InfiniteNature-Zero: Learning Perpetual View Generation of Natural Scenes from Single Images — Supplementary Material —

Zhengqi Li[1], Qianqian Wang[1,2], Noah Snavely[1], and Angjoo Kanazawa[3]

[1] Google Research
[2] Cornell Tech, Cornell University
[3] UC Berkeley

1 Motivations

One additional motivation for our work is that using Internet photo collections of nature scenes for training generative models potentially leads to better results than using current posed video datasets. In particular, we trained StyleGAN2-ADA [6] models on both the LHQ (photo collection) and ACID (posed video frames) datasets, and show randomly sampled results from both models in Figure 1. One can see that generated samples from the model trained on the Internet photos which constitute LHQ are more realistic and diverse than ones sampled from the ACID video–trained model, suggesting the potential use of Internet photos for improving 3D generative view synthesis of nature scenes.

2 Inaccuracy from mono-depth and camera poses

In the original Infinite Nature paper [8], the authors render target viewpoints using Structure from Motion (SfM)–aligned disparity and the corresponding SfM camera poses during both training and evaluation. However, we found that, despite imagery being rendered from the ground truth camera pose, pixel misalignment between the ground truth and rendered images can still occur due to a number of factors, including inaccurate camera pose estimates and monocular depth outputs. In Fig. 2, we show a histogram of normalized disparity error between the SfM sparse point clouds and the SfM-aligned mono-depth over different video clips. We found that there are 25.3% of video clips whose median disparity error is more than 0.75, a threshold we found often indicates inaccurate camera poses or mono-depth via manual inspection of SfM reconstruction or depth maps. In contrast, our approach, which runs training and inference on single-view photos, and evaluates at run-time against 3D Photo–generated pseudo-ground truth, will yield results that are not influenced by such inconsistencies between ground truth camera poses, monocular depth, and rendered imagery.

3 Self-supervised view synthesis

To sample virtual camera viewpoints for self-supervised view synthesis task, we sample relative camera rotations and translation within specified range. In

Fig. 1: **Comparisons StyleGAN2-ADA sampled results from the LHQ and ACID datasets.** We show random samples generated by StyleGAN2 models trained on the LHQ (top row) and ACID (bottom row) datasets. The generated samples from the LHQ-trained model are more realistic compared to those of the ACID-trained model.

particular, we uniformly sample z component of translation vector in $[0, 10]$, and uniformly sample x, y components of translation vector in $[0, 5]$. We parameterize the camera rotation as look at direction, in which we uniformly the horizontal and vertical components within the viewing cone corresponding to 0.05 of image width and height.

To aid refinement network learn to fill in contents in the missing regions (e.g. near disocculusions) after rendering stage, we compute a binary mask M_t by applying thresholding to the gradient magnitude of the disparity warped through a virtual relative camera pose. In particular, we set mask to be zero where the gradient magnitude of the disparity is larger than α, where we sample $\alpha \in [0.35, 0.5]$ during training, and set $\alpha = 0.4$ during inference.

4 Adversarial perpetual view generation

We adopt the auto-pilot algorithm from Liu *et al.* [8] during both training and inference when generating a long camera trajectory from an input RGBD image, in order to avoid the camera crashing into the ground or mountains, or diverging off towards the sky. In particular, at each step, the auto-pilot algorithm determines the yaw and pitch angles of the subsequent relative camera pose by examining the proportion of sky pixels in the current generated view, and determines the vertical component of translation based on the proportion of nearby pixels in the current generated view. Specifically, it empirically defines sky regions as pixels whose disparity value is less than 0.08 and near regions as pixels whose disparity value is larger than 0.4. The auto-pilot algorithm will (1) turn camera to the left or right, such that it moves towards a view having balanced sky regions, (2) turn camera up or down, such that the view has a τ_{sky} fraction of sky regions, (3)

