Flow-Motion and Depth Network for Monocular Stereo and Beyond –Supplementary Material–

Kaixuan Wang and Shaojie Shen

https://github.com/HKUST-Aerial-Robotics/Flow-Motion-Depth

***Abstract*— This supplementary material provides more details about the proposed method, training, and dataset. More qualitative results of the experiments are provided as a supplement to the main letter. To demonstrate the generalization ability of the network and the effectiveness of the GTA-SfM dataset, the trained network is also directly applied to unseen scenes. Finally, we discuss the advantages and limitations of the proposed method.**

I. Network Details

In this section, we introduce more details in the flow-motion network and depth network. The depth observability of the epipolar point is also discussed in this section.

A. Flow Regularization

With an estimated optical flow $\mathbf{w}$, a regularized flow $\mathbf{w}_r$, is calculated such at the corresponding pixels are constrained on epipolar lines. For each pixel $\mathbf{x}_s$ on the source image, the regularized flow $\mathbf{w}_r(\mathbf{x}_s)$ is calculated by

$$\begin{aligned} \underset{\mathbf{w}_r(\mathbf{x}_s)}{\arg\min} \quad & \|\mathbf{w}_r(\mathbf{x}_s) - \mathbf{w}(\mathbf{x}_s)\|_2 \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{x}_s + \mathbf{w}_r(\mathbf{x}_s) \\ 1 \end{bmatrix}^T F \begin{bmatrix} \mathbf{x}_s \\ 1 \end{bmatrix} = 0 \end{aligned}. \tag{1}$$

$\mathbf{w}_r(\mathbf{x}_s)$ can be solved as Equation 3 in the main letter.

B. Pixel Search Space

$F[\mathbf{x}_s, 1]^T$ is the epipolar line on the target image. For notational simplicity, let $[e_x, e_y, e_z]^T = F[\mathbf{x}_s, 1]^T$. $\mathbf{h} = [e_y, -e_x]/(e_x^2 + e_y^2)$ is the normalized vector of the epipolar line direction. One the contrary, $\mathbf{v} = [e_x, e_y]/(e_x^2 + e_y^2)$ is the vector perpendicular to $\mathbf{h}$. With $\mathbf{h}$ and $\mathbf{v}$, we define the search space in the target feature map $\mathbf{f}_t$ as Equation 4 in the main letter.

C. Depth Observability

The epipolar point in the source image is $\mathbf{x}_e = \lambda(-KR^{-1}\mathbf{t})$. For any pixel depth d, the point is projected onto the target image as the same point,

$$\begin{aligned} \mathbf{x}_t &= \lambda(KRK^{-1}[\mathbf{x}_e, 1]^T \cdot d + K\mathbf{t}) \\ &= \lambda(KRK^{-1}[\lambda(-KR^{-1}\mathbf{t}), 1]^T \cdot d + K\mathbf{t}), \\ &= \lambda(K\mathbf{t}) \end{aligned} \tag{2}$$

thus the depth d is unobservable.

All authors are with the Department of ECE, HKUST, Hong Kong, China. {kwangap, eeshaojie}@ust.hk

For the stereo configuration, $\mathbf{x}_e$ is at infinity so that all pixels in the image can be triangulated. However, in some unconstrained SfM cases, the epipolar point is on the image and cannot be triangulated directly. DeMoN [1] uses networks to refine triangulated depth maps with NaN values set to 0. Li *et. al.* [2] set pixels around the camera epipolar to zero. We believe the proposed triangulation layer is an alternative solution.

II. Training

Since the proposed method decouples the two-view SfM problem into flow-motion estimation and depth triangulation, we train the two networks separately. The flow-motion network is first trained and then is used to train the depth network with weight fixed. Adam optimizer [3] is used and the initial learning rate is set to 1e-4. We half the learning rate when the error plateaus. Only color augmentation is used in the training.

III. GTA-SfM dataset

In this letter, we propose a GTA-SfM dataset which is used for the network training. A similar dataset, MVS-Synth, is also rendered in GTA5 environment. In MVS-Synth, cameras usually move randomly with small translations. On the contrary, in the proposed dataset, the trajectory is manually defined that cameras move in large translations and rotations.

