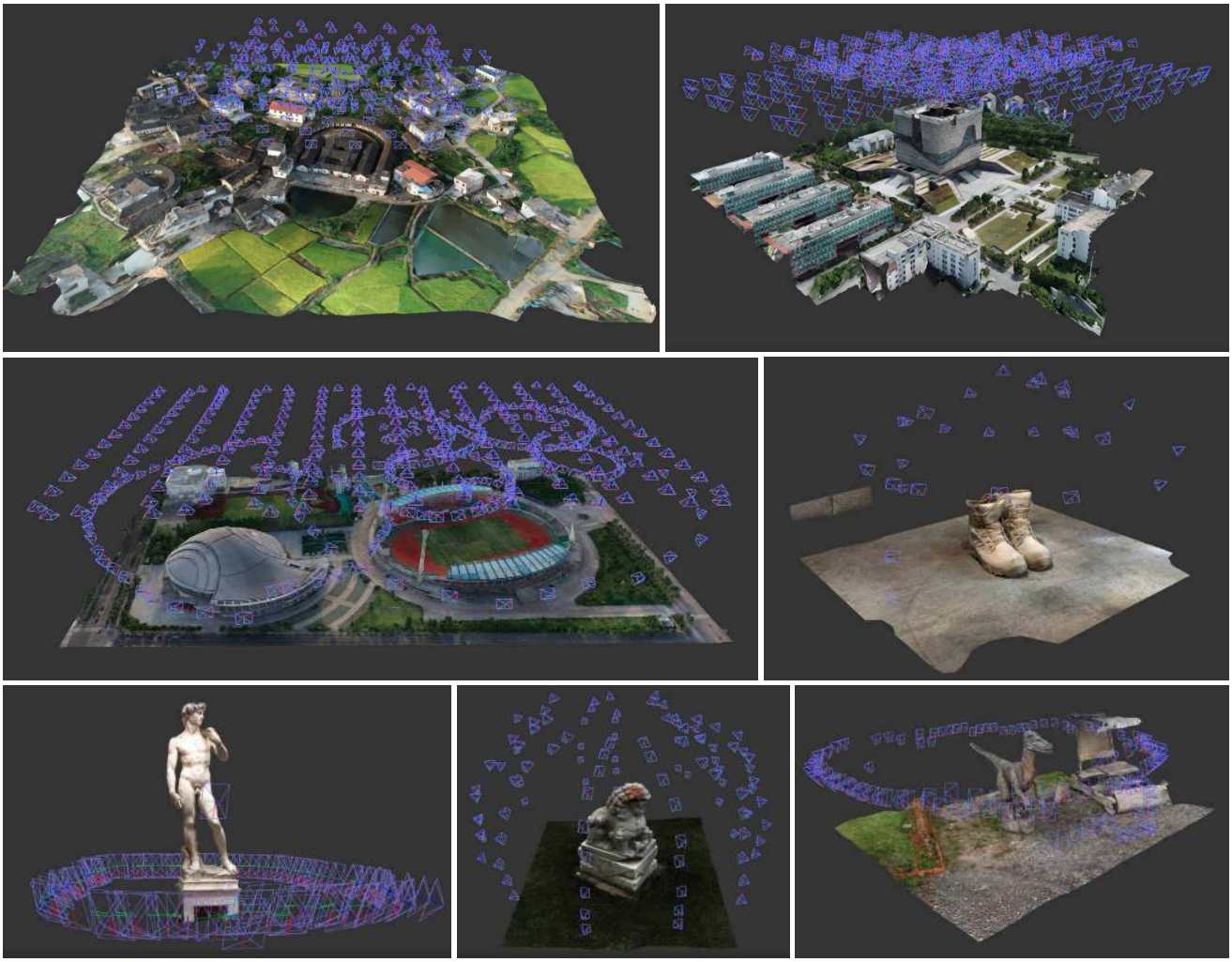

Figure 4: Several textured models with camera trajectories in BlendedMVS dataset. The blue box indicate the camera position in the 3D space. Our dataset contains 113 scenes in total.

R-MVSNet [31] is an extended version of MVSNet for high-resolution MVS reconstruction. Instead of regularizing the whole 3D cost volume $\mathbf{C}$ with 3D CNNs at once, R-MVSNet applies the recurrent neural network to sequentially regularize the 2D cost maps $\mathbf{C}(d)$ through the depth direction, which dramatically reduces the memory consumption for MVS reconstruction. Meanwhile, R-MVSNet treats depth map estimation as a classification problem and applies the cross-entropy loss during the network training.