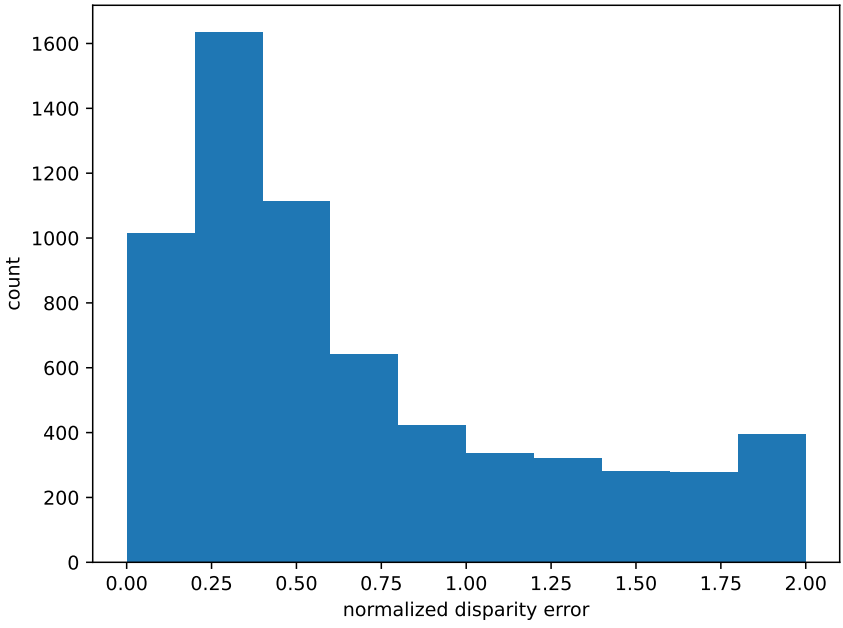

Fig. 2: **Disparity alignment error histogram from SfM on the ACID dataset.** The vertical axis indicates number of video clips, and horizontal axis indicates normalized disparity error between SfM sparse point clouds and SfM-aligned mono-depth.

translate camera up or down, such that the view observes τ_{near} fraction of nearby regions. To ensure a smooth camera trajectory, we compute the next camera pose by interpolating between the next desired relative camera pose and a moving average of previous camera poses. We refer readers to Infinite Nature for more details [8]. During training, we randomly perturb the threshold values of the near-pixel fraction $\tau_{\text{near}} \in [0.2, 0.4]$ and the sky-pixel fraction $\tau_{\text{sky}} \in [0.25, 0.45]]$ for generating different camera trajectories from an input image. During inference, we set $\tau_{\text{near}} = 0.25$ and $\tau_{\text{sky}} = 0.1$. Furthermore, since imagery in LHQ has unknown camera intrinsics, during training we randomly sample the field of view (FoV) of the input image between 45 degrees and 70 degrees, and set the FoV to 55 degrees during testing.

5 Global sky correction

Recall from Section 3.3 of the main manuscript that we compute soft sky masks based on semantic segmentation and disparity maps to perform global sky correction during inference. In particular, we first create a sky mask for the starting view, by identifying pixels whose semantic label is “sky” or “clouds”

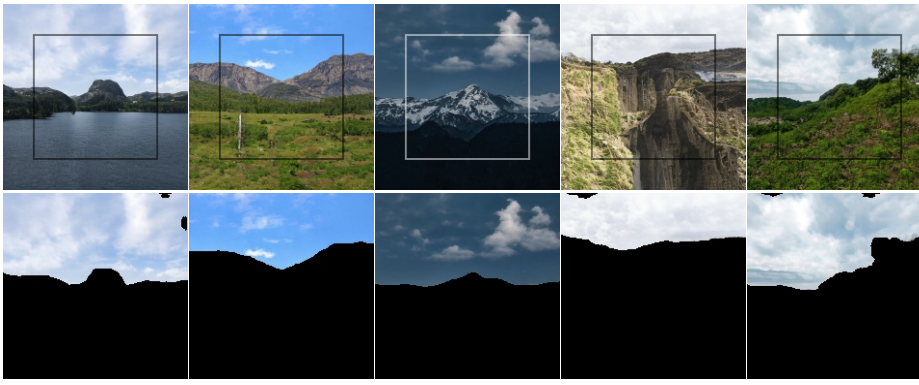

Fig. 3: **Expanded input images and detected sky regions.** We expand input images through GAN inversion to create a canvas with a larger FoV as shown in the first row, where regions within the boxes are the original input view. We then identify sky region of the starting view based on both estimated sky segmentation and disparity maps. These sky masks are shown in the second row.

from a semantic segmentation method [2], and whose corresponding disparity value is also less than 0.08. We then compute a adaptive maximum sky disparity threshold β_{sky} from the 90th percentile of determined sky regions from the input disparity. During view generation, we create a candidate sky mask for the current generated view by thresholding the corresponding refined disparity map by this threshold β_{sky}. The candidate sky mask for current generated view is then multiplied by the homography-warped sky mask from the starting view (warped according to the camera rotation's effect on the plane at infinity) to produce a final sky mask for the current generated view. We finally create a soft version of the determined sky mask by applying a Gaussian blur, then using alpha blending to composite the homography-warped sky contents at starting view with the foreground contents at the current view using the soft mask.

