Crowdsampling the Plenoptic Function

Zhengqi Li(✉), Wenqi Xian, Abe Davis, and Noah Snavely

Cornell Tech, Cornell University, Ithaca, USA
zl548@cornell.edu

Abstract. Many popular tourist landmarks are captured in a multitude of online, public photos. These photos represent a sparse and unstructured sampling of the plenoptic function for a particular scene. In this paper, we present a new approach to novel view synthesis under time-varying illumination from such data. Our approach builds on the recent *multi-plane image* (MPI) format for representing local light fields under fixed viewing conditions. We introduce a new *DeepMPI* representation, motivated by observations on the sparsity structure of the plenoptic function, that allows for real-time synthesis of photorealistic views that are continuous in both space and across changes in lighting. Our method can synthesize the same compelling parallax and view-dependent effects as previous MPI methods, while simultaneously interpolating along changes in reflectance and illumination with time. We show how to learn a model of these effects in an unsupervised way from an unstructured collection of photos without temporal registration, demonstrating significant improvements over recent work in neural rendering. More information can be found at crowdsampling.io.

1 Introduction

There is a thought experiment that goes something like this:

> Imagine a 'camera' with no optics or image sensor of any kind. Rather, it consists only of a box equipped with GPS, a radio for Internet access, a button for 'taking pictures', and a screen for displaying those pictures. When a user presses its button, the box searches the Internet for photos tagged with its current location, and from these selects a best match to display on the screen.

This thought experiment is perhaps best understood in the context of popular tourist attractions, of which one can often find countless images posted online (Fig. 1, second row). When pointed at such an attraction, one can imagine our box producing images very similar to those of a real camera, forcing us to consider

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58452-8_11) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12346, pp. 178–196, 2020.
https://doi.org/10.1007/978-3-030-58452-8_11

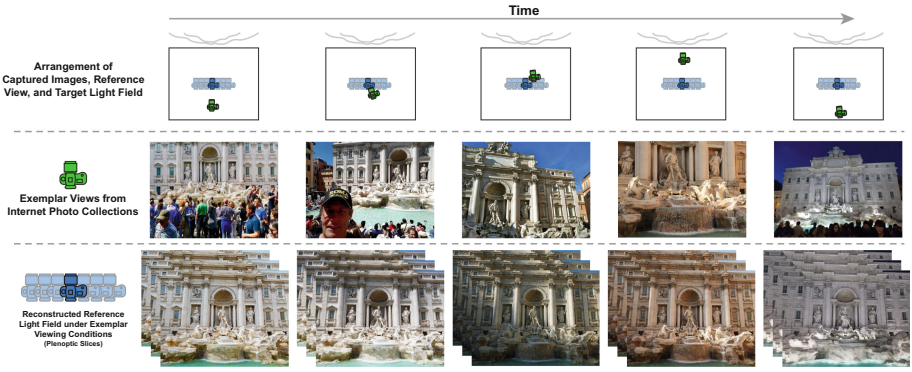


Fig. 1. Crowdsampled plenoptic slices. Given a large number of tourist photos taken at different times of day, our system learns to construct a continuous set of lightfields and to synthesize novel views capturing all-times-of-day scene appearance.

whether an image we capture ourselves is meaningfully different from a near-identical one captured by strangers. For many, the ensuing philosophical debate hinges on whether an image reflects the scene as they remember it. After all, appearance is not generally constant over time, even under a fixed geometry and viewpoint; in outdoor settings, for example, weather changes, shadows move, and day turns to night—all resulting in appearance changes that can be observed from a single view of the scene.

This poses an interesting challenge to the field of image-based rendering: can we use crowdsourced imagery to synthesize arbitrary views of a scene with viewing conditions that change over time? Without changing viewing conditions, this challenge would reduce to the more familiar problem of reconstructing a 4D light field $\mathcal{L}(u, v, x, y)$ that describes all light in our scene [33]. When we add time to our problem, it turns into a 5D reconstruction over what Adelson and Bergen [1] call the *plenoptic function*.[1]

In this paper, we propose a novel approach to neural image-based rendering from crowdsourced images that leverages the sparse structure of the plenoptic function to learn how scene appearance changes over space and time in an unsupervised manner. Our approach takes unstructured Internet photos spanning some range of time-varying appearance in a scene and learns how to reconstruct a *plenoptic slice*—a representation of the light field that respects temporal structure in the plenoptic function when interpolated over time—for each of the viewing conditions captured in our input data. By designing our model to preserve the structure of real plenoptic functions, we force it to learn time-varying phenomena like the motion of shadows according to sun position. This lets us, for example, recover plenoptic slices for images taken at different times of day (Fig. 1, bottom row) and interpolate between them to observe how shadows

[1] [1] describes the plenoptic function as 7D, but we can reduce this to a 4D color light field supplemented by time by applying the later observations of [33].

move as the day progresses (best seen in our supplemental video). In effect, we learn a representation of the scene that can produce high-quality views from a continuum of viewpoints *and* viewing conditions that vary with time.

Our work makes three key contributions: first, a representation, called a DeepMPI, for neural rendering that extends prior work on multiplane images (MPIs) [68] to model viewing conditions that vary with time; second, a method for training DeepMPIs on sparse, unstructured crowdsampled data that is unregistered in time; and third, a dataset of crowdsampled images taken from Internet photo collections, along with details on how it was collected and registered.

Compared with previous work, our approach inherits the advantages of recent methods based on MPIs [8,12,42,56,68], including the ability to produce high-quality novel views of complex scenes in real time and the view consistency that arises from a 3D scene representation (in contrast to neural rendering approaches that decode a separate view for each desired viewpoint). To these advantages we add the key ability to synthesize and interpolate continuous, photo-realistic, time-varying changes in appearance. We compare our approach both quantitatively and qualitatively to recent neural rendering methods, such as Neural Rerendering in the Wild [41], and show that our method produces superior results.