Point-MVSNet [4] is a point-based deep framework for MVS reconstruction. In the network, the authors apply MVSNet framework to generate a coarse depth map, convert it into a point cloud and finally refine the point cloud iteratively by estimating the residual between the depth of the current iteration and that of the ground truth.

Implementations we directly use the open-source implementations of the three networks from their GitHub pages. Compared with the original papers, several modifications have been made to MVSNet and R-MVSNet: 1) The 5-layer 2D CNNs is replaced by a 2D U-Net to enlarge the receptive field during image feature extraction. 2) The batch normalization [9] is replaced with the group normalization [28] with fixed a group channel size of 8 to improve the network performance when training with small batch size. 3) The refinement network in MVSNet is removed as this part only brings limited performance gain. 4) The variational refinement step in R-MVSNet is removed so as to avoid the non-learning component affecting the dataset evaluation.

All models are trained using one GTX 2080 Ti GPU with $batchsize = 1$. MVSNet and R-MVSNet are trained for 160k iterations, while Point-MVSNet is trained for 320k iterations with the first 100k for coarse MVSNet initialization and another 220k for the end-to-end training.

5. Experiments

5.1. Quantitative Evaluation

5.1.1 Depth Map Validation

To demonstrate the capacity of BlendedMVS dataset, we compare models trained on 1) DTU training set, 2) ETH3D low-res training set, 3) MegaDepth dataset and 4) BlendedMVS training set. Evaluations are done on the corresponding validation sets. Three metrics are considered in our experiments: 1) the end point error (EPE), which is the average L_1 loss between the inferred depth map and the ground truth depth map; 2) the > 1 pixel error, which is the ratio of pixels with L_1 error larger than 1 depth-wise pixel; and 3) the > 3 pixel error. Quantitative results are shown in Fig. 5.

Trained on DTU [2] As suggested by previous methods [30, 10, 31], DTU dataset is divided into training, validation and evaluation sets. We train the three networks with a fixed input sample size of $H \times W \times D = 512 \times 640 \times 128$ and fixed depth range of $[d_{min}, d_{max}] = [425, 937]$.

It is reported in Fig. 5 that all three models trained on DTU (black lines) perform very well on DTU validation set, however, produce high validation errors in BlendedMVS and ETH3D datasets. In fact, models are overfitted in small-scale indoor scenes, showing the importance of having rich object categories in MVS training data.

Trained on ETH3D [23] The ETH3D training set contains 5 scenes. To separate the training and the validation, we take *delivery_area*, *electro*, *forest* as our training scenes, and *playground*, *terrains* as our validation scenes. The training sample size is fixed to $H \times W \times D = 480 \times 896 \times 128$. The per-view depth range is determined by the sparse point cloud provided by the dataset.

As shown in Fig. 5, validation errors of models trained on ETH3D (blue dash lines) are high in all validation sets including its own dataset, indicating that ETH3D training set does not provide sufficient data for MVS training.

Trained on MegaDepth [15] MegaDepth dataset is originally built for single-view depth map estimation that it applies multi-view depth map estimation to generate the depth training data. The dataset provides image-depthmap training pairs and SfM output files from COLMAP [21]. To apply MegaDepth for the MVS training, we apply the view seletion and the depth range estimation [30, 31] to generate training files in MVSNet format. Also, as reconstructed depth maps of crowdsourced images are usually incomplete, we only use those training samples with more than 20% valid pixels in the reference depth map during our training. There are 39k MVS training samples in MegaDepth dataset after the proposed pre-processing. The training input size is fixed to $H \times W \times D = 512 \times 640 \times 128$ by applying the resize-and-crop strategy as described in 4.1.

Although MegaDepth contains more training samples than BlendeMVS, models trained on MegaDepth (green dash lines in Fig. 5) are still inferior to models trained on BlendedMVS. We believe there are two major problems of applying MegaDepth for the MVS training: 1) the ground truth depth map is generated through MVS reconstructions. In this case, input images and reconstructed depth maps are not consistently aligned and the network will tend to overfit to the chosen algorithm [22]. 2) MegaDepth is built upon crowdsourced internet photos. The crowdsourced images are not well-captured and the training data quality could have significant influences on the training result.

Trained on BlendedMVS To train MVS networks with BlendedMVS, we resize all training samples to $H \times W = 576 \times 768$ for MVSNet and R-MVSNet, and further crop the samples to $H \times W = 448 \times 768$ for Point-MVSNet. The depth sample number is set to $D = 128$. Our dataset is also divided into 106 training scenes and 7 validation scenes to evaluate the network training.

