Learning Robust Image-Based Rendering on Sparse Scene Geometry via Depth Completion

Yuqi Sun, Shili Zhou, Ri Cheng, Weimin Tan, Bo Yan, Lang Fu
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing,
Shanghai Collaborative Innovation Center of Intelligent Visual Computing, Fudan University, Shanghai, China
{yqsun20, slzhou19, rcheng20, wmtan, byan}@fudan.edu.cn, ful21@m.fudan.edu.cn

1. Video Demo

We provide a video demo here to demonstrate the performance of our method for image-based rendering (IBR) on sparse scene geometry. Three scenes are selected, *Truck*, *Bike*, *Statue* from three datasets, Tanks and Temples, Free View Synthesis and our own Surround dataset respectively. We compare our method with four state-of-the-art (SOTA) IBR methods, FVS [3], SVS [4], EVS [1] and SVNVS [5]. Red boxes in the video indicates failed cases of these methods. We also provide illustration for sparse depth maps and corresponding complete depth maps from geometry recovery (GR) stage in the bottom of our results. It clearly shows the effect of GR stage, which is the main reason why our method is robust to sparse scene geometry.

2. Comparisons and Results

In this section, we compare four SOTA IBR methods with our method and show more quantitative and qualitative results.

2.1. Methods comparisons

Four SOTA IBR methods are compared with our method, FVS [3], SVS [4], EVS [1] and SVNVS [5]. FVS [3] and SVS [4] use a large amount of views to reconstruct a 3D mesh representation and render dense depth maps. They assume dense real views are available and spend lots of time offline to conduct the 3D reconstruction process. Their methods perform well in such a setting but are not robust to sparse input. When the number of available views reduces, the performance of their methods drops off significantly. Our method solves this problem by introducing a learning-based depth completion in our GR stage. With GR stage, our method can preserve a high-fidelity result on sparse scene geometry. We show the quality changes of their methods and our method in Figure 1 at different depth sparsity levels. When the depth map becomes sparse, FVS [3] and SVS [4] produce blur result in the depth missing area, while our method can recover the realistic image content.

EVS [1] and SVNVS [5] take sparse source views to render novel view. EVS [1] apply a deep-learning multi-view stereo (MVS) method named DeepMVS [2] to estimate a depth-probability volume for each view. SVNVS [5] directly implements a MVS module in its network for end-to-end training. Since these two methods both use a layered representation, they require huge memory or computation costs. For example, EVS needs about 4 minutes to render a target view when the number of input views is 5 and the resolution is 250×500. Moreover, they only consider the sparse views setting and are not suitable for dense views. In contrast, our method can support high-resolution input and is more robust and flexible for scene geometry at different sparsity levels.

2.2. Results

Table 1 shows quantitative results on 6 scenes in Free View Synthesis dataset. When K reduces to 4 from 8, the performance of our method changes very little, while FVS [3] and SVS [4] decline rapidly. Our method is also better than EVS [1] and SVNVS [5]. We show some visual comparisons in Figure 2. FVS [3] and SVS [4] tend to blur at the edges of images. EVS [1] will cause image distortions and SVNVS [5] will introduce a lot of noise. Our method can generate most realistic results compared to the ground-truth images.

We supplement EVS [1] results in Table 2. It performs much better than SVNVS [5] in Free View Synthesis and Surround datasets. We believe the reason is that the nearby source views in these two datasets are much closer than in Tanks and Temples dataset, and EVS is more suitable for near view-interpolation.

3. Surround Dataset

Recent IBR applications focus on a circle-around scenario, such as theaters, basketball and soccer stadiums, and they wish to achieve a "bullet time" effect. However, in

Figure 1. **Comparisons on different depth maps.** Three rows shows the performance of FVS [3], SVS [4] and our method when K is 4, 8, and all. When the depth map becomes sparse, FVS [3] and SVS [4] result in blur in the depth missing area, while our method can still have a realistic result. We also show the complete depth map from our GR stage in the last column.

Method	Input	K	Bike		Flowers		Pirate		Playground		Sandbox		Soccertable	
			↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS
FVS [3]	5	4	21.61	0.1234	28.01	0.0644	26.62	0.1192	26.11	0.0874	26.78	0.1245	27.96	0.0501
SVS [4]			23.15	0.1389	27.72	0.0862	25.96	0.1813	28.11	0.1118	26.01	0.1485	27.25	0.0891
Ours			**24.35**	**0.0691**	**31.30**	**0.0399**	**27.60**	**0.1049**	**29.12**	**0.0464**	**30.41**	**0.0813**	**31.46**	**0.0384**
FVS [3]	5	8	22.80	0.0895	27.85	0.0818	26.83	0.1173	26.42	0.0825	27.87	0.1018	28.40	0.0453
SVS [4]			**25.01**	0.0848	28.16	0.1399	27.31	0.1404	28.85	0.0959	27.17	0.1201	28.00	0.0754
Ours			24.64	**0.0660**	**31.25**	**0.0401**	**27.59**	**0.1049**	**29.14**	**0.0465**	**30.72**	**0.0776**	**31.31**	**0.0378**
EVS [1]	5	-	21.28	0.0923	29.79	0.0414	26.31	0.1044	27.28	0.0533	28.26	0.0786	29.48	0.0411
SVNVS [5]	6	-	20.61	0.1507	26.88	0.0592	25.73	0.1267	24.90	0.1182	24.71	0.1565	25.36	0.0925
Ours	5	4	**24.35**	**0.0691**	**31.30**	**0.0399**	**27.60**	**0.1049**	**29.12**	**0.0464**	**30.41**	**0.0813**	**31.46**	**0.0384**

