Self-Supervised Visibility Learning for Novel View Synthesis

Yujiao Shi [1], Hongdong Li [1], Xin Yu [2]
[1]Australian National University and ACRV [2]University of Technology Sydney
yujiao.shi@anu.edu.au, hongdong.li@anu.edu.au, xin.yu@uts.edu.au

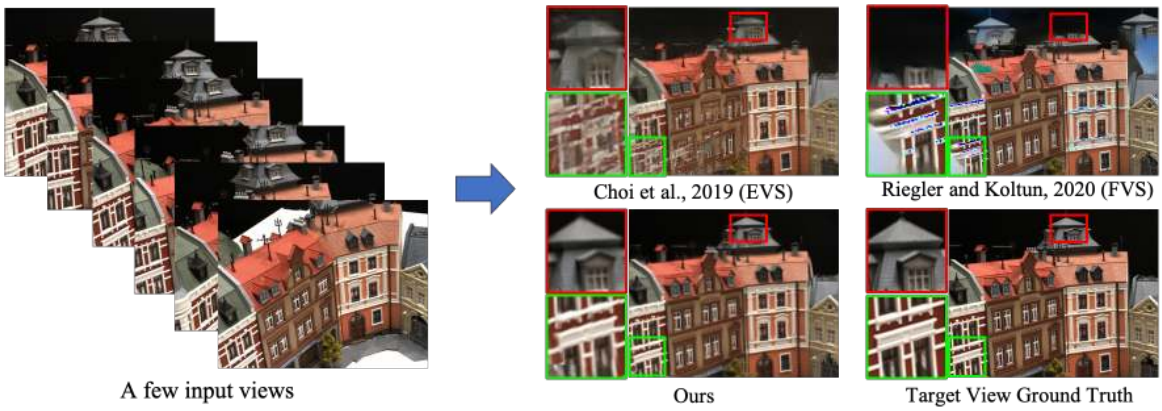


Figure 1: Given a few sparse and unstructured input multi-view images, our goal is to synthesize a novel view from a given target camera pose. Our method estimates target-view depth and source-view visibility in an end-to-end self-supervised manner. Compared with the previous state-of-the-art, such as Choi *et al*. [4] and Riegler and Koltun [24], our method produces superior novel view images of higher quality and with finer details, better conform to the ground-truth.

Abstract

*We address the problem of novel view synthesis (NVS) from a few sparse source view images. Conventional image-based rendering methods estimate scene geometry and synthesize novel views in two separate steps. However, erroneous geometry estimation will decrease NVS performance as view synthesis highly depends on the quality of estimated scene geometry. In this paper, we propose an end-to-end NVS framework to eliminate the error propagation issue. To be specific, we construct a volume under the target view and design a source-view visibility estimation (SVE) module to determine the visibility of the target-view voxels in each source view. Next, we aggregate the visibility of all source views to achieve a consensus volume. Each voxel in the consensus volume indicates a surface existence probability. Then, we present a soft ray-casting (SRC) mechanism to find the most front surface in the target view (*i.e.*, depth). Specifically, our SRC traverses the consensus volume along viewing rays and then estimates a depth probability distribution. We then warp and aggregate source view pixels to synthesize a novel view based on the estimated source-view visibility and target-view depth. At last, our network is trained in an end-to-end self-supervised fashion, thus significantly alleviating error accumulation in view synthesis. Experimental results demonstrate that our method generates novel views in higher quality compared to the state-of-the-art.*

1. Introduction

Suppose after taking a few snapshots of a famous sculpture, we wish to look at the sculpture from some other different viewpoints. This task would require us to generate novel-view images from the captured ones and is generally referred to as "NVS". However, compared with previous solutions, our setting is more challenging, because the number of available real views is very limited, and the underlying 3D geometry is not available. Moreover, the occlusion along target viewing rays and the visibility of target pixels in source views are hard to infer.

Conventional image-based rendering (IBR) methods [4, 24, 10, 42, 23] first reconstruct a proxy geometry by a multi-view stereo (MVS) algorithm [12, 47, 48, 46]. They then aggregate source views to generate the new view according

to the estimated geometry. Since the two steps are separated from each other, their generated image quality is affected by the accuracy of the reconstructed 3D geometry.

However, developing an end-to-end framework that combines geometry estimation and image synthesis is non-trivial. It requires addressing the following challenges. First, estimating target view depth by an MVS method will be no longer suitable for end-to-end training because they need to infer depth maps for all source views. It is time- and memory-consuming. Second, when source view depths are not available, the visibility of target pixels in each source view is hard to infer. A naive aggregation of warped input images would cause severe image ghosting artifacts.

To tackle the above challenges, we propose to estimate target-view depth and source-view visibility directly from source view images, without estimating depths for source views. Specifically, we construct a volume under the target view camera frustum. For each voxel in this volume, when its projected pixel in a source view is similar to the projected pixels in other source views, it is likely that the voxel is visible in this source view. Motivated by this, we design a source-view visibility estimation module (SVE). For each source view, our SVE takes the warped source view features as input, compares their similarity with other source views, and outputs visibility of the voxels in this source view.

Then, we aggregate the estimated visibility of the voxels in all source views, obtaining a consensus volume. The value in each voxel denotes a surface existence probability. Next, we design a soft ray-casting (SRC) mechanism that traverses through the consensus volume along viewing rays and finds the most front surfaces (*i.e.*, depth). Since we do not have ground truth target-view depth as supervision, our SRC outputs a depth probability instead of a depth map to model uncertainty.