In addition, we expand the input image through GAN inversion [4,3] to seamlessly create a canvas of higher resolution and field of view, in order to avoid unnecessarily outpainting sky pixels from the same viewing ray during inference. We show expanded canvases of input views and corresponding sky regions in Fig. 3.

6 Details of experiment setup

Network details. We adopt a variant of the conditional StyleGAN model, CoMod-GAN [13], as our backbone refinement module F_θ, as described in the main manuscript. In particular, this model consists of a global encoder E_g, a multi-scale feature extractor E_f, and a StyleGAN generator G. We encode the input RGBD image (I_0, D_0) through E_g to produce a global latent code $z_0 = E_g(I_0, D_0)$. At every render-refine-repeat step, we feed rendered RGBD images from the

previous viewpoint to the feature extractor E_f to produce multi-scale features, which will then be added to the corresponding features layers of G, via a skip connection. The induct basis from skip connections guarantees our predictions preserve structural information of input images. The intermediate features of the generator G are then modulated by a feature vector consisting of a concatenation of global latent code z_0 and a latent code z mapped from a random Gaussian noise through a MLP. We adopt the same StyleGAN2 discriminator architecture for our discriminator, global encoder and multi-scale feature extractor. Formally, we define the generation process at each step as follows:

$$(\hat{I}_t, \hat{D}_t) = G\left(E_f(\tilde{I}_t, \tilde{D}_t), z_0 \oplus z\right) \tag{1}$$

$$\text{where} \quad z_0 = E_g(I_0, D_0), \tag{2}$$

$$z = MLP(n), \quad n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{3}$$

Super-resolution module. To enable high-resolution synthesis results for higher quality results, we train an extra super-resolution module F_{SR} on top of our pretrained refinement model F_θ with frozen weights, where the inputs of F_{SR} are generated views at 128×128, and the outputs are high-resolution ones at 512×512. Note that directly training on high-resolution imagery would be extremely computational expensive. For instance, it would take estimated 20 days to train a model at resolution of 512×512 using 8 A100 GPUs. Instead, we observe that training an extra super-resolution model only takes one extra day and that the produced results do not suffer significant temporal inconsistency. Interestingly, while we do not have paired high-resolution ground truth sequences, we can still apply the same idea as described in the main manuscript to learn super-resolution. In particular, we perform the same self-supervised view synthesis and adversarial view generation strategies described in the Section 3.1 and Section 3.2 of main manuscript, where we apply a reconstruction loss between a ground truth high-resolution image and predicted high-resolution images in the self-supervised view synthesis stage, and apply a adversarial loss between the starting real image and generated view along a virtual camera trajectory. Additionally, to encourage high-resolution predictions to be structurally consistent with low-resolution inputs, we add an extra reconstruction loss that enforces the downsampled versions of predicted high-resolution images to be consistent with low-resolution ones generated from pre-trained F_θ. In summary, our loss for training super-resolution module is as follows:

$$\mathcal{L}^{F_{SR}} = \mathcal{L}_{\text{adv}}^{F_{SR}} + \lambda_1 \mathcal{L}_{\text{rec}}, \quad \mathcal{L}^{D_{SR}} = \mathcal{L}_{\text{adv}}^{D_{SR}} + \lambda_2 \mathcal{L}_{R_1} \tag{4}$$

where $\mathcal{L}_{\text{adv}}^{F_{SR}}$ and $\mathcal{L}^{D_{SR}}$ are non-saturated GAN losses [5] applied to the predictions and corresponding real images at starting view; $\mathcal{L}_{\text{rec}}$ is a VGG perceptual loss that encourages generated views to be consistent with real high-resolution ground truth at self-supervised view synthesis stage, and enforces downsampled predictions to be consistent with low-resolution RGB input at adversarial perpetual view generation stage. We set $\lambda_1 = 1$ throughout all of experiments, and set weight of

discriminator gradient penalty term $\lambda_2 = 9.6$ and $\lambda_2 = 0.15$ for the LHQ and ACID datasets, respectively.