Table 1. **Quantitative comparisons on Free View Synthesis dataset**."Input" and "K" denotes the number of input source views and depth sparsity levels, respectively. We show the best result in bold.

such big venues, it is difficult to provide enough number of cameras. Therefore, it requires a larger-view interpolation method in a surround scene setting. Driven by this demand, we propose a new dataset called Surround for evaluating IBR methods in surround setting. In order to adapt to real demand in practice, we choose sport stadiums and big meeting rooms. Surround dataset contains 6 scenes, *Basketball*, *Meetingroom*, *Park*, *Philosopher*, *Soccer* and *Statue*. We use a handheld camera to capture a 360-degree video around the scene and extract 150 to 300 source images from it. As source images cover a 360-degree panorama, our dataset can be used to evaluate larger-view interpolation by setting different sampling rates. Following FVS [3], we use COLMAP to estimate camera poses, depth maps and 3D point clouds. We show some illustrations in Figure 3.

4. Dataset Preprocessing

To evaluate IBR methods on sparse scene geometry, we preprocess two public datasets, Tanks and Temples, Free View Synthesis, and our own Surround dataset, to generate depth maps at different sparsity levels. We use the number of images input into COLMAP to divide sparsity levels, defined as K. For example, when K=4, we select 4 nearby source views (ID = 0, 1, 2, 3) as a group and send them into COLMAP to generate sparse depth maps. We set K as 4 and 8 in practice and preprocess all three datasets. The dense depth maps obtained from all source views are considered as ground-truth depth maps, with K = all.

For the test dataset of Tanks and Temples, we provide K=16 and more sampling strategy in addition. Specifically, we perform an isometric sampling according to K. For example, when K=4, we select 4 source views (ID = 0, 4, 8, 12). In this manner, we can provide more detailed sparsity division levels. We use valid depth ratio to represent them, which is the percentage of number of pixels with valid depth values to the total pixel number. After all operations, we can obtain 6 depth sparsity levels, 40%, 50%, 60%, 70%, 80%, 90%.

To be noticed, the camera poses and sparse depth maps generated from K images are not accurate enough, thus they can not be used in warping the images to additional views. To handle this problem, we first find a valid mask for the generated sparse depth map with a threshold. And then we use the valid mask to element-wise multiply the ground-truth depth map to get the final sparse depth map for training.

Method	Input	K	Basketball		Meetingroom		Park		Philosopher		Soccer		Statue	
			↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS	↑ PSNR	↓ LPIPS
EVS [1]	5	-	25.64	0.0684	24.49	0.1175	27.57	0.0837	27.43	**0.0912**	24.41	0.1152	27.87	**0.0722**
SVNVS [5]	6	-	24.27	0.0890	24.55	0.1066	24.09	0.1334	24.37	0.1133	23.59	0.1425	24.78	0.1058
Ours	5	4	**28.46**	**0.0588**	**27.63**	**0.0500**	**28.27**	**0.0688**	**28.79**	0.0946	**26.11**	**0.1129**	**29.23**	0.0770

Table 2. **Quantitative comparisons on Surround dataset**. It is cleat that our method performs much better than EVS [1] and SVNVS [5], especially in PSNR.