2 Related Work

The study of image-based rendering is motivated by a simple question: how do we use a finite set of images to reconstruct an infinite set of views? Different branches of research have explored this question from different angles and with different assumptions. Here we outline the space of approaches, highlighting work most closely related to our own.

Novel View Synthesis. Novel view synthesis has traditionally been approached through either explicit estimation of scene geometry and color [4,21,72], or using coarser estimates of geometry to guide interpolation between captured views [2,10,55]. Light field rendering [3,17,33] pushes the latter strategy to an extreme by using dense structured sampling of the light field to make reconstruction guarantees independent of specific scene geometry. Subsequent works [9,32,44,50,51,61] have leveraged observations on the structure of light fields to build on this approach. However, most IBR algorithms are designed to model static appearance, making them ill-suited for our problem.

Recently, deep learning techniques have been applied to this problem. Several works [22,59] rely on global meshes to guide view synthesis. However, such methods heavily rely on the accuracy of 3D models, and often fail to model complex scene components such as translucent and thin objects. Other works predict appearance flow [69], depth probabilities [13,65], or RGBD light fields [26,57]. However, many of these methods independently synthesize appearance for each view, leading to inconsistent renderings across views.

Our approach builds on the use of multiplane images (MPIs) [68] for novel view synthesis. Several recent methods have shown that MPIs are an effective and

learnable representation for light fields [8,12,42,56]. We build on this representation by introducing the DeepMPI, which further captures viewing condition–dependent appearance. We are also inspired by recent work that poses view synthesis as decoding features from a learned latent space [7,11,36,53,54,59]. However, such work has been limited to synthetic environments or objects captured in controlled settings and is difficult to apply to crowdsampled images.

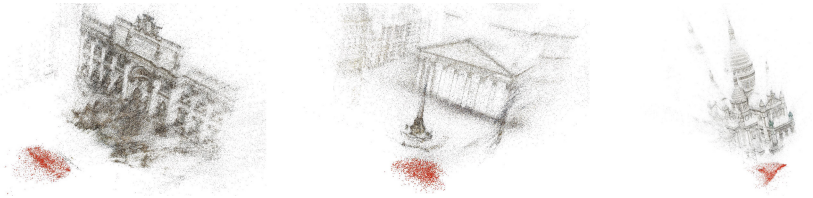
Appearance Modeling. Several works have modeled the time-varying appearance of outdoor scenes using physically-motivated approaches [19,29,48] or by combining data-driven methods and dense geometry [14,40,45,66]. Additionally, Martin-Brualla *et al.* [38,39] reconstruct time-lapses of urban scenes from Internet photos. However, their method relies on timestamps, and models appearance changes at much coarser granularity (scene dynamics across years). The recent work of Meshry *et al.* [41] is probably closest to our own. They model appearance changes across varying times of day by learning an appearance embedding. However, their method relies heavily on dense multi-view stereo geometry, and tends to produce temporal artifacts under complex appearance changes. In contrast, our approach is capable of rendering a more continuous range of photo-realistic views across diverse appearances, without relying on dense input geometry.

Deep Image Synthesis. Our work is also related to the problem of image-to-image translation [6,25,43,62], multi-model image-to-image translation [24,31,70,71] and style transfer [15,23,49,60]. Recently, Generative Adversarial Networks (GANs) [16,18,37] have successfully produced photo-realistic imagery, enabling a variety of applications in deep image synthesis [27,28,30,46,63,64]. However, there has been comparatively little investigation of 3D scene representations for deep image synthesis. Our method demonstrates the ability to learn a generative 3D scene representation and produce high-quality novel views of complex scenes.

3 Approach

Given a set $\mathcal{I} = \{I_1, I_2, ..., I_n\}$ of crowdsampled photos with corresponding camera viewpoints $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ captured in a common scene, we formulate our problem as the reconstruction of *plenoptic slices* (local light fields parameterized by an appearance descriptor) around some reference view r conditioned on each of the scene appearances captured in $\mathcal{I}$ (see Fig. 1 for a geometric sketch of this setup). We present our approach in three parts: first, we describe how the input images $\mathcal{I}$ are collected and registered (Sect. 3.1); then we discuss our representation of the plenoptic function, which extends multiplane images (MPIs) to model appearance changes over time (Sect. 3.2); and finally we describe how to train this representation on our crowdsampled data (Sects. 3.3 and 3.4).

Note on Notation: Throughout the paper, we will use superscripts to denote camera viewpoints and subscripts to denote image or voxel indices.



(a) Trevi Fountain (b) The Pantheon (c) Sacre Coeur

Fig. 2. Registered photo collections. Example SfM reconstructions of clusters of Internet photos sharing similar viewpoints, labeled as red dots. (Color figure online)

3.1 Collecting Crowdsampled Data

We selected a number of popular tourist sites and downloaded ∼50K photos from Flickr for each site. For each scene, we must then register these photos by solving for a camera pose and intrinsic parameters for each image. As running structure from motion (SfM) from scratch on such quantities of images is very expensive, we instead started with a existing SfM reconstruction of each site from the MegaDepth dataset [35], and performed camera relocalization to efficiently register each new image against the existing reconstruction [47].

For each landmark, we then identified a reference viewpoint r to center our reconstruction by using a canonical view selection algorithm similar to that of Simon *et al.*to find viewpoints with a high density of nearby views [52]. We then select all images captured from within a sphere centered at r for use in our method, randomly splitting the set gathered from each landmark into training and test data. We manually set the field of view of the reference viewpoint so that it has good coverage of the scene.