We use the same Co-Mod GAN architecture for super-resolution module, where we co-modulate the intermediate features of super-resolution module through the same latent codes z_0 and z obtained from the low-resolution refinement network. We do not apply dual discrimination technique as shown in recent Style-NeRF paper [1], since we observe it always leads to GAN training divergence for nature scenes.

More implementation details. We use softmax splatting [9] to perform 3D point cloud rendering with estimated disparity map, where warped pixels reaching the same location will be weighted according to their corresponding normalized depth values, and we perform $SE(3)$ transformation to adjust the warped disparity value at the target viewpoint accordingly. We train all our models using Adam optimizer [7] with initial learning rate 1×10^{-3} and perform linear decay until we reach learning rate of 1×10^{-4}. During training, when we apply adversarial loss between prediction at last camera viewpoint c_T and corresponding starting real image, we detach the gradient back-propagation from prediction at c_T to prediction at c_{T-1}, which we observe lead to more stable and efficient training. For fair comparisons, all the methods use depth from MiDas-V2 [10] for training and inference. When we evaluate different baselines, for real and generated images whose aspect ratio and resolution are different, we perform central crops followed by resizing into 128×128 so that outputs of all methods are evaluated under the same settings.

During both training and inference stages, we filter out images that are not suitable for creating 3D fly-through video using estimated disparity maps. In particular, we filter out input images whose minimum raw disparity value is larger than 200, since we observe that such images usually correspond to images looking down to the grounds, close-up, or images having incorrect disparity, which causes camera quickly crash into obstacles after a few steps. We use pretrained StyleGAN2 generator to generate test images on the LHQ dataset, and use original real images for inference on the ACID dataset.

3D Photo–generated pseudo-ground truth. On both LHQ and ACID testsets, we create pseudo ground truth images over a camera trajectory of length 5 using the technique of 3D Photos [11]. Specifically, we build a global mesh represented as layer depth images (LDI) from each training images, we then render 5 target views using autopilot algorithms from the the same LDI mesh. This process guarantees global consistency of generated views with proper contents inpainted at missing regions. We finally perform anti-alias downsampling to each rendered view to a resolution of 128×128 to mitigate blurriness and rendering artifacts. In Figure 4, we show examples of input image and corresponding pseudo ground truth images over 5 steps.

Image based rendering. To ensure that we are able to have smooth videos with high frame rate, following Infinite Nature, we perform depth-aware image based rendering to synthesize images at intermediate viewpoints from the two

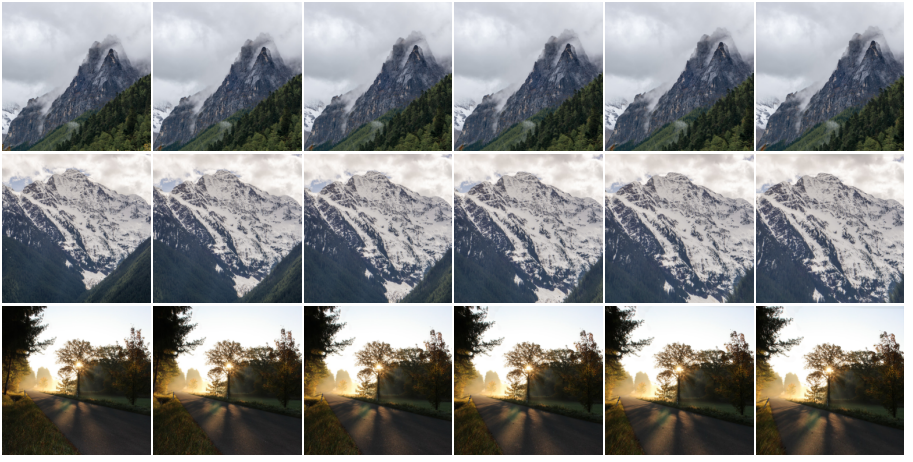

Fig. 4: **3D Photo pseudo ground truths.** We generate pseudo ground truth images from 3D Photo algorithm along a camera trajectory of length 5. From left to right, we show input images and corresponding rendered views over 5 steps from constructed global LDI meshes.

nearby views predicted by our models along the camera trajectory. Note that we only apply this technique in our long trajectory demo in the supplementary video for the purpose of better visualization, and do not apply it at all in all our evaluation.