As shown in Fig. 5, models trained on BlendedMVS (red lines) generalizes well to both DTU and ETH3D scenes. All models achieve the best validation results on BlendedMVS and ETH3D validation sets, and achieve the second best result (very close to the best) on DTU validation set, showing the strong generalization ability brought by our dataset.

5.1.2 Point Cloud Evaluation

We also compare point cloud reconstructions of models trained on DTU, ETH3D, MegaDepth and BlendedMVS on Tanks and Temples [14] training set. As the dataset contains wide-depth-range scenes that cannot be handled by MVSNet and PointMVSNet, we only test R-MVSNet (trained for 150k iterations) in this experiment. We follow methods described in R-MVSNet paper to recover camera parameters of input images, and then perform the per-view source image selection and depth range estimation based on the sparse point cloud. For post-processing, we also follow previous works [30, 31] to apply the visibility-based depth map fusion [17], average depth map fusion and visibility depth map filter to generate the 3D point cloud.

The dataset reports three evaluation metrics, namely *precision* (*accuracy*), *recall* (*completeness*) and the overall *f_score* [14, 23] to quantitatively measure the reconstruction quality. As shown in Table 1, R-MVSNet trained on DTU [2] and MegaDepth [15] achieve similar *f_score* performances, while R-MVSNet trained on the proposed dataset outperforms models trained on the other three datasets for all scenes. The average *f_score* is improved from 0.475 to 0.532 by simply replacing the training data from DTU to BlendedMVS. Qualitative comparisons on depth maps are shown in Fig. 6.

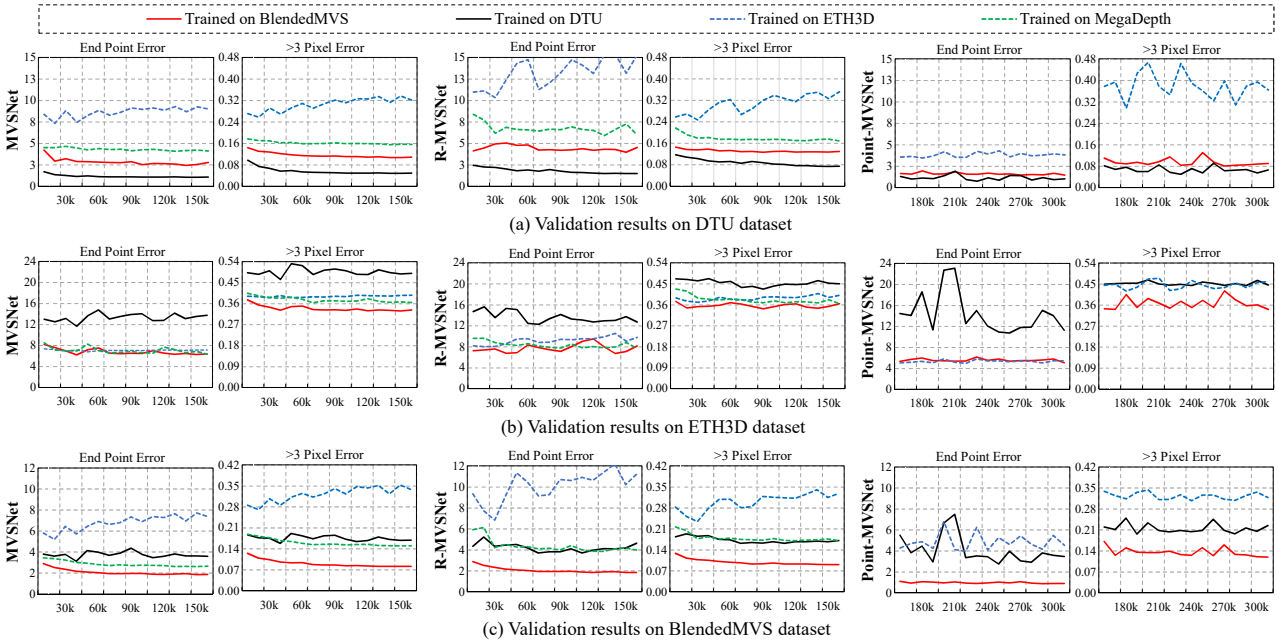

Figure 5: **Depth map validation errors** during the training process on all validation sets. Results of models trained on BlendedMVS (red lines) demonstrate good generalization ability on both DTU and ETH3D validation sets.

