

Neural Rerendering in the Wild

Moustafa Meshry^{*1}, Dan B Goldman², Sameh Khamis², Hugues Hoppe², Rohit Pandey²,
Noah Snavely², Ricardo Martin-Brualla²

¹University of Maryland, ²Google Inc.

Abstract

We explore total scene capture — recording, modeling, and rerendering a scene under varying appearance such as season and time of day. Starting from internet photos of a tourist landmark, we apply traditional 3D reconstruction to register the photos and approximate the scene as a point cloud. For each photo, we render the scene points into a deep framebuffer, and train a neural network to learn the mapping of these initial renderings to the actual photos. This rerendering network also takes as input a latent appearance vector and a semantic mask indicating the location of transient objects like pedestrians. The model is evaluated on several datasets of publicly available images spanning a broad range of illumination conditions. We create short videos demonstrating realistic manipulation of the image viewpoint, appearance, and semantic labeling. We also compare results with prior work on scene reconstruction from internet photos.

1. Introduction

Imagine spending a day sightseeing in Rome in a fully realistic interactive experience without ever stepping on a plane. One could visit the Pantheon in the morning, enjoy the sunset overlooking the Colosseum, and fight through the crowds to admire the Trevi Fountain at night time. Realizing this goal involves capturing the complete appearance space of a scene, *i.e.*, recording a scene under all possible lighting conditions and transient states in which the scene might be observed—be it crowded, rainy, snowy, sunrise, spotlight, etc.—and then being able to summon up any viewpoint of the scene under any such condition. We call this ambitious vision *total scene capture*. It is extremely challenging due to the sheer diversity of appearance—scenes can look dramatically different under night illumination, during special events, or in extreme weather.

^{*}Work performed during an internship at Google.

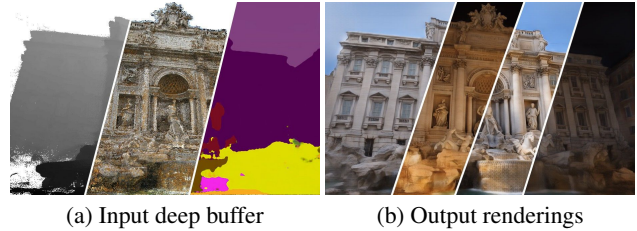


Figure 1: Our neural rerendering technique uses a large-scale internet photo collection to reconstruct a proxy 3D model and trains a neural rerendering network that takes as input a deferred-shading deep buffer (consisting of depth, color and semantic labeling) generated from the proxy 3D model (left), and outputs realistic renderings of the scene under multiple appearances (right).

In this paper, we focus on capturing tourist landmarks around the world using publicly available community photos as the sole input, *i.e.*, photos *in the wild*. Recent advances in 3D reconstruction can generate impressive 3D models from such photo collections [1, 39, 41], but the renderings produced from the resulting point clouds or meshes lack the realism and diversity of real-world images. Alternatively, one could use webcam footage to record a scene at regular intervals but without viewpoint diversity, or use specialized acquisition (*e.g.*, Google Street View, aerial, or satellite images) to snapshot the environment over a short time window but without appearance diversity. In contrast, community photos offer an abundant (but challenging) sampling of appearances of a scene over many years.

Our approach to total scene capture has two main components: (1) creating a factored representation of the input images, which separates viewpoint, appearance conditions, and transient objects such as pedestrians, and (2) rendering realistic images from this factored representation. Unlike recent approaches that extract implicit disentangled representations of viewpoint and content [31, 34, 43], we employ state-of-the-art reconstruction methods to create an explicit intermediate 3D representation, in the form of a dense but noisy point cloud, and use this 3D representation as a “scaffolding” to predict images.

An explicit 3D representation lets us cast the rendering problem as a multimodal image translation [15, 25, 53]. The input is a deferred-shading framebuffer [35] in which each rendered pixel stores albedo, depth, and other attributes, and the outputs are realistic views under different appearances. We train the model by generating paired datasets, using the recovered viewpoint parameters of each input image to render a deep buffer of the scene from the same view, *i.e.*, with pixelwise alignment. Our model effectively learns to take an approximate initial scene rendering and rerender a realistic image. This is similar to recent neural rerendering frameworks [20, 28, 44] but using uncontrolled internet images rather than carefully captured footage.