We found that the camera parameters estimated from relocalization are sometimes inaccurate, and so we reapply a global SfM and bundle adjustment to the smaller set of selected images near each scene's reference view to reestimate these images' camera parameters. We used this data pipeline to gather and register photos for eight locations, and will release this data to the research community. Figure 2 shows final SfM reconstructions for three of these landmarks.

3.2 The DeepMPI Scene Representation

We base our representation on the multiplane image (MPI) format [58,68], which represents light fields locally as a stack of fronto-parallel planar RGBα layers arranged at varying distances from the camera, akin to a stack of transparencies. Novel views are rendered from an MPI by warping the layers into a new view, then performing an *over* operation to composite the warped layers into a rendered image. Individual RGBα elements ("voxels") of an MPI are indexed by (x, y) position and plane depth d.

While MPIs have been remarkably effective for reconstructing fixed light fields from sparse views [12], they do not encode any information about how

viewing conditions may vary with time. Furthermore, even if we were given a regular MPI corresponding to viewing conditions for each of our input images, directly interpolating between these MPIs would still fail to capture temporal structure in the plenoptic function. For example, interpolating between morning and afternoon MPIs would cause shadows cast by the sun to appear in duplicate when, in reality, a single shadow moved over time. This observation highlights the distinction between what we call a light field and what we call a plenoptic slice: we use the latter to describe a reparameterization of the light field that is better-suited for interpolation over time.

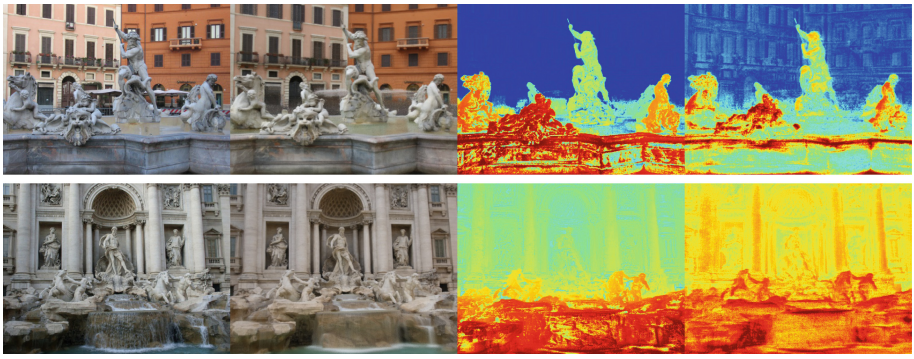
Inspired by DeepVoxels [53], we introduce *DeepMPIs* to help learn this reparameterization. DeepMPIs augment standard RGBα MPIs by appending a *learnable* latent feature vector at each MPI voxel (see Fig. 4). For a given scene, we position a DeepMPI at the reference viewpoint r, and denote this reference DeepMPI as $D^r = (B^r, \alpha^r, F^r)$. Each voxel of D^r at spatial location and depth $\mathbf{p} = (x, y, d)$ consists of a base RGB color $B^r_{\mathbf{p}}$, an alpha weight $\alpha^r_{\mathbf{p}}$, and a latent feature vector $F^r_{\mathbf{p}}$. We set the number of DeepMPI depth planes to 64 with uniform sampling in disparity space, and we adopt the method of Zhou *et al.* [68] to set the depth of the near and far planes of the DeepMPI.

In our supplemental document we relate the design of this representation and its training to priors on the sparse structure of the plenoptic function. At a high level, the α planes encode visibility information, which we expect to remain constant even as lighting and other viewing conditions change with time. The latent feature planes F^r are trained to capture correlations between different viewing conditions that arise from, for example, limited variation in material properties and correlation among surface normals within the scene. A plenoptic slice then consists of a DeepMPI and some exemplar image I_k. We can convert this to a standard RGBα MPI representing appearance under the specific conditions captured in I_k by using a decoder that is trained jointly with our DeepMPI, which we describe in Sect. 3.4.

To compute a DeepMPI from a collection of registered images, we use a two-stage process: first, we first estimate base color and α planes (Sect. 3.3), then optimize latent features F^r jointly with our neural rendering network (Sect. 3.4) to enable controllable, varying appearance.

3.3 Stage 1: Optimizing DeepMPI Color and α Planes

In the first stage of our method, we optimize base color planes B^r and alpha planes α^r in our DeepMPI as if it were a standard RGBα MPI. One simple approach would be to jointly optimize B^r and α^r from scratch so as to minimize a reconstruction loss over all images (i.e., the difference between a known image and an MPI-predicted image from that viewpoint, averaged over all input images). However, as described in [12], such a method exhibits slow convergence and can be prone to local minima. In addition, compared to [12], our setting is more challenging because Internet photos exhibit diversity in camera parameters and viewing conditions. Instead, we propose a simple yet effective approach to estimating B^r and α^r given a set of posed input views.



(a) viewpoint c_k (b) base color $\hat{B}^k$ (c) our depth (d) baseline depth

Fig. 3. Renderings of base color and alpha. From left to right: (a) original photos at target viewpoint c_k, (b) our estimated base color at c_k, (c) pseudo-depth computed from the RGBα MPI at c_k using our two-phase approach, (d) pseudo-depth from the baseline. For depth maps, red = close and blue = far. (Color figure online)

We start by creating a mean RGB plane sweep volume (PSV) at the reference viewpoint by reprojecting every image to the reference viewpoint via each depth plane, then averaging all reprojected images at each depth plane. We initialize the base color planes B^r to this mean RGB PSV. Keeping these color planes fixed, we optimize the alpha planes α^r to minimize reconstruction losses over the training photos. Specifically, given a photo I_k at viewpoint c_k, we project both B^r and α^r to c_k, then apply the over operation from back to front to render a base color image $\hat{B}^k$:

$$\hat{B}^k = \mathcal{O}\left(\mathcal{W}^k(B^r), \mathcal{W}^k(\alpha^r)\right), \tag{1}$$

where $\mathcal{O}$ is the over operation and $\mathcal{W}^k$ is the warping operation from the reference viewpoint r to the target viewpoint c_k. We compare the rendered base color image $\hat{B}^k$ and I_k using a reconstruction loss consisting of a pixel-wise l_1 loss and a multi-scale gradient consistency loss [34,35]. We observe that the gradient consistency loss leads to higher rendering quality and faster convergence.