We provide samples of the dataset in Figure 1. As shown in the figure, the proposed dataset is more similar to that of SfM applications.

IV. Experiments

A. Two-view Evaluation

Pose Estimation Camera poses are estimated using optical flow in different resolutions. Here, we study the pose estimation quality together with the corresponding optical flow quality. The result is shown in Table I. At finer pyramid levels, the error of both optical flow and camera pose estimation decreases. However, contrary to the experience from classic SfM methods, the camera pose estimation is still better than SIFT even the optical flow resolution is 40×32. This can be explained by the dense pixel correspondences from the optical flow.

Depth Estimation As shown in Figure 2, we provide more qualitative results of our method on the DeMoN dataset. In MVS sequence, the depth net has difficult estimating

Fig. 1. Compare the proposed GTA-SfM dataset and previous MVS-Synth dataset. The trajectory of the camera in the proposed dataset contains large translation and rotation.

TABLE I

CAMERA POSE AND OPTICAL FLOW ESTIMATION QUALITY AT DIFFERENT PYRAMID LEVELS.

Level	Resolution	Flow Err.	Rot. Err.	Trans. Err.
0	160×128	3.168	1.491	8.517
1	80×64	3.269	1.566	8.559
2	40×32	3.707	1.776	10.630

the structure of trees such as the second, and the fourth row. These trees usually have very complex structures and introduce occlusions. Such structures and are difficult even for offline methods (the ground truth also misses the trees).

B. Depth Fusion Evaluation

Qualitative Results Multiview images bring more structure information of the scene, thus the depth map estimation can be robust and accurate. Figure 3 illustrates the quality of the depth estimation given different numbers of target images. The depth estimation improves when more target images are given to the depth network. Take the sample (b) and (d) as examples, the fine structures of trees and poles are well recovered by fusing multiple image pairs.

Runtime Comparison Table II shows the runtime comparison between our method and DeepMVS [4] given different numbers of target images (all measured with the same resolution). As shown in the table, our method is much more efficient compared with DeepMVS and scales well w.r.t the number of target images: from 2 target images to 6 target images, the time grows by 26% in our method and 118% in DeepMVS.

TABLE II

ESTIMATION TIME COMPARISON

Target Image Num.	2	3	4	5	6
Ours (**ms**)	49	53	53	57	62
DeepMVS (**s**)	11	14	17	21	24

Quantitative Results Image pairs are *randomly* sampled to compare the performance of our method and DeepMVS. Each source image is observed by three target images. DeepMVS is provided with ground truth camera poses and takes more time to estimate depth maps. The results are shown in Figure 4. At the right side of each sample, we calculate the L1-inv, sc-inv, L1-rel error of the estimated depth maps. Since depth maps from DeepMVS contain many outliers, we remove the maximum and minimum disparities before the evaluation. As shown in the figure, our method estimates smooth depth maps and is more accurate in most of the cases.

Generalization Ability To demonstrate the generalization ability of the model and the effectiveness of the proposed GTA-SfM dataset, we apply the GTA-SfM trained models directly to images from real worlds and Google Earth. Figure 5 and Figure 6 shows the estimated depth maps and point clouds of aerial photographs and indoor images. In Figure 7 and Figure 8, the trained model is further applied to images collected in Google Earth. Both architectures and natural scenes from different locations are covered in the experiment. As shown in the figures, even trained with synthetic images, our method can estimate depth maps from unseen scenes.

V. ADVANTAGE AND LIMITATION

Key to the proposed method is the carefully designed flow-motion network. The high-quality optical flow and camera motion enable accurate and efficient depth triangulation. On the other hand, many prior works (e.g., LS-Net [5] and CodeSLAM [6]) estimate the depth maps and camera poses by iteratively minimizing the reprojection error. Such refinements are prone to local minimums and brightness changes in the images. We have demonstrated that the proposed method generates accurate camera poses and depth maps with less forward-time.

Although achieving state-of-the-art results, the proposed method relies on high-quality optical flow estimation thus occlusion is challenging for the method. In Figure 9, we show the occlusion problem in the MVS sequence. Another reason that makes such complex occlusion difficult for the network is the supervision missing from the ground truth depth maps, which also motivates us to propose the GTA-SfM dataset such that networks can be correctly trained and evaluated.