We explore a novel strategy to train the multimodal image translation model. Rather than jointly estimating an embedding space for the appearance together with the rendering network [15, 25, 53], our system performs staged training of both. First, an appearance encoding network is pretrained using a proxy style-based loss [9], an efficient way to capture the style of an image. Then, the rerendering network is trained with fixed appearance embeddings from the pretrained encoder. Finally, both the appearance encoding and rerendering networks are jointly finetuned. This simple yet effective strategy lets us train simpler networks on large datasets. We demonstrate experimentally how a model trained in this fashion better captures scene appearance.

Our system is a first step towards addressing total scene capture and focuses primarily on the static parts of scenes. Transient objects (*e.g.*, pedestrians and cars) are handled by conditioning the rerendering network on the expected semantic labeling of the output image, so that the network can learn to ignore these objects rather than trying to hallucinate their locations. This semantic labeling is also effective at discarding small or thin scene features (*e.g.*, lampposts) whose geometry cannot be robustly reconstructed, yet are easily identified using image segmentation methods. Conditioning our network on a semantic mask also enables the rendering of scenes free of people if desired. Code will be available at <https://bit.ly/2UzYlWj>.

In summary, our contributions include:

- A first step towards total scene capture, *i.e.*, recording and rerendering a scene under any appearance from in-the-wild photo collections.
- A factorization of input images into viewpoint, appearance, and semantic labeling, conditioned on an approximate 3D scene proxy, from which we can *rerender* realistic views under varying appearance.
- A more effective method to learn the appearance latent space by pretraining the appearance embedding network using a proxy loss.
- Compelling results including view and appearance interpolation on five large datasets, and direct comparisons to previous methods [39].

2. Related work

Scene reconstruction Traditional methods for scene reconstruction first generate a sparse reconstruction using large-scale structure-from-motion [1], then perform Multi-View Stereo (MVS) [7, 38] or variational optimization [13] to reconstruct dense scene models. However, most such techniques assume a single appearance, or else simply recover an average appearance of the scene. We build upon these techniques, using dense point clouds recovered from MVS as proxy geometry for neural rerendering.

In image-based rendering [4, 10], input images are used to generate new viewpoints by warping input pixels into the outputs using proxy geometry. Recently, Hedman *et al.* [12] introduce a neural network to compute blending weights for view-dependent texture mapping that reduces artifacts in poorly reconstructed regions. However, image-based rendering generally assumes the captured scene has static appearance, so it is not well-suited to our problem setup in which the appearance varies across images.

Neural scene rendering [6] applies deep neural networks to learn a latent scene representation that allows generation of novel views, but is limited to simple synthetic geometry.

Appearance modeling A given scene can have dramatically different appearances at different times of day, in different weather conditions, and can also change over the years. Garg *et al.* [8] observe that for a given viewpoint, the dimensionality of scene appearance as captured by internet photos is relatively low, with the exception of outliers like transient objects. One can recover illumination models for a photo collection by estimating albedo using cloudy images [39], retrieving the sun’s location through timestamps and geolocation [11], estimating coherent albedos across the collection [22], or assuming a fixed viewpoint [42]. However, these methods assume simple lighting models that do not apply to nighttime scene appearance. Radenovic *et al.* [33] recover independent day and night reconstructions, but do not enable smooth appearance interpolations between the two.

Laffont *et al.* [23] assign transient attributes like “fall” or “sunny” to each image, and learn a database of patches that allows for editing such attributes. Other works require direct supervision from lighting models estimated using 360-degree images [14], or ground truth object geometry [48]. In contrast, we use a data-driven implicit representation of appearance that is learned from the input image distribution and does not require direct supervision.

Deep image synthesis The seminal work of pix2pix [16] trains a deep neural network to translate an image from one domain, such as a semantic labeling, into another domain, such as a realistic image, using paired training data. Image-to-image (I2I) translation has since been applied to many tasks [5, 24, 32, 49, 47, 50]. Several works propose im-

provements to stabilize training and allow for high-quality image synthesis [18, 46, 47]. Others extend the I2I framework to unpaired settings where images from two domains are not in correspondence [21, 26, 52], multimodal outputs where an input image can map to multiple images [46, 53], or unpaired datasets with multimodal outputs where an image in one domain is converted to another domain while preserving the content [2, 15, 25].

Image translation techniques can be used to re-render scenes in a more realistic domain, to enable facial expression synthesis [20], to fix artifacts in captured 3D performances [28], or to add viewpoint-dependent effects [44]. In our paper, we demonstrate an approach for training a neural rerendering framework *in the wild*, *i.e.*, with uncontrolled data instead of captures under constant lighting conditions. We cast this as a multimodal image synthesis problem, where a given viewpoint can be rendered under multiple appearances using a latent appearance vector, and with editable semantics by conditioning the output on the desired semantic labeling of the output.