Using the estimated target-view depth and source-view visibility, we warp and aggregate source view pixels to generate the novel view. Since the 3D data acquisition is expensive to achieve in practice, we do not have any explicit supervision on the depth or visibility. Their training signals are provided implicitly by the final image synthesis error. We then employ a refinement network to further reduce artifacts and synthesize realistic images. To tolerate the visibility estimation error, we feed our refinement network the aggregated images along with warped source view images.

2. Related Work

Traditional approaches. The study of NVS has a long history in the field of computer vision and graphics [9, 15, 28, 5]. It has important applications in robot navigation, film industry, and augmented/virtual reality [31, 33, 32, 30, 11]. Buehler *et al*. [2] define an unstructured Lumigraph and introduce the desirable properties for image-based rendering. Fitzgibbon *et al*. [7] solve the image-based rendering problem as a color reconstruction without explicit 3D geometry modelling. Penner and Zhang [23] propose a soft 3D reconstruction model that maintains continuity across views and handles depth uncertainty.

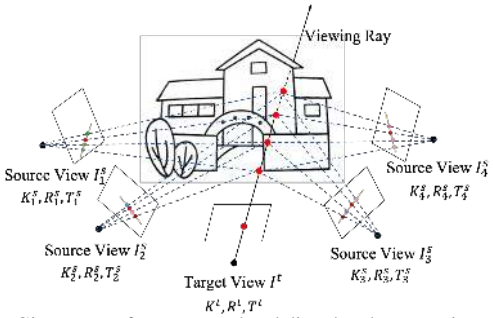


Figure 2: Given a set of unstructured and disordered source views, we aim to synthesize a new view from a position not in the original set. For a 3D point lies in a target viewing ray, when its projected pixels on source view images are consistent with each other, it is of high probability that a surface exists at the corresponding location. The color of this surface can be computed as a visibility-aware combination of source view pixel colors.

Learning-based methods. Recently, learning-based approaches have demonstrated their powerful capability of rendering new views. Several works have been proposed to train a neural network that learns geometry implicitly and then synthesizes new views [53, 39, 20, 22, 6, 51]. Most of those methods can synthesize arbitrarily new views from limited input views. However, their performance is limited due to the lack of built-in knowledge of scene geometry.

Scene representations. Some end-to-end novel view synthesis methods model geometry by introducing specific scene representations, such as multi-plane images (MPI) [52, 18, 38, 43, 8] and layered depth images (LDI) [29, 44, 34, 44]. MPI represents a scene by a set of front-parallel multi-plane images, and then a novel view image is rendered from it. Similarly, LDI depicts a scene in a layered-depth manner.

Deep networks have also been used as implicit functions to represent a specific scene by encapsulating both geometry and appearance from 2D observations [19, 49, 37, 21, 36, 41]. Those neural scene representations are differentiable and theoretically able to remember all the details of a specific scene. Thus, they can be used to render high-quality images. However, since these neural representations are used to depict specific scenes, models trained with them are not suitable to synthesize new views from unseen data.

Image-based rendering. Image-based rendering techniques incorporate geometry knowledge for novel view synthesis. They project input images to a target view by an estimated geometry and blend the re-projected images [4, 24, 10, 42, 23]. Thus, they can synthesize free-viewpoint images and generalize to unseen data. However, as geometry estimation and novel view synthesis are two separate steps, these techniques usually produce artifacts when inaccurate

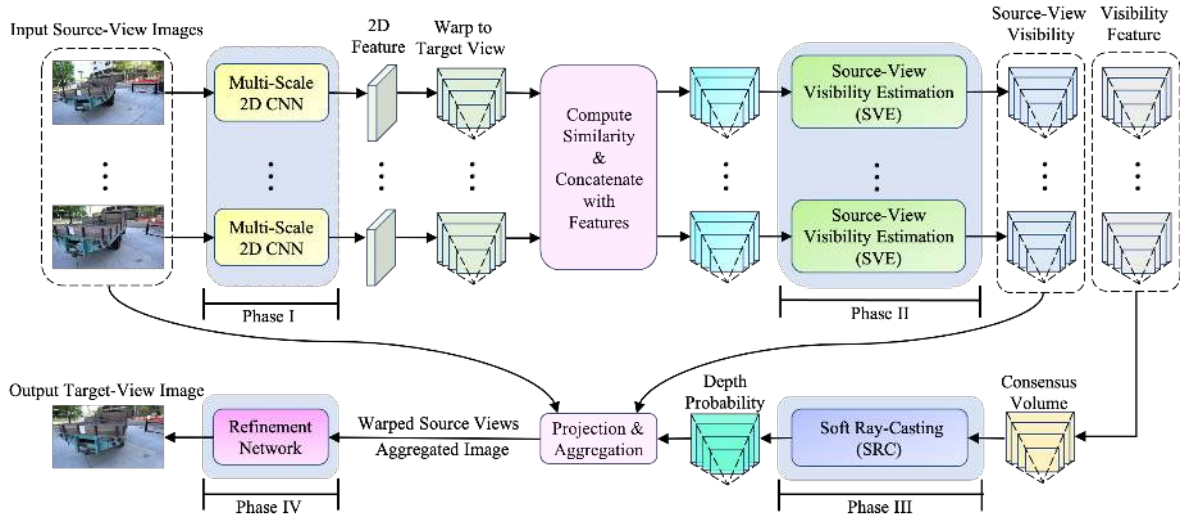


Figure 3: An overall illustration of the proposed framework. We first extract multi-scale features (Phase I) from source images and warp them to the target view. We then design a source-view visibility estimation (SVE) module (Phase II) to estimate the visibility of target voxels in each source view. By aggregating visibility features from all source views, we construct a consensus volume to represent surface existence at different voxels. Next, we design an LSTM-based soft ray-casting (SRC) mechanism (Phase III) to render the depth probability from the consensus volume. By using the estimated source-view visibility and target-view depth, we warp and aggregate source view images. We finally apply a refinement network (Phase IV) to further reduce artifacts in the aggregated images and synthesize realistic novel views.

geometry or occlusions occur. Choi *et al*. [4] estimate a depth map in the target view by warping source view probability distributions computed by DeepMVS [12]. To tolerate inaccurate depths, aggregated images as well as original patches extracted from source view images are fed to their proposed refinement network. Riegler and Koltun [24] leverage COLMAP [26, 27] to reconstruct 3D meshes from a whole sequence of input images and obtain target view depths using the estimated geometry.