REFERENCES

[1] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: Depth and motion network for learning monocular stereo. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[2] Z. Li, T. Dekel, F. Cole, R. Tucker, N. Snavely, C. Liu, and W. Freeman. Learning the depths of moving people by watching frozen people. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] D. P. Kingma and J. L. Ba. ADAM: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2015.

[4] P. Huang, K. Matzen, J. Kopf, N. Ahuja, and J. Huang. DeepMVS: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[5] R. Clark, M. Bloesch, J. Czarnowski, S. Leutenegger, and A. J. Davison. Learning to solve nonlinear least squares for monocular stereo. In *European Conference on Computer Vision (ECCV)*, 2018.

[6] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. CodeSLAM learning a compact, optimisable representation for dense visual SLAM. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

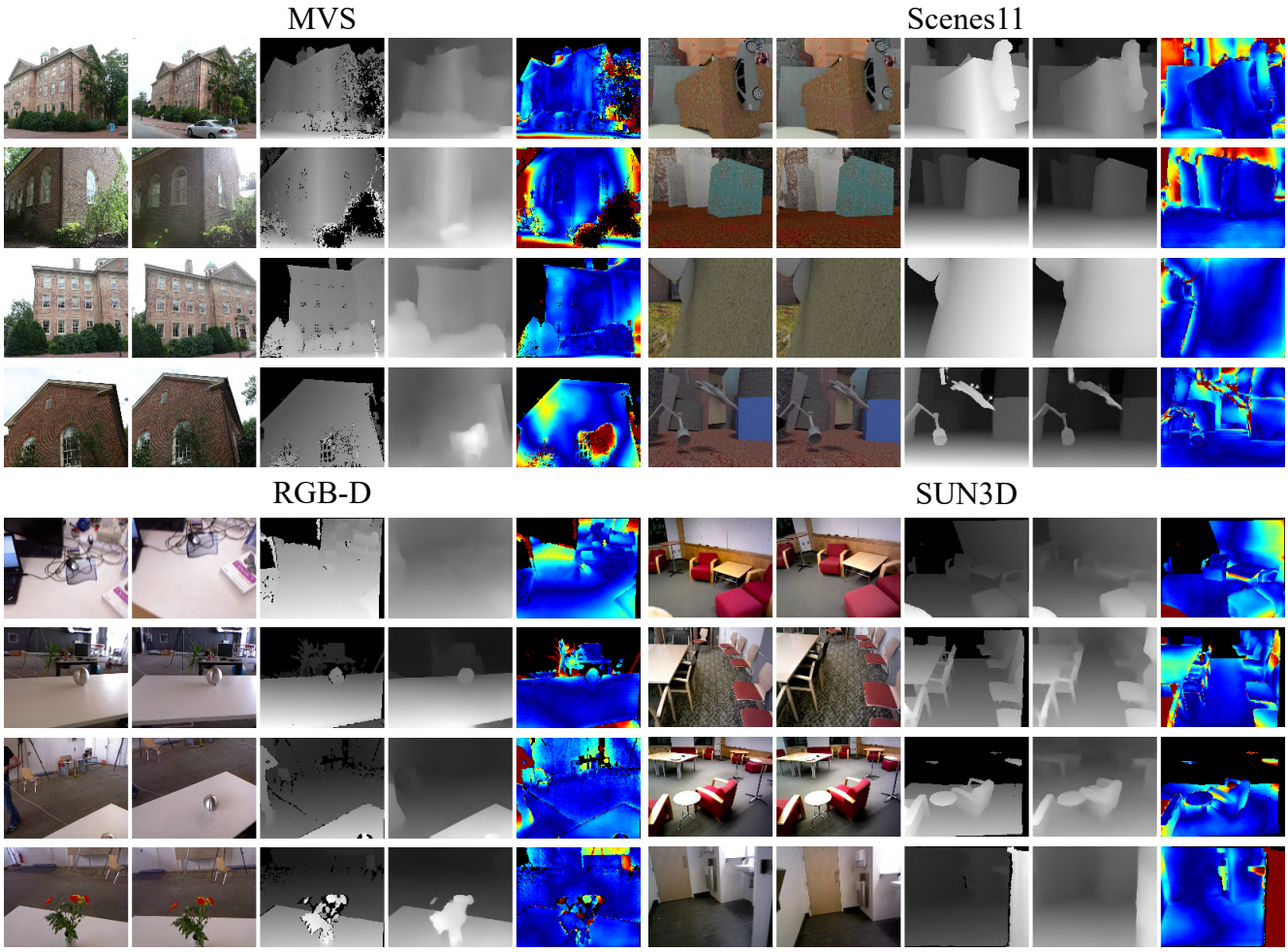


Fig. 2. Depth map estimation of the DeMoN dataset. For each sample, from left to right: source image, target image, ground truth depth map, estimated depth map, and the L1-rel error map.

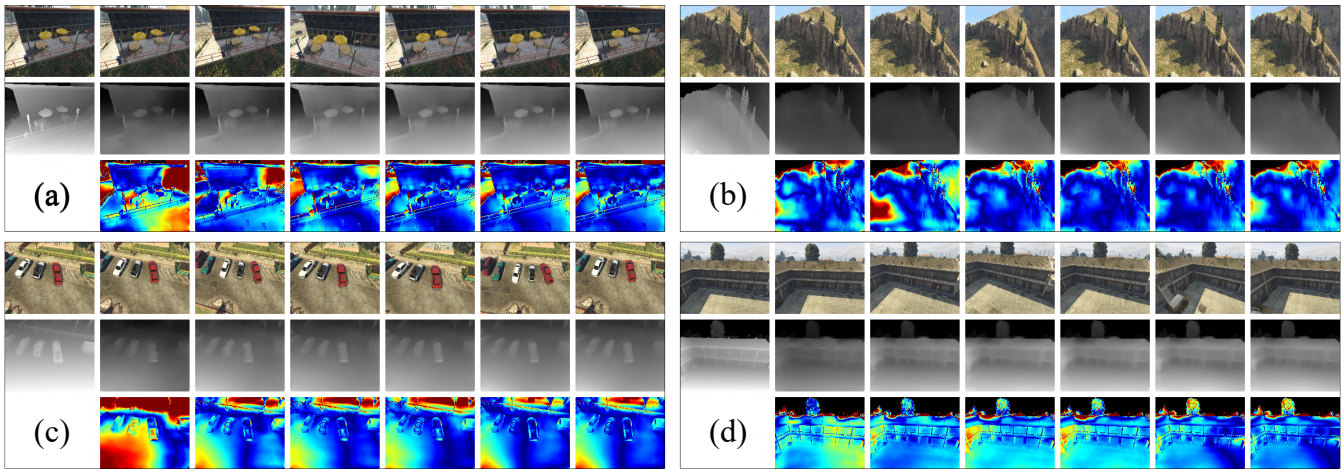
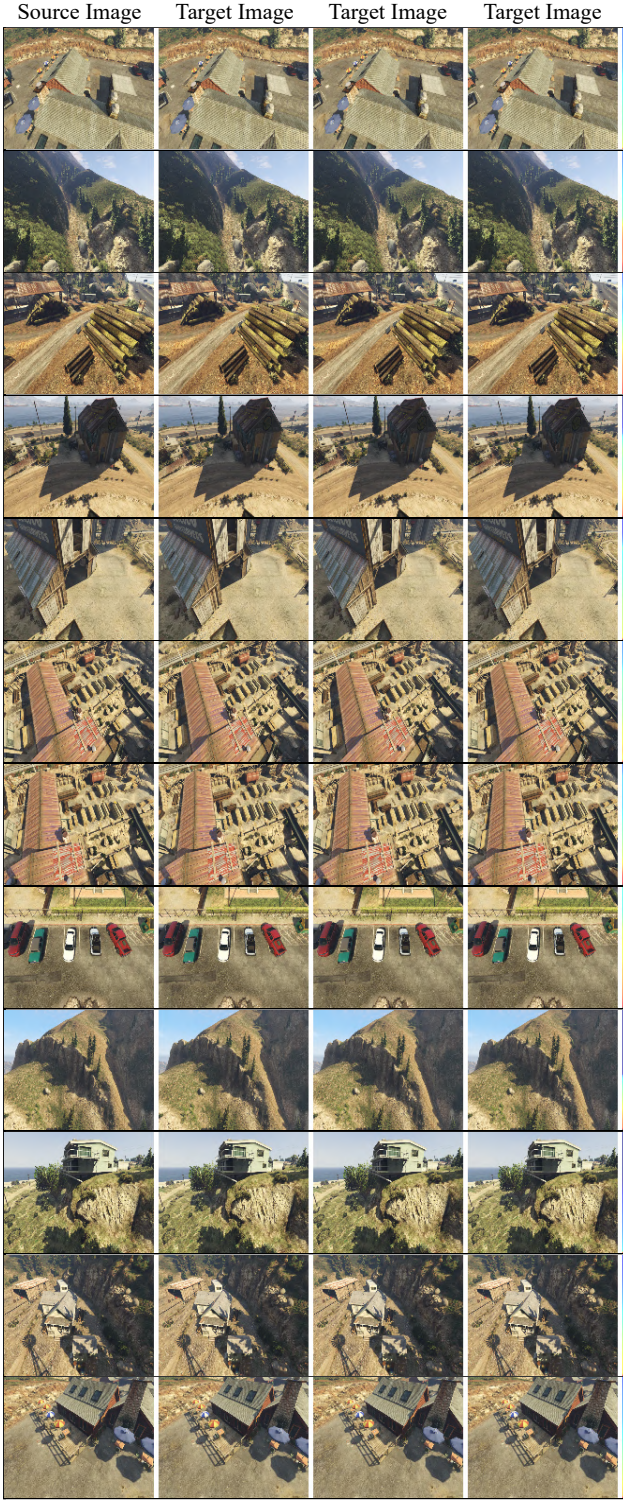