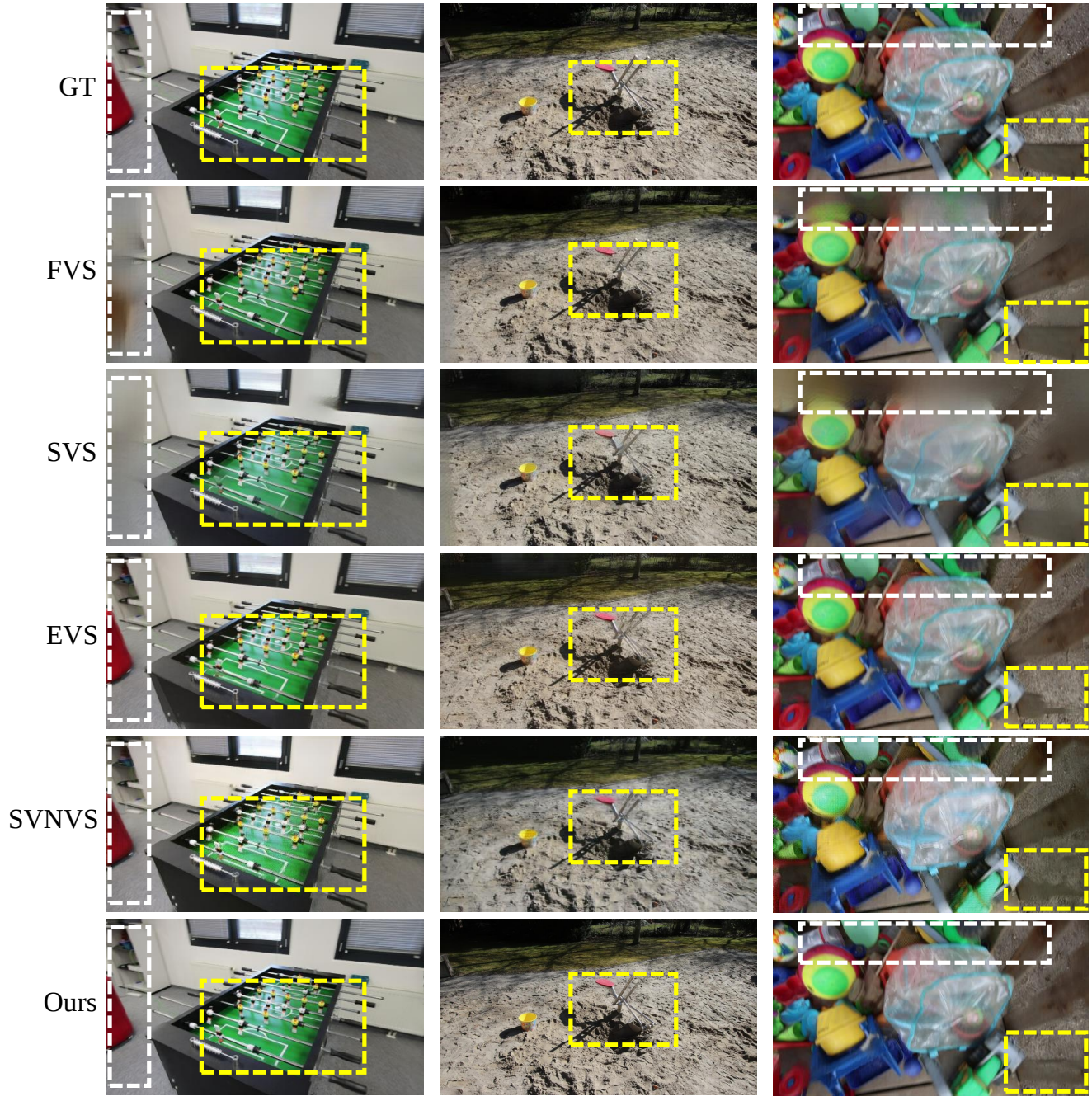

Figure 2. Visual comparisons on Free View Synthesis dataset when K=4.

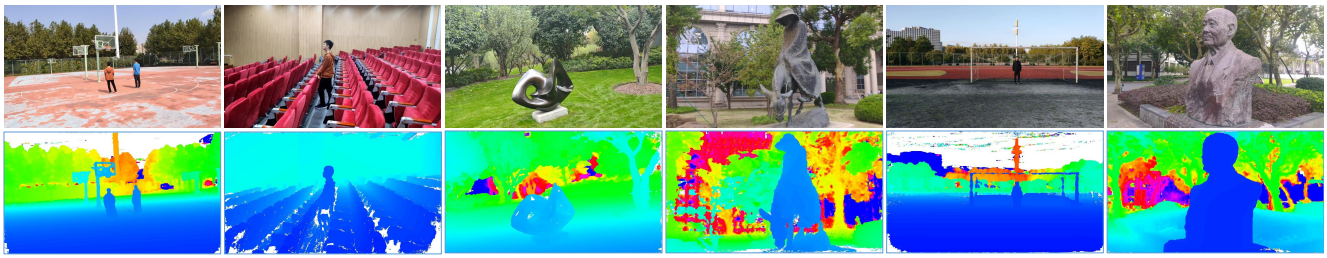

Figure 3. **Illustrations for Surround dataset.** The first row shows the RGB images and the second row shows the corresponding depth map rendered from 3D point cloud.

References

[1] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7781–7790, 2019. 1, 2, 3

[2] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 1

[3] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision*, pages 623–640. Springer, 2020. 1, 2

[4] Gernot Riegler and Vladlen Koltun. Stable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12216–12225, 2021. 1, 2

[5] Yujiao Shi, Hongdong Li, and Xin Yu. Self-supervised visibility learning for novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9675–9684, 2021. 1, 2, 3