3. Problem Statement

Our goal is to synthesize a novel view I^t, given target camera parameters K^t, R^t, T^t, from a set of input images, I_i^s, $i = 1, 2, ..., N$. We assume there is sufficient overlap between the source views such that correspondences can be established. We estimate source view camera intrinsic and extrinsic by a well-established structure-from-motion (SfM) pipeline, *e.g.*COLMAP [26]. Fig. 2 illustrates the situation. Mathematically, we formulate this problem as:

$$I^{t*} = \underset{I^t}{\operatorname{argmax}}\ p\left(I^t | I_1^s, I_2^s, ..., I_N^s\right), \tag{1}$$

where $p(\cdot)$ is a probability function.

Due to the expensive accessibility of 3D data (*e.g.*, depths) and a limited number of input views, it is hard to compute accurate 3D geometry from input source views. Therefore, our intuition is to develop an end-to-end framework that combines geometry estimation and image synthesis, to eliminate the error propagation issue. We achieve this goal by estimating target-view depth and source-view visibility for target pixels directly under the target view.

We assume a uniform prior on the target view depth, and reformulate Eq. (1) as a probability conditioned on depth d:

$$\begin{aligned} I^{t*} &= \underset{I^t}{\operatorname{argmax}} \sum_{d=d_{\min}}^{d_{\max}} p\left(I^t | d\right) p\left(d\right) \\ &= \sum_{d=d_{\min}}^{d_{\max}} \left[\underset{I^t}{\operatorname{argmax}}\ p\left(I^t | d\right)\right] p(d), \end{aligned} \tag{2}$$

where $d_{\min}$ and $d_{\max}$ are statistical minimum and maximum depths of a target view respectively. As the source view images are given, we omit them in this equation.

Following conventional methods, we compute the target view color I^t with the highest probability given depth d as a visibility-aware combination of source view colors:

$$\underset{I^t}{\operatorname{argmax}}\ p\left(I^t | d\right) = \sum_{i=1}^{N} \mathbf{w}_i^d \mathbf{C}_i^d, \tag{3}$$

where $\mathbf{C}_i^d \in \mathbb{R}^{H \times W \times 3}$ is a collection of re-projected target pixels in source view i by inverse warping [16], $\mathbf{w}_i^d \in \mathbb{R}^{H \times W}$ is the blending weight of source view i, and it is computed from the visibility of target pixels in each source view:

$$\mathbf{w}_i^d = exp\left(\mathbf{V}_i^d\right) / \sum_{i=1}^{N} exp\left(\mathbf{V}_i^d\right), \tag{4}$$

where $\mathbf{V}_i^d \in \mathbb{R}^{H \times W}$ is the visibility of target pixels in source view i given the target-view depth d.

In the next section, we will provide technical details on how to estimate the source-view visibility $\mathbf{V}_i^d$ and target-view depth probability distribution $p(d)$.

4. The Proposed Framework

We aim to construct an end-to-end framework for novel view synthesis from a few sparse input images. By doing so, inaccurately-estimated geometry can be corrected by image synthesis error during training. We achieve this goal by estimating target-view depth and source-view visibility directly under the target view. Fig. 3 depicts the proposed pipeline.

Start from a blank volume in the target-view camera frustum. Our goal is to select pixels from source-view images to fill in the voxels of this volume. After that, we can render the target view image from this colored volume. In this process, the visibility of the voxels in each source view, and the target-view depth, are two of the most crucial issues.

4.1. A multi-scale 2D CNN to extract features

When a voxel of this volume is visible in a source view, its projected pixel in this source view should be similar to the projected pixels in other source views. This is the underlying idea for the source-view visibility estimation. However, the pixel-wise similarity measure is not suitable for textureless and reflective regions. Hence, we propose to extract high-level features from source view images for the visibility estimation by a 2D CNN.

Our 2D CNN includes a set of dilation convolutional layers with dilation rates as 1, 2, 3, 4 respectively. Its output is a concatenation of extracted multi-scale features. This design is to increase the receptive field of extracted features and retain low-level detailed information [45], thus increasing discriminativeness for source view pixels.

Denote $\mathbf{F}_i \in \mathbb{R}^{H\times W\times D\times C}$ as the warped features of source view i, where D is the sampled depth plane number in the target view. For target-view voxels at depth d, we compute the similarity between corresponding features in source view i and other source views as:

$$\mathbf{S}_i^d = \sum_{j=1,j\neq i}^{N} Sim\left(\mathbf{F}_i^d, \mathbf{F}_j^d\right)/(N-1), \tag{5}$$

where $Sim(\cdot,\cdot)$ is a similarity measure between its two inputs, and we adopt cross-correlation in this paper.

4.2. Source-view visibility estimation (SVE) module

In theory, deep networks are able to learn viewpoint invariant features. However, we do not have any explicit supervision on the warped source-view features. The computation of source-view visibility for the voxels is too complex to be modeled by Eq. (5). Hence, we propose to learn a source-view visibility estimation (SVE) module to predict the visibility of the voxels in each source view.