3. Total scene capture

We define the problem of *total scene capture* as creating a generative model for all images of a given scene. We would like such a model to:

- encode the 3D structure of the scene, enabling rendering from an arbitrary viewpoint,
- capture all possible appearances of the scene, *e.g.*, all lighting and weather conditions, and allow rendering the scene under any of them, and
- understand the location and appearance of transient objects in the scene, *e.g.*, pedestrians and cars, and allow for reproducing or omitting them.

Although these goals are ambitious, we show that one can create such a generative model given sufficient images of a scene, such as those obtained for popular tourist landmarks.

We first describe a neural rerendering framework that we adapt from previous work in controlled capture settings [28] to the more challenging setting of unstructured photo collections (Section 3.1). We extend this model to enable appearance capture and multimodal generation of renderings under different appearances (Section 3.2). We further extend the model to handle transient objects in the training data by conditioning its inputs on a semantic labeling of the ground truth images (Section 3.3).

3.1. Neural rerendering framework

We adapt recent neural rerendering frameworks [20, 28] to work with unstructured photo collections. Given a large internet photo collection $\{I_i\}$ of a scene, we first generate a proxy 3D reconstruction using COLMAP [36, 37, 38], which applies Structure-from-Motion (SfM) and Multi-View Stereo (MVS) to create a dense colored point cloud.

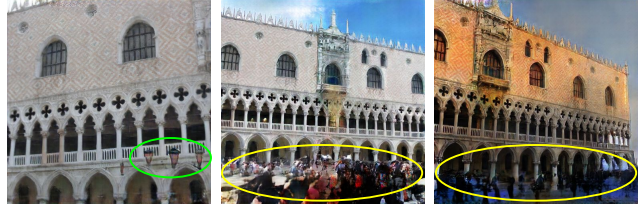


Figure 2: Output frames of a standard image translation network [16] trained for neural rerendering in a small dataset of 250 photos of San Marco. The network overfits the dataset and learns to hallucinate lampposts close to their approximate location in the scene (green), and virtual tourists (yellow), as well as memorizing a per-viewpoint appearance matching the specific input photos.

An alternative to a point cloud is to generate a textured mesh [19, 45]. Although meshes generate more complete renderings, they tend to also contain pieces of misregistered floating geometry which can occlude large regions of the scene [39]. As we show later, our neural rerendering framework can produce highly realistic images given only point-based renderings as input.

Given the proxy 3D reconstruction, we generate an aligned dataset of rendered images and real images by rendering the 3D point cloud from the viewpoint v_i of each input image I_i , where v_i consists of camera intrinsics and extrinsics recovered via SfM. We generate a deferred-shading deep buffer B_i for each image [35], which may contain per-pixel albedo, normal, depth and any other derivative information. In our case, we only use albedo and depth and render the point cloud by using point splatting with a z-buffer with a radius of 1 pixel.

However, the image-to-image translation paradigm used in [20, 28] is not appropriate for our use case, as it assumes a one-to-one mapping between inputs and outputs. A scene observed from a particular viewpoint can look very different depending on weather, lighting conditions, color balance, post processing filters, etc. In addition, a one-to-one mapping fails to explain transient objects in the scene, such as pedestrians or cars, whose location and individual appearance is impossible to predict from the static scene geometry alone. Interestingly, if one trains a sufficiently large neural network on this simple task on a dataset, the network learns to (1) associate viewpoint with appearance via memorization and (2) hallucinate the location of transient objects, as shown in Figure 2.

3.2. Appearance modeling

To capture the one-to-many relationship between input viewpoints (represented by their deep buffers B_i) and output images I_i under different appearances, we cast the rerendering task as multimodal image translation [53]. In such a formulation, the goal is to learn a latent appearance vector z_i^a that captures variations in the output domain I_i that cannot be inferred from the input domain B_i . We compute the latent appearance vector as $z_i^a = E^a(I_i, B_i)$ where

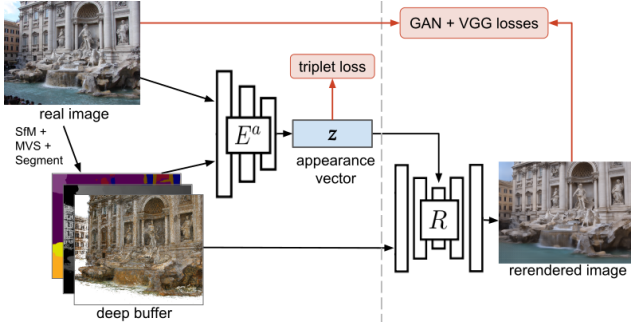


Figure 3: An aligned dataset is created using Structure from Motion (SfM) and Multi-View Stereo (MVS). Our staged approach pre-trains the appearance encoder E^a using a triplet loss (left). Then the renderer R is trained using standard reconstruction and GAN losses (right), and finally fine-tune together with E^a . Photo Credits Rafael Jimenez (Creative Commons).