R-MVSNet Models	Metrics	Barn	Caterpillar	Church	Courthouse	Ignatius	Meetingroom	Truck	Average
Trained on DTU [2]	*Precision*	0.387	0.301	0.498	0.399	0.409	0.391	0.559	0.421
	Recall	0.674	0.755	0.313	0.731	0.856	0.213	0.846	0.623
	F_score	0.492	0.430	0.384	0.517	0.553	0.276	0.673	0.475
Trained on ETH3D [23]	*Precision*	0.334	0.297	0.497	0.347	0.362	0.324	0.492	0.379
	Recall	0.564	0.608	0.221	0.598	0.750	0.112	0.706	0.508
	F_score	0.420	0.399	0.306	0.439	0.488	0.166	0.580	0.400
Trained on MegaDepth [15]	*Precision*	0.414	0.291	0.566	0.441	0.408	0.418	0.522	0.437
	Recall	0.676	0.724	0.282	0.741	0.854	0.152	0.815	0.606
	F_score	0.513	0.415	0.376	0.553	0.552	0.223	0.636	0.467
Trained on BlendedMVS	*Precision*	0.432	0.352	0.570	0.462	0.492	0.444	0.602	0.479
	Recall	0.715	0.770	0.387	0.765	0.901	0.251	0.845	0.662
	F_score	**0.539**	**0.484**	**0.461**	**0.577**	**0.636**	**0.321**	**0.703**	**0.532**

Table 1: **Point cloud evaluations** on *Tanks and Temples* training set [14]. R-MVSNet trained on BlendedMVS outperforms models trained on other datasets in all scenes.

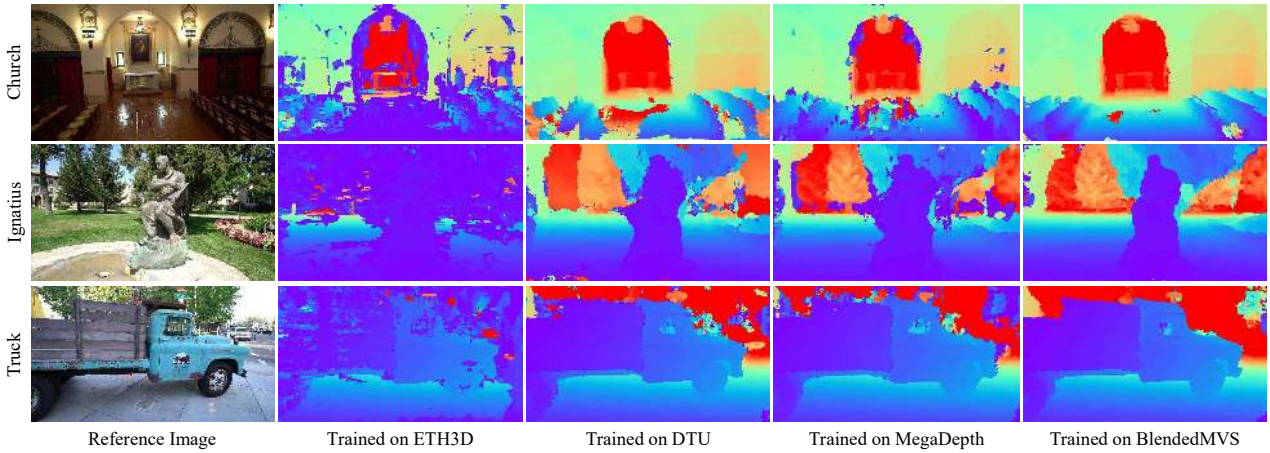

Figure 6: Qualitative comparisons on depth map reconstructions using R-MVSNet [31]. The model trained on BlendedMVS generates much cleaner results than models trained on the other three datasets.

Networks	Training Images	EPE	<1 Px. Err	<3 Px. Err
MVSNet [30]	Rendered	2.99	0.245	0.136
	Input	3.70	0.243	0.135
	Blended	2.88	0.224	0.118
	Rendered+Aug.	2.94	0.225	0.116
	Input+Aug.	3.16	0.234	0.123
	Blended+Aug.	**2.53**	**0.219**	**0.107**
R-MVSNet [31]	Rendered	5.54	0.251	0.148
	Input	4.47	0.242	0.134
	Blended	5.77	0.239	0.137
	Rendered+Aug.	5.10	0. 238	0.132
	Input+Aug.	**3.86**	0.241	**0.126**
	Blended+Aug.	3.95	**0.234**	**0.127**

Table 2: Ablation study on using different images for training. Validation errors on DTU dataset [2] show that blended images with online augmentation produces the best result.

5.2. Ablation Study on Training Image

Next, we study the differences of using 1) input images, 2) rendered images and 3) blended images as our training images. For these three setting, we also study the effectiveness of the online photometric augmentation. All models are trained for 150k iterations and are validated on DTU validation set. Comparison results are shown in Table 2.