Our SVE module is designed as an encoder-decoder architecture with an LSTM layer at each stage. The LSTM layer is adopted to encode sequential information along the depth dimension. Our SVE module takes into account self-information ($\mathbf{F}_i^d$), local information ($\mathbf{S}_i^d$) and global information ($\sum_{i=1}^{N}\mathbf{S}_i^d/N$) to determine the visibility of the voxels in each source view. Mathematically, we express it as:

$$\mathbf{V}_i^d, \mathbf{B}_i^d, \text{state}_f^d = f\left(\left[\mathbf{F}_i^d, \mathbf{S}_i^d, \sum_{i=1}^{N}\mathbf{S}_i^d/N\right], \text{state}_f^{d-1}\right), \tag{6}$$

where $[\cdot]$ is a concatenation operation, $\mathbf{V}_i^d \in \mathbb{R}^{H\times W}$ is the estimated visibility for target-view voxels at depth d in source view i, $\mathbf{B}_i^d \in \mathbb{R}^{H\times W\times 8}$ is the associated visibility feature, $f(\cdot)$ denotes the proposed SVE module, state_f^{d-1} is the past memory of our SVE module before depth d and state_f^d is the updated memory at depth d.

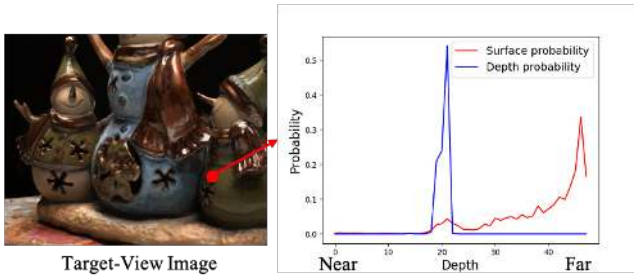


Figure 4: Illustration of our soft ray-casting mechanism. Our SRC traverses through the surface probability curve along target viewing rays from near to far, increases the depth probability of the first opaque element, and decreases depth probabilities of later elements no matter they are opaque or not.

4.3. Soft ray-casting (SRC) mechanism

By aggregating visibility from all source views, we obtain a surface existence probability for each voxel in the target view. As shown in Fig. 4, the surface probability curve along a target viewing ray might be multi-modal, where a smaller peak indicates that a surface is visible by fewer source views and a larger peak suggests that its corresponding surface is visible by a large number of source views.

To increase the representative ability, we aggregate the visibility features, instead of visibility, of source views to compute a consensus volume:

$$\mathcal{C} = \sum_{i=1}^{N}\mathbf{B}_i/N, \tag{7}$$

where $\mathcal{C} \in \mathbb{R}^{H\times W\times D\times 8}$ is the obtained consensus volume.

Then, we design a soft ray-casting (SRC) mechanism to render the target view depth from the consensus volume. Our SRC is implemented in the form of an LSTM layer. Similar to our SVE module, the LSTM layer is to encode sequential relationship along the depth dimension.

The LSTM layer traverses though the consensus volume along target viewing rays from near to far. When meeting the most front surface, it outputs a large depth probability

value for the corresponding voxel. For the later voxels, the LSTM layer sets their probability values to zero. Denote the LSTM cell as $r(\cdot)$. At each depth d, it takes as input the current consensus feature $\mathcal{C}^d$ and its past memory state$_r^{d-1}$, and outputs the depth probability $p(d)$ along with the updated memory state$_r^d$:

$$\text{state}_r^d, p(d) = r(\mathcal{C}^d, \text{state}_r^{d-1}). \tag{8}$$

4.4. Refinement network

Using the estimated source-view visibility and target-view depth probability, we aggregate the source images and obtain I^{t*} by Eq. (2). We then employ a refinement network to further reduce artifacts on the aggregated image.

Our refinement network is designed in an encoder-decoder architecture with convolutional layers. To tolerate errors caused by the visibility estimation block, the encoder in our refinement network is in two branches: one for the aggregated image I^{t*} and another for a warped source view $I_i^{\text{warp}} = \sum_{d=1}^{D} \mathbf{C}_i^d p(d)$. Its outputs are a synthesized target view image $\hat{I}_i^t$ along with a confidence map m_i:

$$\hat{I}_i^t, m_i = \text{Refinement}(I^{t*}, I_i^{\text{warp}}). \tag{9}$$

The final output of our refinement network is computed as:

$$\widetilde{I}^t = \sum_{i=1}^{N} m_i \hat{I}_i^t. \tag{10}$$

4.5. Training objective

We employ the GAN training scheme to train our framework. For brevity, we omit the adversarial loss in this paper. Interested readers are referred to Isola *et al*. [13]. For the target image supervision, we adopt the perceptual loss of Chen and Koltun [3]:

$$\mathcal{L}_{per} = \left\| \widetilde{I}^t - I^t \right\|_1 + \sum_l \lambda_l \left\| \phi_l(\widetilde{I}^t) - \phi_l(I^t) \right\|_1, \tag{11}$$

where $\phi(\cdot)$ indicates the outputs of a set of layers from a pretrained VGG-19 [35], and $\|\cdot\|_1$ is the L_1 distance. The settings for coefficients λ_l are the same as Zhou *et al*. [52].

Self-supervised training signal for our SRC and SVE. Generally, it is difficult for our SRC to decide which is the most front surface in a viewing ray, especially when the surface probability curve is multi-modal. We expect this soft determination can be learned statistically from training. Particularly, when the estimated depth is incorrect, the color of warped pixels from source-view images will deviate from the ground truth target view color. This signal would punish the LSTM and helps it to make the right decisions. The same self-supervised training scheme is applied to our SVE module. We illustrate the estimated depth for a target pixel in Fig. 4, and an example of the visibility-aware aggregated image in Fig. 7.