E^a is an appearance encoder that takes as input both the output image I_i and the deep buffer B_i . We argue that having the appearance encoder E^a observe the input B_i allows it to learn more complex appearance models by correlating the lighting in I_i with scene geometry in B_i . Finally, a rendering network R generates a scene rendering conditioned on both viewpoint B_i and the latent appearance vector z^a . Figure 3 shows an overview of the overall process.

To train the appearance encoder E^a and rendering network R , we first adopted elements from recent methods in multimodal synthesis [15, 25, 53] to find a combination that is most effective in our scenario. However, this combination still has shortcomings as it is unable to model infrequent appearances well. For instance, it does not reliably capture night appearances for scenes in our datasets. We hypothesize that the appearance encoder (which is jointly trained with the rendering network) is not expressive enough to capture the large variability in the data.

To improve the model expressiveness, our approach is to stabilize the joint training of R and E^a by pretraining the appearance network E^a independently on a proxy task. We then employ a staged training approach in which the rendering network R is first trained using fixed appearance embeddings, and finally we jointly fine-tune both networks. This staged training regime allows for a simpler model that captures more complex appearances.

We present our baseline approach, which adapts state-of-the-art multimodal synthesis techniques, and then our staged training strategy, which pretrains the appearance encoder on a proxy task.

Baseline Our baseline uses BicycleGAN [53] with two main adaptations. First, our appearance encoder also takes as input the buffer B_i , as described above. Second, we add a cross-cycle consistency loss similar to [15, 25] to encourage appearance transfer across viewpoints. Let $z_1^a = E^a(I_1, B_1)$ be the captured appearance of an input image I_1 . We apply a reconstruction loss between image I_1 and cross-cycle reconstruction $\hat{I}_1 = R(B_1, z_1^a)$, where \hat{z}_1^a

is computed through a cross-cycle with a second image (I_2, B_2) , i.e. $\hat{z}_1^a = E^a(R(B_2, z_1^a), B_2)$. We also apply a GAN loss on the intermediate appearance transfer output $R(B_2, z_1^a)$ as in [15, 25].

Staged appearance training The key to our staged training approach is the appearance pretraining stage, where we pretrain the appearance encoder E^a independently on a proxy task. We then train the rendering network R while fixing the weights of E^a , allowing R to find the correlations between output images and the embedding produced by the proxy task. Finally, we fine-tune both E^a and R jointly.

This staged approach simplifies and stabilizes the training of R , enabling training of a simpler network with fewer regularization terms. In particular, we remove the cycle and cross-cycle consistency losses, leaving only a direct reconstruction loss, and the KL-divergence loss, leaving only a direct reconstruction loss and a GAN loss. We show experimentally in Section 4 that this approach results in better appearance capture and renderings than the baseline model.

Appearance pretraining To pretrain the appearance encoder E^a , we choose a proxy task that optimizes an embedding of the input images into the appearance latent space using a suitable distance metric between input images. This training encourages embeddings such that if two images are close under the distance metric, then their appearance embeddings should also be close in the appearance latent space. Ideally the distance metric we choose should ignore the content or viewpoint of I_i and B_i , as our goal is to encode a latent space that is independent of viewpoint. Experimentally we find that the style loss employed in neural style-transfer work [9] has such a property; it largely ignores content and focuses on more abstract properties.

To train the embedding, we use a triplet loss, where for each image I_i , we find the set of k closest and furthest neighbor images given by the style loss, from which we can sample a positive sample I_p and negative sample I_n , respectively. The loss is then:

$$\mathcal{L}(I_i, I_p, I_n) = \sum_j \max \left(\|g_i^j - g_p^j\|^2 - \|g_i^j - g_n^j\|^2 + \alpha, 0 \right)$$

where g_i^j is the Gram matrix of activations at the j^{th} layer of a VGG network of image I_i , and α is a separation margin.