Fig. 3. Depth map quality given different numbers of target images. For each sample, the up row shows the source image I_s, and five target images (from $I_{t,1}$ to $I_{t,5}$). The middle row shows the ground truth depth map and estimated depth maps with different target images. The bottom row shows the L1-rel error maps of the estimation. For example, from left to right, the fourth column shows target image $I_{t,3}$, estimated depth map which uses I_s, $I_{t,1}$, $I_{t,2}$, and $I_{t,3}$ as input, and the corresponding error map.



Metric	Ours Error	DeepMVS Error
L1-inv	0.003	**0.002**
sc-inv	0.075	**0.062**
L1-rel	0.059	**0.030**
L1-inv	**0.003**	**0.003**
sc-inv	**0.235**	0.893
L1-rel	0.148	**0.139**
L1-inv	**0.005**	0.013
sc-inv	**0.182**	0.342
L1-rel	**0.099**	0.263
L1-inv	0.003	**0.002**
sc-inv	**0.240**	0.460
L1-rel	0.106	**0.085**
L1-inv	**0.006**	0.007
sc-inv	**0.138**	0.187
L1-rel	**0.090**	0.108
L1-inv	**0.002**	**0.002**
sc-inv	**0.079**	0.090
L1-rel	0.059	**0.051**
L1-inv	0.003	**0.002**
sc-inv	**0.092**	0.138
L1-rel	0.073	**0.065**
L1-inv	**0.002**	0.003
sc-inv	**0.042**	0.089
L1-rel	**0.027**	0.041
L1-inv	**0.001**	**0.001**
sc-inv	**0.127**	0.664
L1-rel	**0.095**	0.128
L1-inv	**0.003**	0.007
sc-inv	**0.223**	1.321
L1-rel	**0.128**	0.232
L1-inv	0.004	**0.001**
sc-inv	0.096	**0.069**
L1-rel	0.071	**0.032**
L1-inv	**0.005**	0.027
sc-inv	**0.135**	0.569
L1-rel	**0.065**	0.697

Fig. 4. Depth comparison between our method and DeepMVS using *randomly* sampled images. The right side is the calculated depth error. Best results are highlighted in bold. As shown, our method estimates smooth and detailed depth maps, and is much more efficient than DeepMVS.