5. Experiments

Dataset and evaluation metric. We conduct experiments on two datasets, Tanks and Temples [14], and DTU [1]. Camera movements in the two datasets include both rotations and translations.

On the Tanks and Temples, we use the training and testing split provided by Riegler and Koltun [24]. In this dataset, 17 out of 21 scenes are selected out as the training set. The remaining four scenes, *Truck*, *Train*, *M60*, and *Playground*, are employed as the testing set. We apply the leave-one-out strategy for training, namely, designating one of the images as target image and selecting its nearby N images as input source images. For testing, different from Riegler and Koltun [24] which uses whole sequences as input, we select N nearby input images for each target view.

For the DTU dataset, it is employed to further demonstrate the generalization ability of trained models. We do not train on this dataset and use the validation set provided by Yao *et al*. [47]. The validation set includes 18 scenes. Each of the scenes contains 49 images. We apply the same leave-one-out strategy as on the Tanks and Temples dataset.

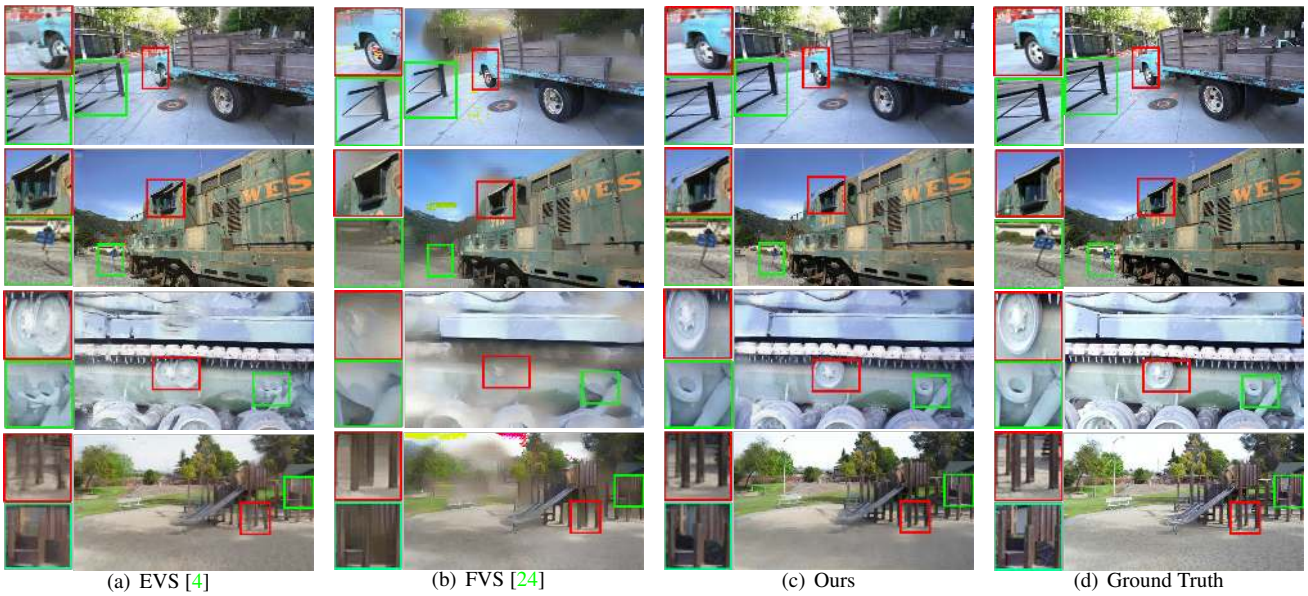
Following recent NVS works [4, 24, 17, 8], we adopt the commonly used SSIM, PSNR and LPIPS [50] for quality evaluation on synthesized images.

Implementation details. For the Tanks and Temples, we experiment on image resolution of 256×448. For the DTU dataset, the input image resolution is 256×320. We use a TITAN V with 12GB memory to train and evaluate our models. We train 10 epochs on the Tanks and Temples dataset with a batch size of 1. It takes 20 hours for training using 6 input images, and $0.35s$ per image (average) for evaluation. We apply inverse depth sampling strategy with depth plane number $D = 48$. For outdoor scenes, *i.e*., the Tanks and Temples, we set $d_{\text{min}} = 0.5$m and $d_{\text{max}} = 100$m. For constrained scenes, *i.e*., the DTU dataset, we employ the minimum (425mm) and maximum (937mm) depth in the whole dataset. The source code of this paper is available at https://github.com/shiyujiao/SVNVS.git.

5.1. Comparison with the state-of-the-art

We first compare with two recent and representative IBR methods, Extreme View Synthesis (EVS) [4] and Free View Synthesis (FVS) [24], with six views as input. We present the quantitative evaluation in the first three rows of Tab. 1. Qualitative comparisons on the Tanks and Temples are presented in Fig. 5.

Both EVS and FVS first estimate depth maps for source views. In their methods, the visibility of target pixels in source views is computed by a photo-consistency check between source and target view depths. EVS aggregates source views simply based on the source-target camera distance. Their aggregation weights do not have the ability to



(a) EVS [4] (b) FVS [24] (c) Ours (d) Ground Truth

Figure 5: Qualitative visualization of generated results on the Tanks and Temples dataset with six views as input. The four examples are from scene *Truck*, *Train*, *M60*, *Playground* respectively.

Table 1: Quantitative comparison with the state-of-the-art. Here, "Whole" denotes using the whole sequence as input; "*" indicates that results are from Zhang *et al*. [49]; and "† " represents that results are from Riegler and Koltun [24].