3.3. Semantic conditioning

To account for transient objects in the scene, we condition the rendering network on a semantic labeling S_i of image I_i that depicts the location of transient objects such as pedestrians. Specifically, we concatenate the semantic labeling S_i to the deep buffer B_i wherever the deep buffer was previously used. This discourages the network from encoding variations caused by the location of transient objects in the appearance vector, or associating such transient objects with specific viewpoints, as shown in Figure 2.

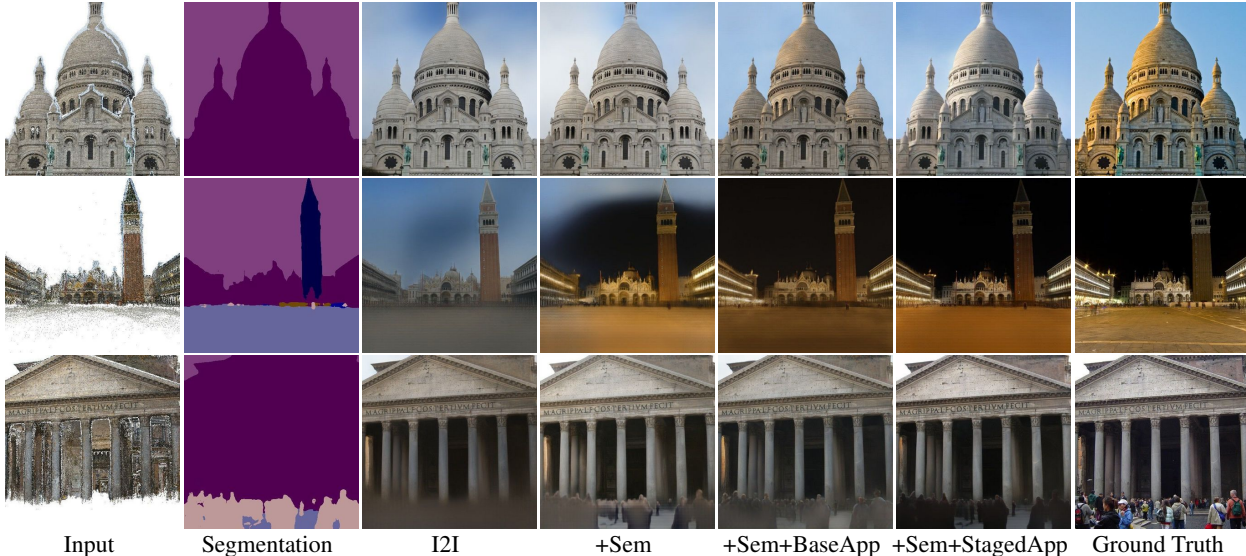


Figure 4: Example visual results of our ablation study in Table 1. From left to right, input color render, segmentation mask from the corresponding ground truth images, result using an image-to-image baseline (I2I), with semantic conditioning (+Sem), and with semantic conditioning and a baseline appearance modeling based on [53] (+Sem+BaseApp), with semantic conditioning and staged appearance training (+Sem+StagedApp). Photo Credits: Flickr users Gary Campbell-Hall, Steve Collis, and Tahbepet (Creative Commons).

A separate benefit of semantic labeling is that it allows the rerendering network to reason about static objects in the scene not captured in the 3D reconstruction, such as lamp-posts in San Marco Square. This prevents the network from haphazardly introducing such objects, and instead lets them appear where they are detected in the semantic labeling, which is a significantly simpler task. In addition, by adding the segmentation labeling to the deep buffer, we allow the appearance encoder to reason about semantic categories like sky or ground when computing the appearance latent vector.

We compute “ground truth” semantic segmentations on the input images I_i using DeepLab [3] trained on ADE20K [51]. ADE20K contains 150 classes, which we map to a 3-channel color image. We find that the quality of the semantic labeling is poor on the landmarks themselves, as they contain unique buildings and features, but is reasonable on transient objects.

Using semantic conditioning, the rerendering network takes as input a semantic labeling of the scene. In order to rerender virtual camera paths, we need to synthesize semantic labelings for each frame in the virtual camera path. To do so, we train a separate semantic labeling network that takes as input the deep buffer B_i , instead of the output image I_i , and estimates a “plausible” semantic labeling \hat{S}_i for that viewpoint given the rendered deep buffer B_i . For simplicity, we train a network with the same architecture as the rendering network (minus the injected appearance vector) on samples (B_i, S_i) from the aligned dataset, and we modify the semantic labelings of the ground truth images S_i and mask out the loss on pixels labeled as transient as defined by a curated list of transient object categories in ADE20K.