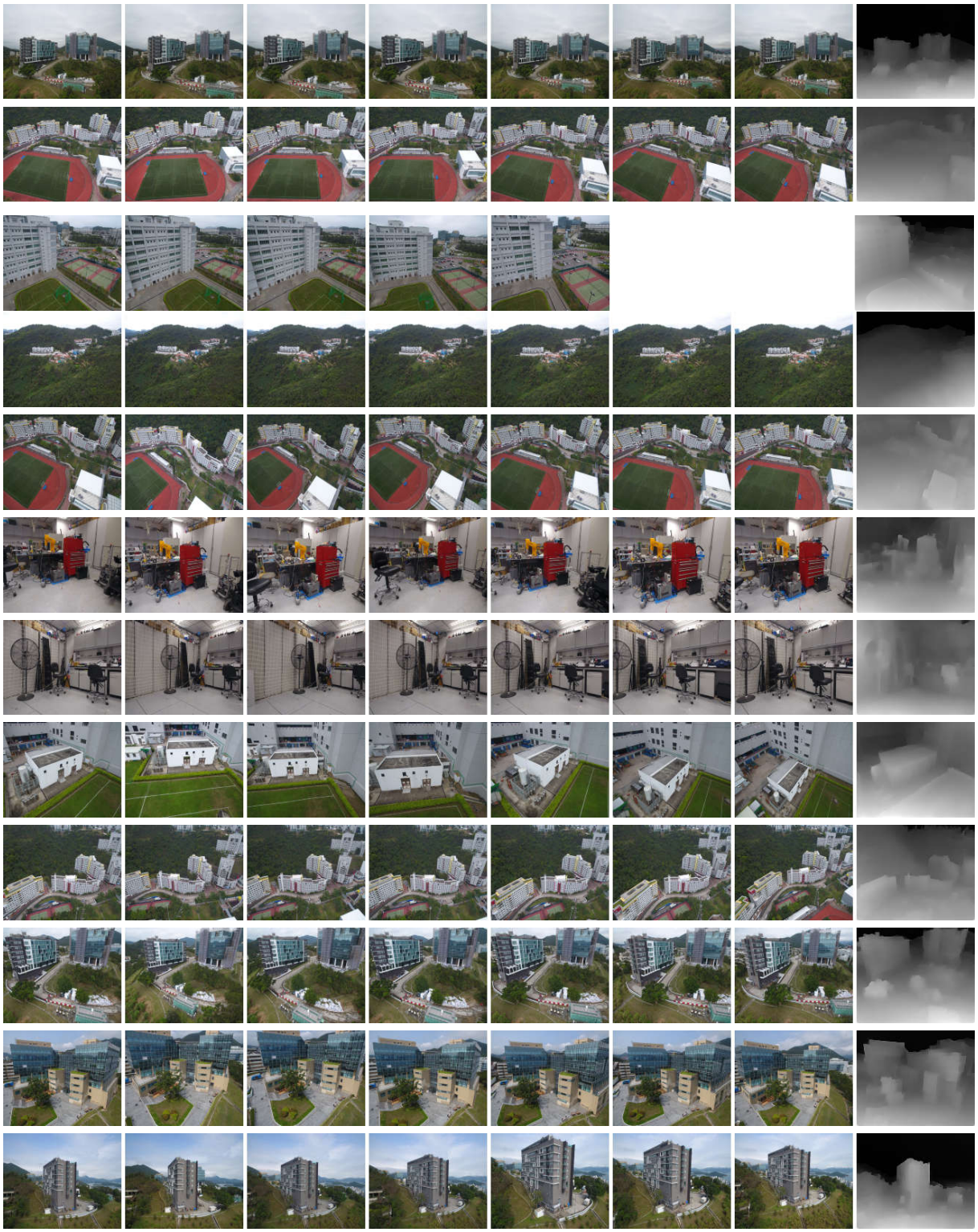


Fig. 5. Applying the GTA-SfM trained models to aerial photographs and indoor images. From left to right in each row is the source image, target images, and the estimated depth map. The source image on the third row is observed by four target images, and other source images are observed by six target images.

Fig. 6. Corresponding point cloud visualization of the estimated depth maps in Figure 5. In each sample, left is the source image and right is the rendered point cloud. Pixels with depth larger than 200 are considered as the sky and not visualized.