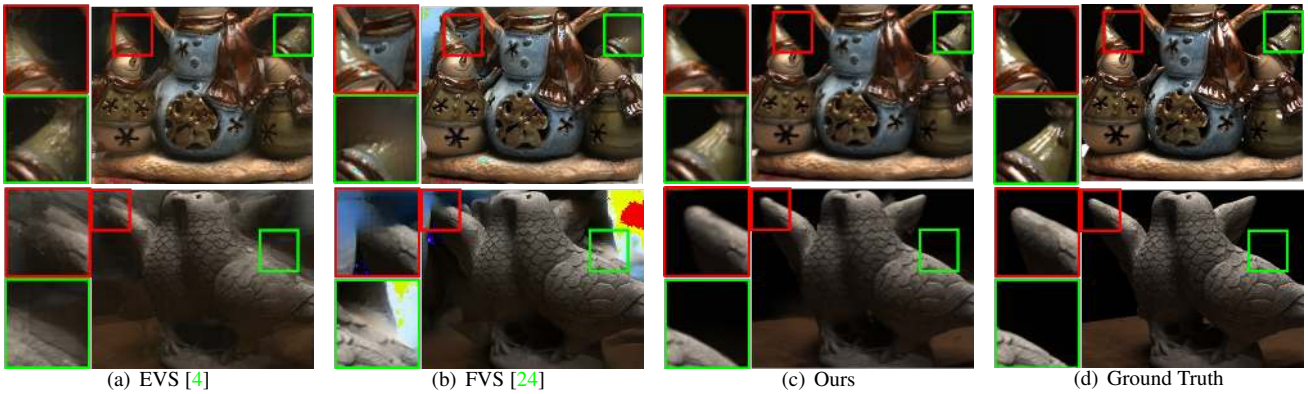
	Input View Number	Tanks and Temples												DTU		
		Truck			Train			M60			Playground					
		LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
EVS[4]	6	0.301	0.588	17.74	0.434	0.434	15.38	0.314	0.585	16.40	**0.243**	0.702	21.57	0.32	0.645	17.83
FVS[24]	6	0.318	0.638	15.82	0.447	0.502	13.71	0.486	0.548	11.49	0.417	0.586	14.95	0.47	0.530	10.45
Ours	6	**0.233**	**0.708**	**21.33**	**0.386**	**0.542**	**18.81**	**0.250**	**0.732**	**19.20**	0.245	**0.710**	**22.12**	**0.303**	**0.721**	**19.20**
Ours	2	0.294	0.627	19.20	0.475	0.464	17.44	0.303	0.667	18.14	0.350	0.604	19.98	0.362	0.625	17.54
	4	0.254	0.682	20.41	0.430	0.495	17.82	0.283	0.698	19.37	0.275	0.663	21.31	0.359	0.646	17.84
	6	**0.233**	**0.708**	**21.33**	**0.386**	**0.542**	**18.81**	**0.250**	**0.732**	**19.20**	**0.245**	**0.710**	**22.12**	**0.303**	**0.721**	**19.20**
NeRF[19]	Whole	0.513*	0.747*	20.85*	0.651*	0.635*	16.64*	0.602*	0.702*	16.86*	0.529*	0.765*	21.55*	–	–	–
FVS[24]	Whole	0.11†	0.867†	22.62†	0.22†	0.758†	17.90†	0.29†	0.785†	17.14†	0.16†	0.837†	22.03†	–	–	–

tolerate visibility error caused by in-accurate depth. Thus, the synthesized images by EVS suffer severe ghosting artifacts, as shown in Fig. 5(a). FVS employs a COLMAP to reconstruct the 3D mesh. When input images densely cover a scene, the reconstructed geometry is exceptionally good, and the synthesized images are of high-quality, as shown in the last row of Tab. 1. However, when the number of input images are reduced, *i.e*., 6, the reconstructed mesh by COLMAP is of poor quality, and the depth-incorrect regions in the synthesized images are blurred, as indicated in 5(b). In contrast, our method does not rely on the accuracy of estimated source-view depths or reconstructed 3D mesh. Instead, we directly recover target-view depth and source-view visibility from input images. Thus, our synthesized images show higher quality than the recent state-of-the-art.

Generalization ability. To further demonstrate the generalization ability, we employ the trained models of the three algorithms to test on the DTU dataset. Quantitative results are presented in the last column of Tab. 1. Our method consistently outperforms the recent state-of-the-art algorithms. We present two visualization examples in Fig. 6. More qualitative results are provided in the supplementary material.

Different input view number. We further conduct experiments on reducing the number of input views of our method. Quantitative results are presented in the bottom part of Tab. 1. Increasing the input view number improves the quality of synthesized images. This conforms to our general intuition that image correspondences can be easily established and more disoccluded regions can be observed when more input views are available.

Comparison with NeRF. For completeness, we present the performance of NeRF [19] with the whole sequence as



(a) EVS [4] (b) FVS [24] (c) Ours (d) Ground Truth

Figure 6: Qualitative visualization of generated results on the DTU dataset with six views as input. The two examples are from scene *scan3* and *scan106*, respectively.

input in the penultimate row of Tab. 1. The major difference between NeRF and our method is the different problem settings. NeRF is more suitable to view synthesis on a specific scene with many images as input. When the scene is changed, NeRF needs to be re-trained on the new scene. In contrast, we expect our network to learn common knowledge from its past observations (training data) and be able to apply the learned knowledge to unseen scenes without further fine-tuning. Thus, our approach is a better choice when the trained model is expected to generalize, and the number of input images is small.