4. Evaluation

Here we provide an extensive evaluation of our system. Please also refer to the supplementary video to best appreciate the quality of the results, available in the project website: <https://bit.ly/2UzY1Wj>.

Implementation details Our rerendering network is a symmetric encoder-decoder with skip connections, where the generator is adopted from [18] without using progressive growing. We use a multiscale-patchGAN discriminator [46] with 3 scales and employ a LSGAN [27] loss. As a reconstruction loss, we use the perceptual loss [17] evaluated at $conv_{i,2}$ for $i \in [1, 5]$ of VGG [40]. The appearance encoder architecture is adopted from [25], and we use a latent appearance vector $z^a \in \mathbb{R}^8$. We train on 8 GPUs for ~ 40 epochs using 256×256 crops of input images, but we show compelling results on up to 600×900 at test time. The generator runtime for the staged training network is 330 ms for a 512×512 frame on a TitanV without fp16 optimizations. Architecture and training details can be found in the supplementary material.

Datasets We evaluate our method on five datasets reconstructed with COLMAP [36] from public images, summarized in Table 1. A separate model is trained for each dataset. We create aligned datasets by rendering the reconstructed point clouds with a minimum dimension of 600 pixels, and throw away sparse renderings ($>85\%$ empty pixels), and small images (<450 pixels across). We randomly select a validation set of 100 images per dataset.

Ablative study We perform an ablation study of our system and compare the proposed methods in Figure 4. The

Dataset	#Images	#Points	I2I			+Sem			+Sem+BaseApp			+Sem+StagedApp		
			VGG	L_1	PSNR	VGG	L_1	PSNR	VGG	L_1	PSNR	VGG	L_1	PSNR
Sacre Coeur	1165	33M	70.78	39.98	14.36	66.17	34.78	15.62	60.06	21.58	18.98	61.23	25.22	17.81
Trevi	3006	35M	86.52	42.95	14.14	81.82	36.46	15.57	79.10	28.12	17.37	75.55	25.00	18.19
Pantheon	4972	9M	68.28	39.77	14.50	67.47	36.27	15.13	64.06	28.85	16.76	60.66	23.77	17.95
Dubrovnik	5891	33M	78.42	40.60	14.21	78.58	39.88	14.51	76.61	34.57	15.38	71.65	27.48	17.01
San Marco	7711	7M	80.18	44.04	13.97	78.36	39.34	14.58	70.35	26.24	17.87	68.96	23.11	18.32

Table 1: Dataset statistics (number of registered images and size of reconstructed point cloud) and average error on the validation set using VGG/perceptual loss (lower is better), L_1 loss (lower is better), and PSNR (higher is better), for four methods: an image-to-image baseline (I2I), with semantic conditioning (+Sem), with semantic conditioning and a baseline appearance modeling based on [53] (+Sem+BaseApp), and with semantic conditioning and staged appearance training (+Sem+StagedApp).