Since the mean RGB PSV cannot accurately model scene content that is occluded in the reference view, after optimizing α^r with fixed B^r, we unfreeze B^r and jointly optimize B^r and α^r using the reconstruction loss described above. We observe that this two-phase training method leads to more accurate estimates of α^r than the alternative of optimizing B^r and α^r together from scratch. Figure 3, shows examples of input viewpoints and rendered base color images, as well as a comparison of pseudo-depths derived from alpha planes α^r computed by our two-phase training method and by the baseline. Once B^r and α^r are estimated, they are fixed for the subsequent stage of training, described below.

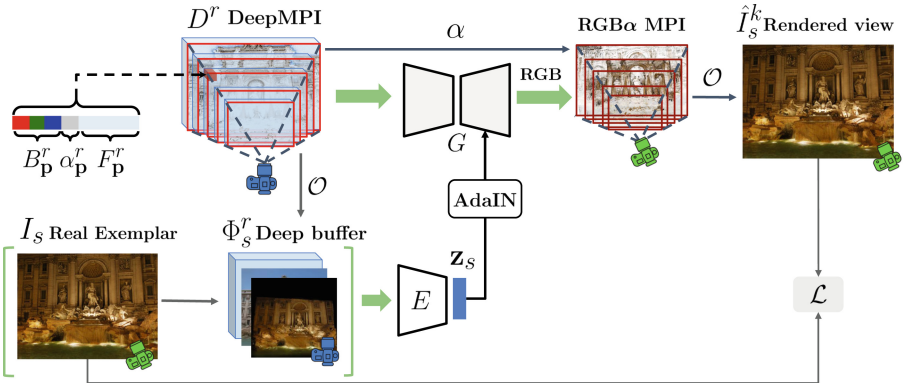


Fig. 4. Learning framework. Our method builds a reference DeepMPI D^r, consisting of base color, alpha, and latent feature components organized into planar layers. A rendering network G takes a DeepMPI projected to a target viewpoint c_k, and predicts corresponding RGB color layers. The appearance of these layers is modulated by an appearance vector $\mathbf{z}_s$ produced by encoder E. The over operation $\mathcal{O}$ is applied to the resulting RGBα MPI to render a view. We jointly train the encoder E, rendering network G, and latent features F^r in the DeepMPI by comparing a rendered view with an original exemplar image $I_k = I_s$. During inference, given an exemplar photo I_s, we can synthesize novel views close to the reference viewpoint, while also preserving the exemplar's appearance.

3.4 Stage 2: Learning How Appearance Changes with Time

Our method's second stage optimizes the latent features F^r in our DeepMPI, together with an appearance encoder E and rendering network G, to capture and render time-varying appearance. Our learning framework is summarized in Fig. 4.

Appearance Encoder. To model appearance variation, we devise a method wherein an encoder E learns to map an exemplar image I_s and an auxiliary *deep buffer* Φ_s^r to a latent appearance vector $\mathbf{z}_s$. Prior work, such as Meshry *et al.* [41], represents such variation by learning an appearance vector from the exemplar image and a deep buffer containing semantic and depth information. However, their deep buffer is aligned with the viewpoint of the exemplar image. This makes the encoding of exemplar data view-dependent when, under fixed conditions, the information (e.g., sun direction) it reflects should be largely view-independent. In contrast, we utilize a deep buffer *aligned with the reference viewpoint.*

In particular, our encoder E computes a latent appearance vector $\mathbf{z}_s$:

$$\mathbf{z}_s = E\left(I_s, \Phi_s^r\right) \tag{2}$$

where I_s is an exemplar image and Φ_s^r is a reference viewpoint–aligned deep buffer containing (1) a rectified RGB image over-composited from a PSV that reprojects exemplar I_s to the reference viewpoint via the depth planes of the

reference DeepMPI, (2) a flattened base color image over-composited from base color layers B^r, and (3) a flattened latent feature map at the reference viewpoint over-composited from DeepMPI features F^r.

Such a deep buffer allows E to learn complex appearance by aligning the illumination information in the exemplar image with the shared scene intrinsic properties encoded in the reference DeepMPI. Without such alignment, it is difficult for E to consistently establish appearance correspondence across different viewpoints. Column (d) of Fig. 5 shows examples of rendered images without use of such a deep buffer. One can see that the deep buffer guides the model to capture complex illumination effects such as the realistic shadows highlighted in the first row. Moreover, integrating the base color and latent feature map at the reference viewpoint into Φ_s^r, and adding I_s as inputs to E can help the model to extrapolate appearance outside the field of view of the exemplar image, as shown in the last row of Fig. 6.

Neural Renderer. A plenoptic slice is now represented by the reference DeepMPI D^r and an appearance vector $\mathbf{z}_s$. Given these inputs, our neural renderer G predicts the corresponding RGB color planes. We could either predict these RGB planes at the reference viewpoint, or after first warping the DeepMPI to the target viewpoint. We choose the latter because it simplifies efficient implementation, as noted below. Let D^k denote the reference DeepMPI D^r after warping into target viewpoint c_k. Given D^k and $\mathbf{z}_s$, G predicts the RGB color planes C_s^k of a standard RGBα MPI at target viewpoint c_k:

$$C_s^k = G\left(D^k,\ \mathbf{z}_s\right) \tag{3}$$

In particular, G takes in each layer of D^k *independently* and predicts a corresponding RGB layer whose appearance is controlled by $\mathbf{z}_s$. A rendered RGB image with the appearance of I_s at viewpoint c_k can then be obtained using the over operation with precomputed alpha weights α^k in D^k:

$$\hat{I}_s^k = \mathcal{O}\left(C_s^k, \alpha^k\right) \tag{4}$$

As shown in Fig. 4, during training we set exemplar image $I_s = I_k$, i.e., we aim to reconstruct image I_k at viewpoint c_k. At inference, I_s is not necessarily I_k.