Running time. At inference stage, our method takes 0.037 second/step to generate a novel view at resolution 512×512 using a single A100 GPU, which demonstrates potential of our approach in real-time applications.

7 Additional results

Perpetual view generation from different camera motions. Our method can also be trained to generate long camera trajectories with different camera motions such as moving backward or panning from a starting image. We show perpetual view generation results of backward and panning motions over 500 steps in Fig. 5.

Results from DIGAN. In Fig. 6, we show results from ACID-trained DIGAN [12], a state-of-the-art video generation model, over 100 steps. Note that DIGAN does not have ability for explicit viewpoint control, and is only able to generate sequences within limited viewpoints, or completely fail to produce plausible views.

Plots of style consistency and $\mathbf{FID_{sw}}$. We show the plots of style consistency error and sliding window FID (FID_{sw}) over 50 steps on the two datasets, as shown in Figure 7. On the LHQ dataset, we show comparisons between ours

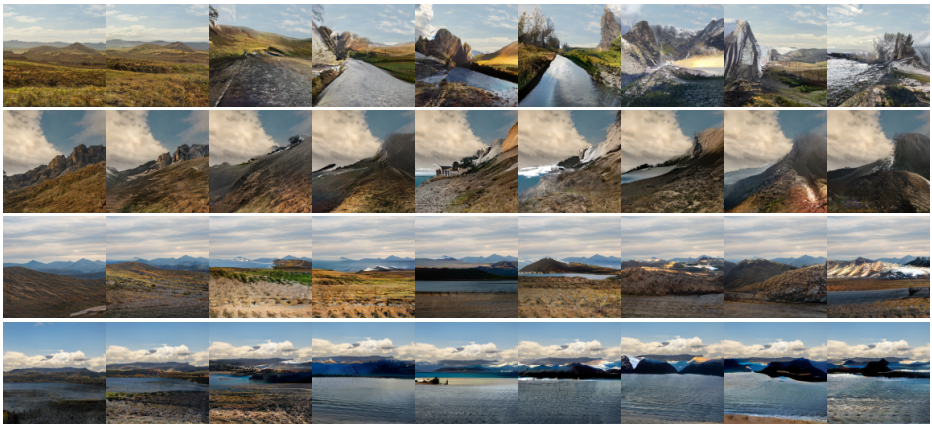

Fig. 5: **Perpetual view generation results from different camera motions.** Our model can also be trained with different motion patterns to generate plausible views over long camera trajectories. The top two rows show results generated from backward camera motions. The bottom two rows show results generated from left and right panning, respectively. All the results are generated over camera trajectories length of 500.

(full) and the naive baseline as described in the main manuscript. On the ACID dataset, we show comparisons between our approach and Infinite Nature. Figure 7 show that our approach has better style consistency, and generated views are more realistic and diverse over time.

8 Limitations

Unseen camera motions. Since our method was trained using auto-pilot algorithms, our method can fail to generate plausible views from an unseen camera trajectory such as pure rotations, a similar issue which was also observed by original Infinite Nature work.

Motion planning. Although auto-pilot algorithms are effective most of time, Due to its heuristic natures, we observe it can still lead to camera crashing into obstacles over a long camera trajectory, as shown in the first row of Figure 8. This is due to the fact that approaching speed of generated contents can be faster than camera turning speed determined by the auto-pilot algorithms. Designing a better learning based motion planning algorithms for controlling camera viewpoints during long term generation process is an interesting future direction.

Inputs. Inaccurate mono-depth estimate can cause inaccurate depth of sky as well as inaccurate sky identification. This issue is mainly manifested in the images at sunset or night times that are barely seen by mono-depth or semantic segmentation networks. As a result, this can generate unrealistic view sequences,

Fig. 6: **Results from DIGAN.** We show results of unconditional video generation method DIGAN [12] over 100 steps. Note that DIGAN does not have camera viewpoint control and can only produce video sequence within limited viewpoint changes.