Comparison with Szeliski and Golland [40]. We found our work shares the same spirit with a classical work [40]. Both works construct a virtual camera frustum under the target view and aim to estimate the color and density (depth probability in our work) for each of its elements. Szeliski and Golland [40] first compute an initial estimation by finding agreement among source views. Next, they project the estimation to the source views, compute the visibility of source views, and then refine the estimation iteratively. Benefited from learning-based techniques, our approach encodes the visibility estimation as a single forward step (compared to iterative refinement in Szeliski and Golland [40]) and can handle more complex scenarios, such as textures and reflective regions, as shown in the qualitative visualizations of this paper and supplementary material.

5.2. Ablation study

In this section, we conduct experiments to verify the effectiveness of each component in the proposed framework.

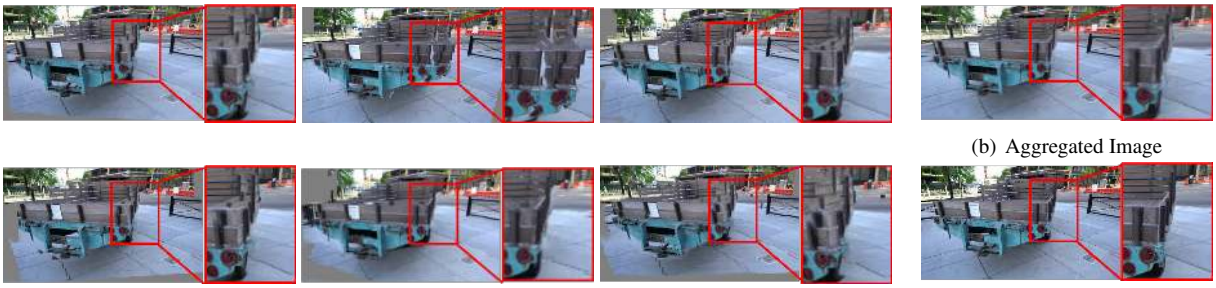
Source-view visibility estimation. We first remove the visibility-aware source view aggregation (indicated by Eq. (2)) from our framework, denoted as "Ours w/o visibility". Instead, we feed the warped source images to our refinement network directly and equally. We expect the refinement network to learn the visibility-aware blending weights for source view images automatically. The results are presented in the first row of Tab. 2. It can be seen that the performance drops significantly compared to our whole pipeline. This indicates that it is hard for the refinement network to select visible pixels from source views.

We present a visualization example of our visibility-aware aggregated result in Fig. 7. As shown in Fig. 7(a), directly warped source images contain severe ghosting artifacts due to occlusions, *i.e.*, disoccluded regions in target view are filled by replicas of visible pixels from a source view. By using the proposed SVE module to estimate the visibility of source views, our aggregated result, Fig. 7(b), successfully reduces the ghosting artifacts and is much more similar to the ground truth image, Fig. 7(c).

Soft ray-casting. We first remove the soft ray-casting mechanism from our whole pipeline, expressed as "Ours w/o ray-casting". Instead, we use the surface probability, *i.e.*, the red curve in Fig. 4, as the depth probability to warp and aggregate source views. As indicated by the second row of Tab. 2, the results are significantly inferior to our whole pipeline. Furthermore, we replace the SRC as the conventional over alpha compositing scheme, denoted as "Ours w over compositing". The results are presented in the third row of Tab. 2. It can be seen that our SRC is necessary and cannot be replaced by the over alpha compositing scheme.

Both SRC and over alpha compositing are neural renderers in NVS. Over-compositing uses opacity to handle occlusions, while our method does not regress opacity for voxels. Our input curve to SRC is obtained by majority voting from source views. A smaller peak in the curve indicates that a surface is visible by fewer source views and a larger peak suggests that a surface is visible by more source views. Due to the fixed weight embedding in over-compositing, the smaller peak at a nearer distance will be ignored while the larger peak will be highlighted. By using LSTM, our SRC can be trained to decide which peak is the top-most surface.

Refinement network. We further ablate the refinement network. In doing so, we remove the warped source



(a) Warped Source-View Images in Target View (b) Aggregated Image (c) Ground Truth

Figure 7: Qualitative illustration on our visibility-aware aggregation. (a) Warped source images in target view. There are severe ghosting artifacts due to occlusions. (b) Aggregated image by using our visibility-aware blending weights. (c) Target view ground truth image.

Table 2: Necessity of each module in the proposed framework.

	Tanks and Temples												DTU		
	Truck			Train			M60			Playground					
	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑
Ours w/o visibility	0.271	0.660	20.35	0.437	0.482	17.93	0.322	0.656	16.83	0.277	0.665	21.40	0.367	0.641	17.03
Ours w/o ray-casting	0.761	0.643	20.04	0.665	0.487	17.83	0.766	0.654	17.45	0.737	0.649	20.98	0.397	0.649	18.71
Ours w over compositing	0.308	0.695	21.19	0.445	0.531	18.14	0.321	0.709	19.32	0.334	0.689	21.57	0.409	0.684	19.02
Ours w/o warped sources	0.250	0.697	20.89	0.402	0.534	18.58	0.263	0.729	19.31	0.254	0.704	22.51	0.322	0.705	19.65
Ours w/o refinement	0.237	0.675	20.99	0.380	0.533	18.28	**0.220**	0.717	**20.13**	**0.244**	0.689	**22.88**	**0.239**	**0.765**	**21.44**
Our whole pipeline	**0.233**	**0.708**	**21.33**	**0.386**	**0.542**	**18.81**	0.250	**0.732**	19.20	0.245	**0.710**	22.12	0.303	0.721	19.20

view images from the refinement network input, denoted as "Ours w/o warped sources". As shown by the results in Tab. 2, there is only a small performance drop compared to our whole pipeline. This indicates that our visibility-aware aggregated images are already powerful enough to guide the refinement network to synthesize realistic images.