Our rendering network G is a U-Net variant with an encoder-decoder architecture. Prior methods [41,71] embed $\mathbf{z}$ in the bottleneck or input of G. Instead, we use Adaptive Instance Normalization (AdaIN) layers [23] whose parameters are dynamically generated from $\mathbf{z}$ via an MLP. AdaIN has been shown to be effective in capturing both global and spatially varying appearance of exemplar images. We find that AdaIN not only helps model natural scene appearance, but also stabilizes training. Column (b) of Fig. 5 shows examples of our rendered images without AdaIN; one can see the model using AdaIN preserves more faithful scene appearance including the style and color of exemplar images.

In practice, feeding a full-resolution DeepMPI into G and performing backpropagation is very memory intensive. Hence, during training, we operate on random 256×256 crops of training images, and only the necessary portion of D^r is warped to c_k and fed to G. At test time, any size input can be used.

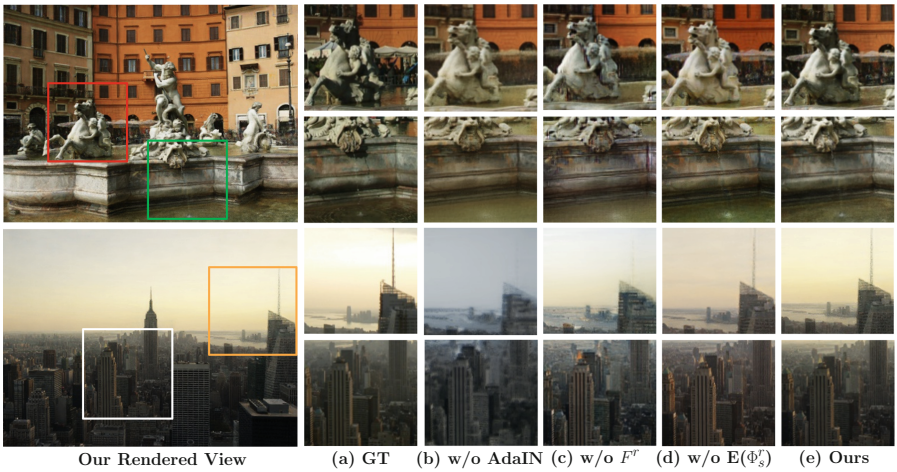


Fig. 5. Comparisons of images reconstructed with different configurations of our method. The images rendered from our full approach (e) are more similar to the ground truth images (a) than other configurations. In particular, the images rendered from the models without AdaIN (b) or the DeepMPI (c) are less realistic, and the model that does not feed the deep buffer Φ_s^r to the encoder (d) fails to capture accurate scene appearance, as indicated in the highlighted regions. (Color figure online)

Losses. To train G and E, we compute losses between output views and ground-truth exemplar views. Our training loss is composed of three terms:

$$\mathcal{L} = \mathcal{L}_{\mathsf{VGG}} + w_{\mathsf{GAN}}\mathcal{L}_{\mathsf{GAN}} + w_{\mathsf{style}}\mathcal{L}_{\mathsf{style}}, \tag{5}$$

where $\mathcal{L}_{\mathsf{VGG}}$, $\mathcal{L}_{\mathsf{GAN}}$, and $\mathcal{L}_{\mathsf{style}}$ denote VGG perceptual loss, adversarial loss, and style loss. For $\mathcal{L}_{\mathsf{VGG}}$, we adopt the formulation of [6,68]; $\mathcal{L}_{\mathsf{GAN}}$ is computed from multi-scale discriminators [62] with an objective similar to LSGAN [37].

To further enforce that the appearance of rendered images matches that of exemplar images, our style loss $\mathcal{L}_{\mathsf{style}}$ compares l_1 differences between Gram matrices constructed from VGG features at different layers. We empirically observe $\mathcal{L}_{\mathsf{style}}$ can guide our model to correctly capture the appearance of exemplar images, especially for rare photos such as those taken at sunset.

4 Experiments

We conduct extensive experiments to validate our proposed approach on our Internet photo dataset. We first compare with two baseline methods both quantitatively and qualitatively on the tasks of view synthesis, appearance transfer and appearance interpolation. We also present an ablation study to examine the impact of different configurations of our model. Finally, we perform a user study whose results demonstrate the quality of our synthesized novel views.

Table 1. Quantitative comparisons on our test set. Lower is better for l_1 and LPIPS and higher is better for PSNR. l_1 errors are scaled by 10 for ease of presentation.