Environmental Lightings The proposed setting of blended images with photometric augmentation produces the best result, while rendered images only produces the worst result among all. Also, all images with photometric augmentation results in lower validation errors than without, showing that view-dependent lightings are indeed important for MVS network training.

Training with Input Images It is noteworthy that while input images are not completely consistent with rendered depth maps, training R-MVSNet with input images (with or without the augmentation) also produces satisfying results (Table 2). The reason might be that 3D structures have been correctly recovered for most of the scenesc as all scenes are well-selected in advance. In this case, rendered depth maps can be regarded as the semi ground truth given input images, which could be jointly used for MVS network training.

5.3. Discussions

Imperfect Reconstruction One concern about using the reconstructed model for the MVS training is whether defects or imperfect reconstructions in textured models would affect the training process. In fact, blended images inherit detailed visual cues from rendered images, which are always consistent with rendered depth maps even if defects occur. In this case, the training process will not be deteriorated.

For the same reason, we could change the Altizure online platform to any other 3D reconstruction pipelines to recover the mesh model. What we have presented is a low-cost MVS training data generation pipeline that does not rely on any particular textured model reconstruction method.

Input Image Blended Image

Figure 7: Privacy preserving with blended images. Humans will be removed or blurred in the blended images.

Occlusion and Normal Information While current learning-based approaches [30, 31, 4, 13] does not take into account the pixel-wise occlusion and normal information, our dataset provides such ground truth information as well. The occlusion and normal information could be useful for future visibility-aware and patch-based MVS networks.

Privacy Using blended images could also help preserve the data privacy. For example, pedestrians in input images are usually dynamic, which will not be reconstructed in the textured model and rendered images (first row in Fig. 7). Furthermore, if pedestrians appear in front of the reconstructed object, our image blending process will only extract blurred human shapes from the input image, which helps conceal user identities in the blended image (second row in Fig. 7).

6. Conclusion

We have presented the BlendedMVS dataset for MVS network training. The proposed dataset provides more than 17k high-quality training samples covering a variety of scenes for multi-view depth estimation. To build the dataset, we have reconstructed textured meshes from input images, and have rendered these models into color images and depth maps. The rendered color image has been further blended with the input image to generate the training image input. We have trained recent MVS networks using BlendedMVS and other MVS datasets. Both quantitative and qualitative results have demonstrated that models trained on BlendedMVS achieve significant better generalization abilities than models trained on other datasets.

7. Acknowledgments

This work is supported by Hong Kong RGC GRF 16206819, Hong Kong RGC GRF 16203518 and Hong Kong T22-603/15N. We thank Rui Chen for helping train and validate PointMVSNet [4] in our dataset.

References

[1] Altizure: Mapping the world in 3d. https://www.altizure.com.

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

[3] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 2012.

[4] Rui Chen, Songfang Han, Jing Xu, and Hao Su. Point-based multi-view stereo network. In *International Conference on Computer Vision (ICCV)*, 2019.

[5] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Computer Vision and Pattern Recognition*, 2016.

[6] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *International Conference on Computer Vision (ICCV)*, 2017.

[8] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.

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

[11] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[12] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, and Peter Henry. End-to-end learning of geometry and context for deep stereo regression. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Luo Keyang, Guan Tao, Ju Lili, Huang Haipeng, and Luo Yawei. Pmvsnet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *International Conference on Computer Vision (ICCV)*, 2019.

[14] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. In *ACM Transactions on Graphics (TOG)*, 2017.

[15] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] Paul Merrell, Amir Akbarzadeh, Liang Wang, Philippos Mordohai, Jan-Michael Frahm, Ruigang Yang, David Nistér, and Marc Pollefeys. Real-time visibility-based fusion of depth maps. In *International Conference on Computer Vision (ICCV)*, 2007.

[18] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[19] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, 2016.

[20] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Computer Vision and Pattern Recognition*, 2016.

[21] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[23] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. 2017.

[24] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[25] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[26] Christoph Strecha, Wolfgang von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[27] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018.

[28] Yuxin Wu and Kaiming He. Group normalization. In *European Conference on Computer Vision (ECCV)*, 2018.

[29] Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao. Mvscrf: Learning multi-view stereo with conditional random fields. In *International Conference on Computer Vision (ICCV)*, 2019.

[30] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2018.

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

[32] Yi Zhang, Weichao Qiu, Qi Chen, Xiaolin Hu, and Alan Yuille. Unrealstereo: Controlling hazardous factors to analyze stereo vision. In *International Conference on 3D Vision)*, 2018.