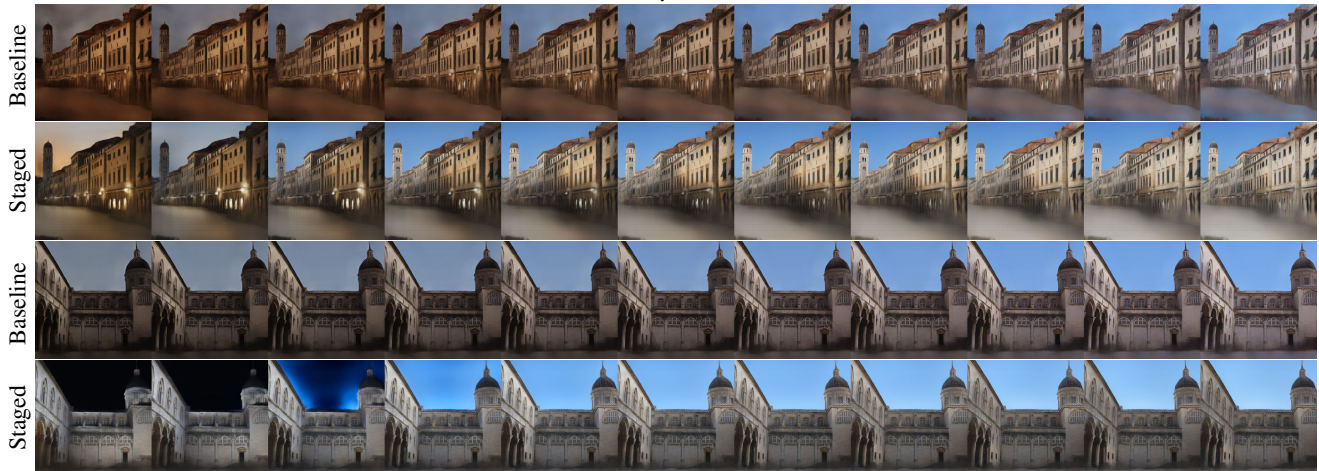


Figure 5: Examples of appearance interpolation for a fixed viewpoint. The left- and rightmost appearances are captured from real images, and the intermediate frames are generated by linearly interpolating the appearances in the latent space. Notice how the baseline method is unable to capture complex scenes, like the sunset and night scene, and its interpolations are rather linear, as can be appreciated in the street lamps (top). The staged training method performs better, but generates twilight artifacts in the sky when interpolating between day and night appearances (bottom).

results of the image-to-image translation baseline method contain additional blurry artifacts near the ground because it hallucinates the locations of pedestrians. Using semantic conditioning, the results improve slightly in those regions. Finally, encoding the appearance of the input photo allows the network to match the appearance. The staged training recovers a closer appearance in San Marco and Pantheon datasets (two bottom rows). However, in Sacre Coeur (top row), the smallest dataset, the baseline appearance model is able to better capture the general appearance of the image, although the staged training model reproduces the directionality of the lighting with more fidelity.

Reconstruction metrics We report image reconstruction errors in the validation set using several metrics: perceptual loss [17], L_1 loss, and PSNR. We use the ground truth semantic mask from the source image, and we extract the appearance latent vector using the appearance encoder. Staged training of the appearance fares better than the baseline for all but the smallest dataset (Sacre Coeur), where the staged training overfits to the training data and is unable to generalize. The baseline method assumes a prior distribution of

the latent space and is less prone to overfitting at the cost of poorer modeling of appearance.

Appearance interpolation The rerendering network allows for interpolating the appearance of two images by interpolating their latent appearance vectors. Figure 5 depicts two examples, showing that the staged training approach is able to generate more complex appearance changes, although its generated interpolations lack realism when transitioning between day and night. In the following, we only show results for the staged training model.

Appearance transfer Figure 6 demonstrates how our full model can transfer the appearance of a given photo to others. It shows realistic renderings of the Trevi fountain from five different viewpoints under four different appearances obtained from other photos. Note the sunny highlights and the spotlight night illumination appearance of the statues. However, these details can flicker when synthesizing a smooth camera path or smoothly interpolating the appearance in the latent space, as seen in the supplementary video.