We further remove the refinement network from our whole pipeline, denoted as "Ours w/o refinement". The results are presented in the penultimate row in Tab. 2. For the test scenes on the Tanks and Temples dataset, the performance drops compared to our whole pipeline. While for the results on the DTU dataset, "Ours w/o refinement" achieves significantly better performance. This is due to the huge color differences between the training (outdoor) and testing(indoor) scenes. In practice, we suggest the users first visually measure the color differences between the training and testing scenes and then choose a suitable part of our approach. We found that a concurrent work [25] provides a solution to this problem. Interested readers are referred to this work for detailed illustration.

Limitations. The major limitation of our approach is the GPU memory. The required GPU memory increases with the depth plane sampling number and the input view number. By using a 12GB memory GPU, our approach can handle a maximum of 6 input views and 48 depth plane numbers. The advantage of inverse depth plane sampling is that it can recover near objects precisely. The downside is that it handles worse for scene objects with fine structures at distance, because the depth planes at distance is sampled sparsely and the correct depth of some image pixels cannot be accurately searched. Another limitation of our method is that we have not incorporated temporal consistency into our method when synthesizing a sequence of new views. There might be shifting pixels between the synthesized images, especially for thin objects. We expect these limitations can be handled in future works.

6. Conclusion

In this paper, we have proposed a novel geometry-based framework for novel view synthesis. Different from conventional image-based rendering methods, we combine geometry estimation and image synthesis in an end-to-end framework. By doing so, inaccurately estimated geometry can be corrected by image synthesis error during training. Our major contribution as well as the central innovation is that we estimate the target-view depth and source-view visibility in an end-to-end self-supervised manner. Our network is able to generalize to unseen data without further fine-tuning. Experimental results demonstrate that our generated images have higher-quality than the recent state-of-the-art.

7. Acknowledgments

This research is funded in part by the ARC Centre of Excellence for Robotics Vision (CE140100016) and ARC-Discovery (DP 190102261). The first author is a China Scholarship Council (CSC)-funded PhD student to ANU. We thank all anonymous reviewers and ACs for their constructive suggestions.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. 5

[2] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001. 2

[3] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 5

[4] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7781–7790, 2019. 1, 2, 3, 5, 6, 7

[5] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 11–20, 1996. 2

[6] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 2

[7] Andrew Fitzgibbon, Yonatan Wexler, and Andrew Zisserman. Image-based rendering using image-based priors. *International Journal of Computer Vision*, 63(2):141–151, 2005. 2

[8] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2, 5

[9] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 2

[10] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 1, 2

[11] Yunzhong Hou, Liang Zheng, and Stephen Gould. Multiview detection with feature perspective transformation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2020. 2

[12] 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

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 5

[14] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 5

[15] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2

[16] Hongdong Li and Richard Hartley. Inverse tensor transfer with applications to novel view synthesis and multi-baseline stereo. *Signal Processing: Image Communication*, 21(9):724–738, 2006. 3

[17] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Geometry-aware deep network for single-image novel view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4616–4624, 2018. 5

[18] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[19] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020. 2, 6

[20] Phong Nguyen-Ha, Lam Huynh, Esa Rahtu, and Janne Heikkila. Sequential neural rendering with transformer. *arXiv preprint arXiv:2004.04548*, 2020. 2

[21] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020. 2

[22] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 3500–3509, 2017. 2

[23] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017. 1, 2

[24] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *European Conference on Computer Vision*, pages 623–640. Springer, 2020. 1, 2, 3, 5, 6, 7

[25] Gernot Riegler and Vladlen Koltun. Stable view synthesis. *arXiv preprint arXiv:2011.07233*, 2020. 8

[26] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 3

[27] 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*, pages 501–518. Springer, 2016. 3

[28] Steven M Seitz and Charles R Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30, 1996. 2

[29] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242, 1998. 2

[30] Yujiao Shi, Dylan Campbell, Xin Yu, and Hongdong Li. Geometry-guided street-view panorama synthesis from satellite imagery. *arXiv preprint arXiv:2103.01623*, 2021. 2

[31] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. In *Advances in Neural Information Processing Systems*, pages 10090–10100, 2019. 2

[32] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. *arXiv preprint arXiv:2005.03860*, 2020. 2

[33] Yujiao Shi, Xin Yu, Liu Liu, Tong Zhang, and Hongdong Li. Optimal feature transport for cross-view image geo-localization. In *AAAI*, pages 11990–11997, 2020. 2

[34] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8028–8038, 2020. 2

[35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[36] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 2

[37] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, pages 1121–1132, 2019. 2

[38] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 175–184, 2019. 2

[39] Shao-Hua Sun, Minyoung Huh, Yuan-Hong Liao, Ning Zhang, and Joseph J Lim. Multi-view to novel view: Synthesizing novel views with self-learned confidence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 155–171, 2018. 2

[40] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. *International Journal of Computer Vision*, 32(1):45–61, 1999. 7

[41] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[42] Justus Thies, Michael Zollhöfer, Christian Theobalt, Marc Stamminger, and Matthias Nießner. Image-guided neural object rendering. In *International Conference on Learning Representations*, 2019. 1, 2

[43] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 2

[44] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 2

[45] Jianfeng Yan, Zizhuang Wei, Hongwei Yi, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Dense hybrid recurrent multi-view stereo net with dynamic consistency checking. In *European Conference on Computer Vision*, pages 674–689. Springer, 2020. 4

[46] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020. 1

[47] 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, 5

[48] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019. 1

[49] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields, 2020. 2, 6

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[51] Yiran Zhong, Yuchao Dai, and Hongdong Li. Stereo computation for a single mixture image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–450, 2018. 2

[52] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2, 5

[53] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pages 286–301. Springer, 2016. 2