	Trevi fountain			Sacre coeur			The pantheon			Top of the rock			Piazza navona		
Method	l_1	LPIPS	PSNR	l_1	LPIPS	PSNR	l_1	LPIPS	PSNR	l_1	LPIPS	PSNR	l_1	LPIPS	PSNR
MUNIT [24]	0.768	2.62	20.1	0.740	2.08	20.2	0.560	1.51	21.4	0.876	3.68	18.2	0.984	2.80	17.4
NRW [41]	0.779	2.07	20.0	0.808	1.90	19.6	0.592	1.35	21.1	0.802	2.76	19.3	1.050	2.64	17.1
w/o 2-phase	0.651	1.68	21.0	0.695	1.61	20.8	0.515	1.12	21.9	0.694	2.19	20.4	1.010	2.52	17.4
w/o AdaIN	0.780	1.87	19.8	0.801	1.89	19.6	0.609	1.30	20.9	0.773	2.58	19.3	1.150	2.97	17.1
w/o F^r	0.712	1.74	20.5	0.737	1.78	20.2	0.556	1.25	21.5	0.720	2.47	19.9	1.045	2.62	17.0
w/o $E(\Phi_s^r)$	0.670	1.70	20.9	0.715	1.66	20.5	0.549	1.16	21.5	0.703	2.24	20.0	1.017	2.52	17.2
Ours (full)	**0.618**	**1.56**	**21.8**	**0.676**	**1.57**	**21.0**	**0.495**	**1.08**	**22.5**	**0.642**	**2.48**	**20.7**	**0.933**	**2.32**	**17.6**

Data and Implementation. We evaluate our approach on five of our reconstructed scenes, which contain on average 2,064 images. For each scene, images are randomly split into training and test sets with a 85:15 ratio. We train a separate model for each scene. To mask out transient objects such as people and cars during training and evaluation, we adopt state-of-the-art semantic and instance segmentation algorithms [5,20] to create binary object masks. We set the dimension of the latent appearance vector to $\mathbf{z}_s \in \mathbb{R}^{16}$, and that of our latent DeepMPI features to $F^r_{\mathbf{p}} \in \mathbb{R}^8$. We refer readers to the supplemental material for scene statistics, network architectures, and other implementation details.

Baselines. We compare our approach to two state-of-the-art multi-modal image-to-image translation methods, adapted to our task: MUNIT [24] and Neural Rerendering in the Wild (NRW) [41]. To compare to MUNIT, we adapt their network G to predict an RGB image at the target viewpoint from a corresponding base color input, and train with a bidirectional reconstruction loss. For NRW, both E and G take as input base color, per-frame depth derived from the DeepMPI, and semantic segmentation at the target viewpoint. G then predicts a corresponding RGB image conditioned on the appearance vector extracted by E. We follow the same staged training strategy and use the same losses as in [41].

Error Metrics. Similar to [41], we report test image reconstruction errors using three error metrics: l_1 error, peak signal-to-noise ratio (PSNR), and perceptual similarity (via LPIPS [67]). Prior work has found the LPIPS metric to be better correlated with human visual perception than other metrics.

Quantitative Comparison. For fair comparison, we train and evaluate the baselines using the same data and hyperparameter settings as our method. Table 1 shows results of quantitative comparisons on our test set. Our proposed approach outperforms the two baseline methods by a large margin in terms of l_1 and PSNR, and is significantly better in terms of LPIPS, indicating that our method achieves higher rendering quality and realism.

Ablation Study. We perform an ablation study to analyze the effect of individual system components. In particular, we replace four components with simpler configurations: (1) using a train-from-scratch baseline to estimate alpha,

(a) exemplar I (b) MUNIT [24] (c) NRW [41] (d) Ours

Fig. 6. Appearance transfer comparison. From left to right: (a) exemplar images used to extract appearance vectors, (b) predictions from MUNIT [24], (c) predictions from NRW [41], (d) predictions from our method. Compared to the baselines, our rendered images are more photo-realistic and are more faithful to the appearance of the exemplar images. Please zoom in to highlighted regions for better visual comparisons.

as described in Sect. 3.3 (w/o 2-phase), (2) including $\mathbf{z}$ as an input to G rather than using AdaIN (w/o AdaIN), (3) removing latent features from the DeepMPI (w/o F^r), and (4) encoding $\mathbf{z}$ only from the exemplar image and not additionally from the deep buffer (w/o $E(\Phi_s^r)$). Quantitative results are reported in Table 1. Latent DeepMPI features, as well as the use of AdaIN in our neural renderer, yield significant improvements, and lead to better rendering quality for thin structures and attached shadows, as highlighted in Fig. 5. Encoding the reference deep buffer also yields rendered images that better match the exemplar image.

Fig. 7. Appearance interpolation. The left- and rightmost exemplar images indicate start and end appearance. Intermediate images are generated by linearly interpolating latent vectors from the two images. Odd rows show interpolation results from NRW [41], and even rows from our method. Moving shadows are indicated in highlighted regions.

Rendering with Appearance Transfer. Fig. 6 shows qualitative comparisons between our method and the two baselines on our test set in terms of rendering quality and appearance transferability (i.e., how well the model can transfer illumination and appearance of an exemplar image to a target viewpoint). We demonstrate compelling results in challenging cases such as sunset, which is a rare condition in the input photos. Compared to MUNIT, our rendered images are more realistic and exhibit fewer artifacts. For example, our rendered images successfully model specularities on glass windows, details on running water and droplets, cast shadows, and directional lighting effect as shown in the highlighted regions in Fig. 6. Our approach can also generate complex highlights and cast shadows from the sun. Compared with NRW, our rendered images are more faithful to the illumination in the exemplar image (e.g., for sunset appearance). Moreover, our approach can extrapolate appearance beyond the field of view of the exemplar image, as shown in the last row of Fig. 6. We refer readers to the supplemental video for visual comparisons with animated camera trajectories.

Appearance Interpolation. A key advantage of our method is the ability to interpolate between plenoptic slices in the latent appearance space. We conduct qualitative comparisons between our approach and NRW on appearance interpolation. We choose two images to define the start and end appearance, and linearly interpolate their latent vectors to produce in-between appearances. Figure 7 shows a comparison of interpolation results. In the first two rows of the figure, we observe that our method can simulate the progression of surfaces exposed to sunlight as the sun moves, while NRW fails to produce this effect. In the last row, our approach recovers the gradual motion of shadows throughout the day, while shadows in the NRW results tend to fade less naturally during

Fig. 8. 4D Photos. We demonstrate an application of creating *4D photos* by performing spatial-temporal interpolation in which both camera viewpoint and scene illumination change simultaneously. Results are best appreciated in the supplementary videos.