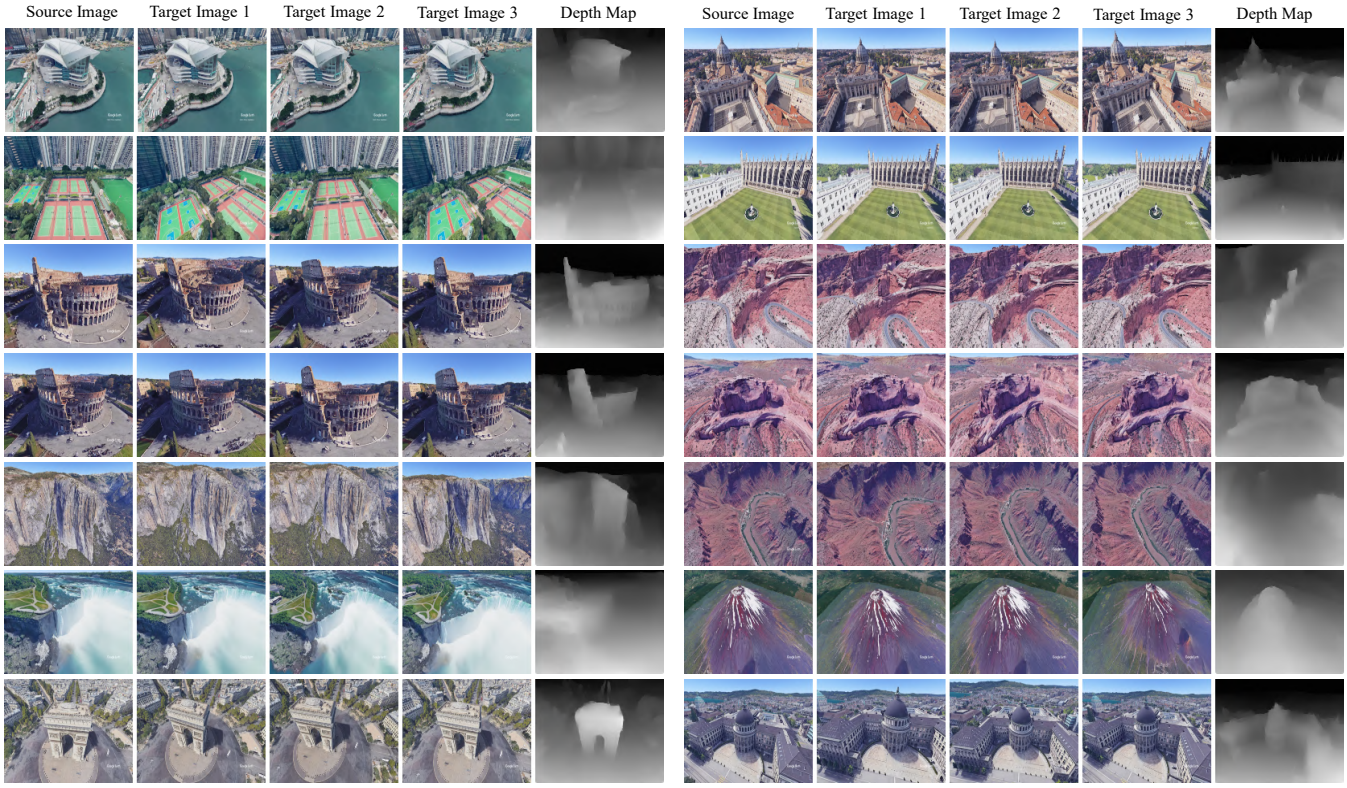


Fig. 7. Applying the GTA-SfM trained models to reconstruct images from Google Earth. Different scenes are used to show the generalization ability of the proposed method.