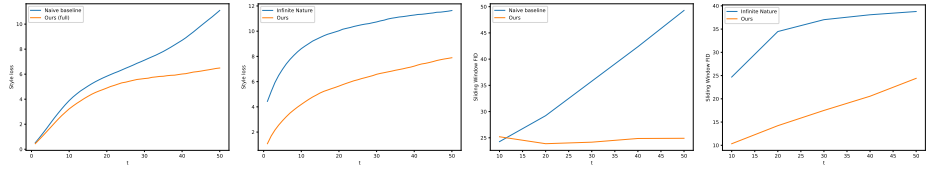

Fig. 7: **Comparisons of style loss and FID_{sw} over 50 steps**. From left to right, we show (a) style loss comparisons between ours (full) and naive baselines on the LHQ dataset, (b) style loss comparisons between ours and Infinite Nature [8], on the ACID dataset, (c) FID_{sw} comparisons between ours (full) and naive baseline on the LHQ dataset, (d) FID_{sw} comparisons between ours and Infinite Nature [8] on the ACID dataset.

as shown in the second row of Fig. 8, where part of sun turns into foreground objects during view generation.

Global consistency. Similar to prior video and view generation methods, our method only keeps global consistency of background, but does not ensure global consistency of foreground contents. Addressing this issue requires designing a generative model that enables synthesizing entire 3D models of nature landscapes, which is beyond the scope of this work and capability of current state-of-the-art. However, we believe this challenge is an exciting future directions to pursue.

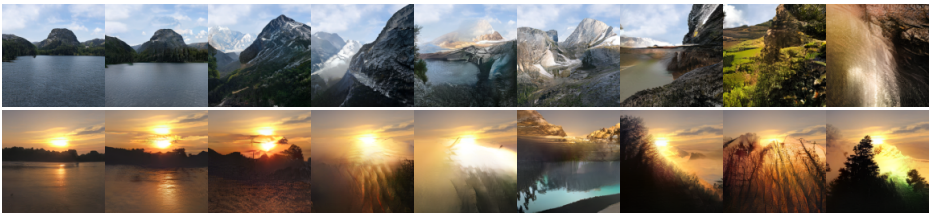

Fig. 8: **Limitations from motion planning and inputs.** Auto-pilot algorithms sometimes cannot avoid camera crashing into mountains, lands or obstacles (left). Furthermore, generated views can be less realistic if mono-depth or sky segmentation maps from input image is inaccurate (right). We show failure examples generated over 500 steps.

References

1. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3d generative adversarial networks. In: Proc. Computer Vision and Pattern Recognition (CVPR) (2022)
2. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
3. Cheng, Y.C., Lin, C.H., Lee, H.Y., Ren, J., Tulyakov, S., Yang, M.H.: In&out: Diverse image outpainting via gan inversion. In: Proc. Computer Vision and Pattern Recognition (CVPR) (2022)
4. Chong, M.J., Lee, H.Y., Forsyth, D.: StyleGAN of All Trades: Image Manipulation with Only Pretrained StyleGAN. arXiv preprint arXiv:2111.01619 (2021)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Neural Information Processing Systems (2014)
6. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proc. Computer Vision and Pattern Recognition (CVPR). pp. 8110–8119 (2020)
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proc. Int. Conf. on Learning Representations (ICLR) (2015)
8. Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In: Proc. Int. Conf. on Computer Vision (ICCV). pp. 14458–14467 (2021)
9. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: Proc. Computer Vision and Pattern Recognition (CVPR). pp. 5437–5446 (2020)
10. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. Trans. Pattern Analysis and Machine Intelligence (2020)
11. Shih, M.L., Su, S.Y., Kopf, J., Huang, J.B.: 3D photography using context-aware layered depth inpainting. In: Proc. Computer Vision and Pattern Recognition (CVPR) (2020)
12. Yu, S., Tack, J., Mo, S., Kim, H., Kim, J., Ha, J.W., Shin, J.: Generating videos with dynamics-aware implicit generative adversarial networks. In: Proc. Int. Conf. on Learning Representations (ICLR) (2022)

13. Zhao, S., Cui, J., Sheng, Y., Dong, Y., Liang, X., Chang, E.I., Xu, Y.: Large scale image completion via co-modulated generative adversarial networks. In: Proc. Int. Conf. on Learning Representations (ICLR) (2021)