(a) insufficient view (b) MPI limits (c) exemplar for (d) (d) missing shadow

Fig. 9. Limitations. Some failure cases include: (a) input photo collections that do not span the full range of desired viewpoints, or (b) intrinsic limitations of MPI leading to poor extrapolation to large camera motions. In addition, as shown in (c) (exemplar image with strong shadow) and (d) (resulting rendering), our method can fail to model strong cast shadows produced by occluders outside the reference field of view.

interpolation. We refer readers to the supplemental videos for animated comparisons.

4D Photos. Figure 8 shows an application of our method to generating animated *4D photos* by animating the 3D viewpoints and simultaneously interpolating between latent appearance features. Our results achieve convincing changes across a variety of times of day and lighting conditions. The parallax effect of our results is best appreciated in the supplemental videos.

User Study. We ran a user study using 24 random sets of videos with camera movements and synthesized images from 5 different scenes. Each video is a sequence of novel views generated by our method, NRW [41], or MUNIT [24]. To quantify the performance of appearance transfer, we also show comparisons of results generated from different exemplar images selected from our test set. We invited 46 participants and asked them to rank the results of the three approaches. 88% of the time, participants responded that the videos produced by our system are the most temporally coherent. 82% of the time, they responded that the results from our method best reproduce the details of structure and illumination one would expect of a real-world scene. 77% of the time, they responded that the results from our method are the most faithful to the corresponding exemplar.

5 Discussion and Conclusion

Limitations. Our method inherits limitations from MPIs. For example, MPIs fail to generalize to viewpoints that are not well-sampled, or that are far from the reference view of the MPI (see Fig. 9(a–b)). In addition, our model can also sometimes fail to model cast shadows from occluders outside of the reference field of view, as shown in Fig. 9(c) and (d). Despite these limitations, we believe our work represents a significant advance towards photo-realistic capture and rendering of the world from crowd photography.

Conclusion. We presented a method for synthesizing novel views of scenes under time-varying appearance from Internet photos. We proposed a new DeepMPI representation and a method for optimizing and decoding DeepMPIs conditioned on viewing conditions present in different photos. Our method can synthesize plenoptic slices that can be interpolated to recover local regions of the full plenoptic function. In the future, we envision enabling even larger changes in viewpoint and illumination, including 4D walkthroughs of large-scale scenes.

Acknowledgements. We thank Kai Zhang, Jin Sun, and Qianqian Wang for helpful discussions. This research was supported in part by the generosity of Eric and Wendy Schmidt by recommendation of the Schmidt Futures program.

References

1. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In: Computational Models of Visual Processing, pp. 3–20. MIT Press (1991)
2. Buehler, C., Bosse, M., McMillan, L., Gortler, S., Cohen, M.: Unstructured lumigraph rendering. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 425–432 (2001)
3. Chai, J.X., Tong, X., Chan, S.C., Shum, H.Y.: Plenoptic sampling. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, pp. 307–318. ACM Press/Addison-Wesley Publishing Co., USA (2000). https://doi.org/10.1145/344779.344932
4. Chaurasia, G., Duchene, S., Sorkine-Hornung, O., Drettakis, G.: Depth synthesis and local warps for plausible image-based navigation. ACM Trans. Graph. **32**(3), 1–12 (2013)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1511–1520 (2017)
7. Chen, Z., et al.: A neural rendering framework for free-viewpoint relighting. arXiv preprint arXiv:1911.11530 (2019)
8. Choi, I., Gallo, O., Troccoli, A., Kim, M.H., Kautz, J.: Extreme view synthesis. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 7781–7790 (2019)
9. Davis, A., Levoy, M., Durand, F.: Unstructured light fields. Comput. Graph. Forum **31**, 305–314 (2012)

10. Debevec, P.E., Taylor, C.J., Malik, J.: 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, pp. 11–20 (1996)
11. Eslami, S.A., et al.: Neural scene representation and rendering. Science **360**(6394), 1204–1210 (2018)
12. Flynn, J., et al.: DeepView: view synthesis with learned gradient descent. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 2367–2376 (2019)
13. Flynn, J., Neulander, I., Philbin, J., Snavely, N.: DeepStereo: learning to predict new views from the world's imagery. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 5515–5524 (2016)
14. Garg, R., Du, H., Seitz, S.M., Snavely, N.: The dimensionality of scene appearance. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1917–1924. IEEE (2009)
15. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proc. Computer Vision and Pattern Recognition (CVPR). pp. 2414–2423 (2016)
16. Goodfellow, I., et al.: Generative adversarial nets. In: Neural Information Processing Systems, pp. 2672–2680 (2014)
17. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 43–54 (1996)
18. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Neural Information Processing Systems, pp. 5767–5777 (2017)
19. Hauagge, D.C., Wehrwein, S., Upchurch, P., Bala, K., Snavely, N.: Reasoning about photo collections using models of outdoor illumination. In: Proceedings of the British Machine Vision Conference (BMVC) (2014)
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 2961–2969 (2017)
21. Hedman, P., Alsisan, S., Szeliski, R., Kopf, J.: Casual 3D photography. ACM Trans. Graph. **36**, 234:1–234:15 (2017)
22. Hedman, P., Philip, J., Price, T., Frahm, J.M., Drettakis, G., Brostow, G.: Deep blending for free-viewpoint image-based rendering. ACM Trans. Graph. **37**(6), 1–15 (2018)
23. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1501–1510 (2017)
24. Huang, X., Liu, M.-Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 179–196. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_11
25. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 1125–1134 (2017)
26. Kalantari, N.K., Wang, T.C., Ramamoorthi, R.: Learning-based view synthesis for light field cameras. ACM Trans. Graph. **35**(6), 1–10 (2016)
27. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)

28. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 4401–4410 (2019)
29. Laffont, P.Y., Bousseau, A., Paris, S., Durand, F., Drettakis, G.: Coherent intrinsic images from photo collections. ACM Trans. Graph. **31**, 202:1–202:11 (2012)
30. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 4681–4690 (2017)
31. Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., Yang, M.-H.: Diverse image-to-image translation via disentangled representations. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11205, pp. 36–52. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01246-5_3
32. Levin, A., Durand, F.: Linear view synthesis using a dimensionality gap light field prior. In: Proceedings Computer Vision and Pattern Recognition (CVPR), pp. 1831–1838 (2010)
33. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 31–42 (1996)
34. Li, Z., et al.: Learning the depths of moving people by watching Frozen people. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 4521–4530 (2019)
35. Li, Z., Snavely, N.: MegaDepth: learning single-view depth prediction from internet photos. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 2041–2050 (2018)
36. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: learning dynamic renderable volumes from images. ACM Trans. Graph. **38**(4), 65 (2019)
37. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 2794–2802 (2017)
38. Martin-Brualla, R., Gallup, D., Seitz, S.M.: 3D time-lapse reconstruction from internet photos. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1332–1340 (2015)
39. Martin-Brualla, R., Gallup, D., Seitz, S.M.: Time-lapse mining from internet photos. ACM Trans. Graph. **34**(4), 1–8 (2015)
40. Matzen, K., Snavely, N.: Scene chronology. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8695, pp. 615–630. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10584-0_40
41. Meshry, M., et al.: Neural rerendering in the wild. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 6871–6880 (2019)
42. Mildenhall, B., et al.: Local light field fusion: practical view synthesis with prescriptive sampling guidelines. ACM Trans. Graph. **38**(4), 1–14 (2019)
43. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 2337–2346 (2019)
44. Penner, E., Zhang, L.: Soft 3D reconstruction for view synthesis. ACM Trans. Graph. **36**(6), 1–11 (2017)
45. Philip, J., Gharbi, M., Zhou, T., Efros, A.A., Drettakis, G.: Multi-view relighting using a geometry-aware network. ACM Trans. Graph. **38**(4), 1–14 (2019)
46. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: controlling deep image synthesis with sketch and color. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 5400–5409 (2017)

47. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 4104–4113 (2016)
48. Shan, Q., Adams, R., Curless, B., Furukawa, Y., Seitz, S.M.: The visual turing test for scene reconstruction. In: International Conference on 3D Vision (3DV), pp. 25–32 (2013)
49. Sheng, L., Lin, Z., Shao, J., Wang, X.: Avatar-net: multi-scale zero-shot style transfer by feature decoration. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2018)
50. Shi, L., Hassanieh, H., Davis, A., Katabi, D., Durand, F.: Light field reconstruction using sparsity in the continuous Fourier domain. ACM Trans. Graph. **34**, 12:1–12:13 (2014)
51. Shi, L., Hassanieh, H., Davis, A., Katabi, D., Durand, F.: Light field reconstruction using sparsity in the continuous Fourier domain. ACM Trans. Graph. **34**(1) (2015). https://doi.org/10.1145/2682631
52. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 1–8. IEEE (2007)
53. Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollhofer, M.: DeepVoxels: learning persistent 3D feature embeddings. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 2437–2446 (2019)
54. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: continuous 3D-structure-aware neural scene representations. In: Neural Information Processing Systems, pp. 1119–1130 (2019)
55. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3D. ACM Trans. Graph. (SIGGRAPH) (2006)
56. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 175–184 (2019)
57. Srinivasan, P.P., Wang, T., Sreelal, A., Ramamoorthi, R., Ng, R.: Learning to synthesize a 4D RGBD light field from a single image. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 2243–2251 (2017)
58. Szeliski, R., Golland, P.: Stereo matching with transparency and matting. Int. J. Comput. Vis. **32**, 45–61 (1998)
59. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: image synthesis using neural textures. ACM Trans. Graph. **38**(4), 1–12 (2019)
60. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: maximizing quality and diversity in feed-forward stylization and texture synthesis. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 6924–6932 (2017)
61. Vagharshakyan, S., Bregovic, R., Gotchev, A.P.: Light field reconstruction using shearlet transform. Trans. Pattern Anal. Mach. Intell. **40**, 133–147 (2015)
62. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 8798–8807 (2018)
63. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 318–335. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_20

64. Xian, W., et al.: TextureGAN: controlling deep image synthesis with texture patches. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 8456–8465 (2018)
65. Xu, Z., Bi, S., Sunkavalli, K., Hadap, S., Su, H., Ramamoorthi, R.: Deep view synthesis from sparse photometric images. ACM Trans. Graph. **38**(4) (2019)
66. Yu, Y., Smith, W.A.: InverseRenderNet: learning single image inverse rendering. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 3155–3164 (2019)
67. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 586–595 (2018)
68. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: learning view synthesis using multiplane images. ACM Trans. Graph. **37**, 1–12 (2018)
69. Zhou, T., Tulsiani, S., Sun, W., Malik, J., Efros, A.A.: View synthesis by appearance flow. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 286–301. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_18
70. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the International Conference on Computer Vision (ICCV), pp. 2223–2232 (2017)
71. Zhu, J.Y., et al.: Toward multimodal image-to-image translation. In: Neural Information Processing Systems, pp. 465–476 (2017)
72. Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S.A.J., Szeliski, R.: High-quality video view interpolation using a layered representation. In: SIGGRAPH 2004 (2004)