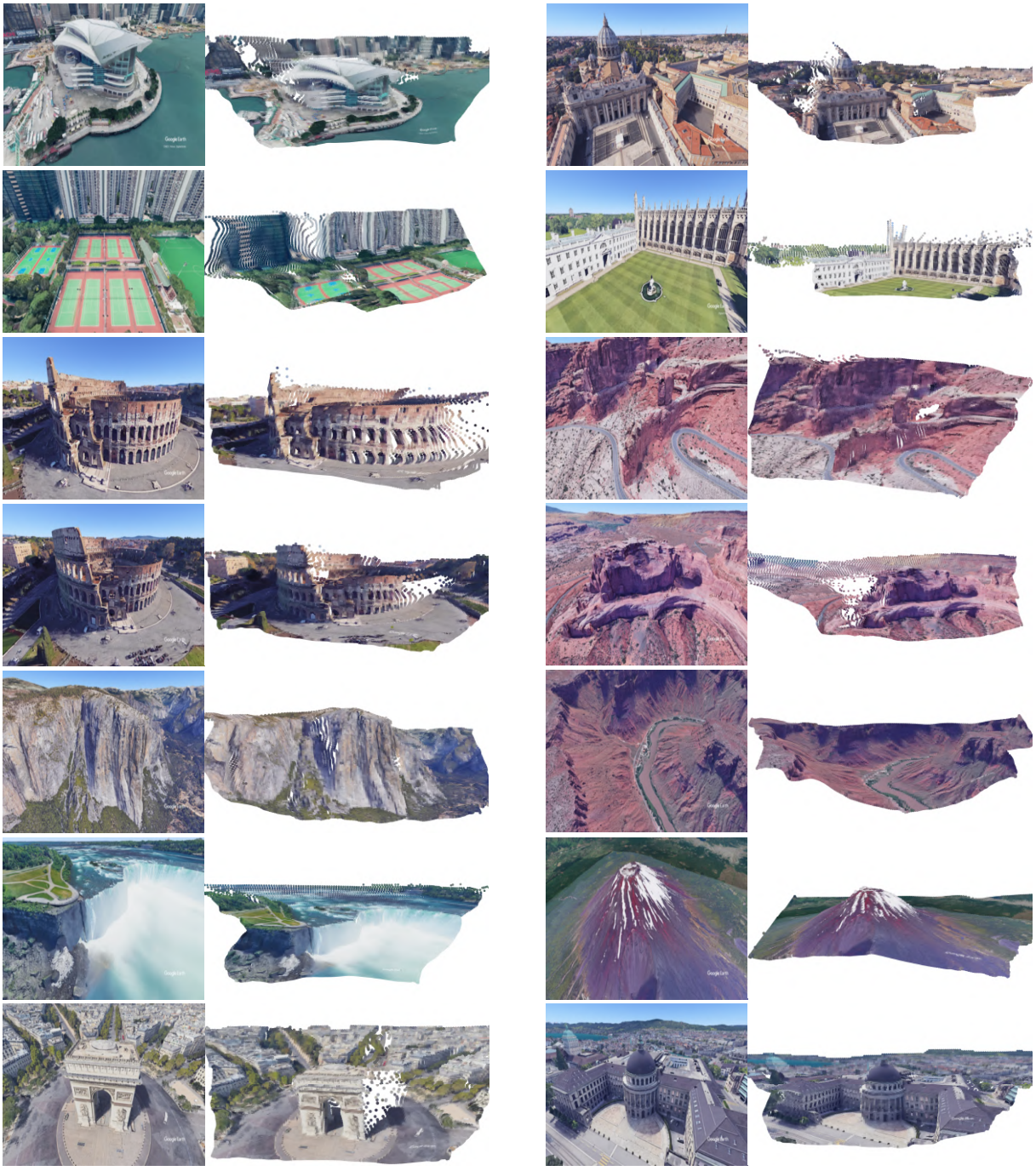


Fig. 8. Point cloud visualization of reconstructed images from Google Earth.

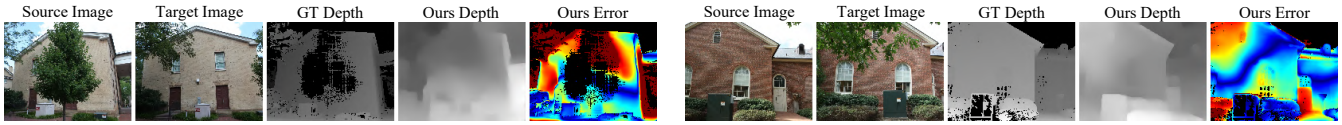


Fig. 9. Occlusion problem in the MVS sequence. Our method utilizes the multiview observations to triangulate the depth of each pixel thus the depth of occluded parts is difficult to be estimated. Even 'ground truth' method cannot deal with such occlusion problems.