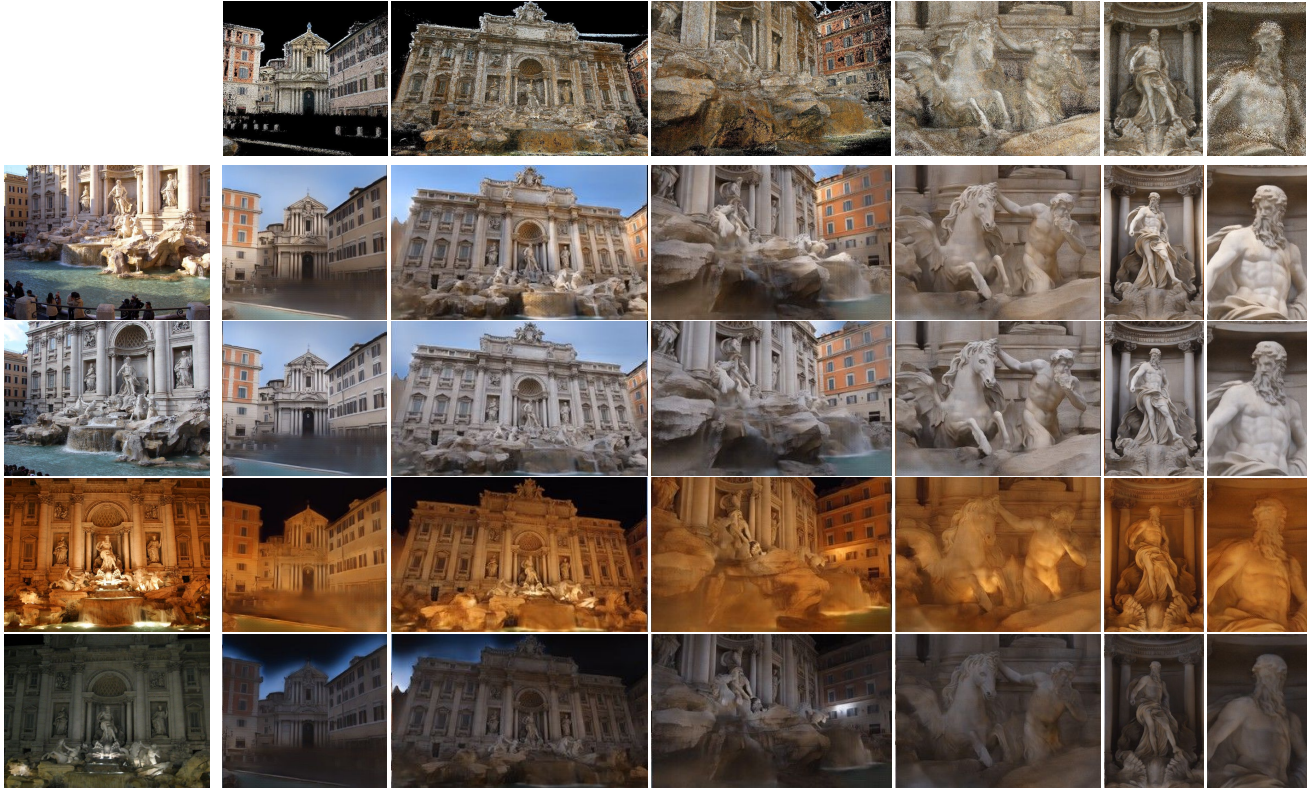


Figure 6: We capture the appearance of the original images in the left column, and re-render several viewpoints under them. The last column is a detail of the previous one. The top row shows the renderings part of the input to the renderer, that exhibit artifacts like incomplete features in the statue, and an inconsistent mix of day and night appearances. Note the hallucinated twilight scene in the sky using the last appearance. Image credits: Flickr users William Warby, Neil Rickards, Rafael Jimenez, acme401 (Creative Commons).

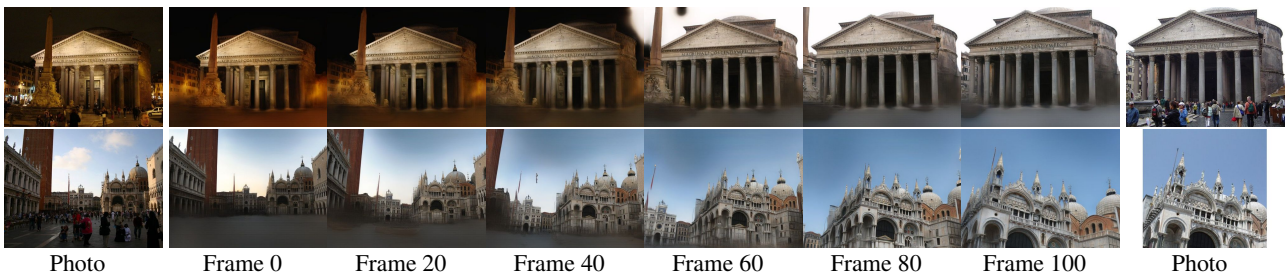


Figure 7: Frames from a synthesized camera path that smoothly transitions from the photo on the left to the photo on the right by smoothly interpolating both viewpoint and the latent appearance vectors. Please see the supplementary video. Photo Credits: Allie Caulfield, Tahbepet, Till Westermayer, Elliott Brown (Creative Commons).

Image interpolation Figure 7 shows sets of two images and frames of smooth image interpolations between them, where both viewpoint and appearance transition smoothly between them. Note how the illumination of the scene can transition smoothly from night to day. The quality of the results is best appreciated in the supplementary video.

Semantic consistency Figure 8 shows the output of the staged training model with ground truth and predicted segmentation masks. Using the predicted masks, the network produces similar results on the building and renders a scene free of people. Note however how the network depicts pedestrians as black, ghostly figures when they appear in the segmentation mask.



(a) w/ GT segmentation (b) w/ predicted segmentations

Figure 8: Example semantic labelings and output renders when using the “ground truth” segmentation mask computed from the corresponding real image (from the validation set) and the predicted one from the associated deep buffer. Note the artifacts on the bottom right where the ground is misclassified as building.

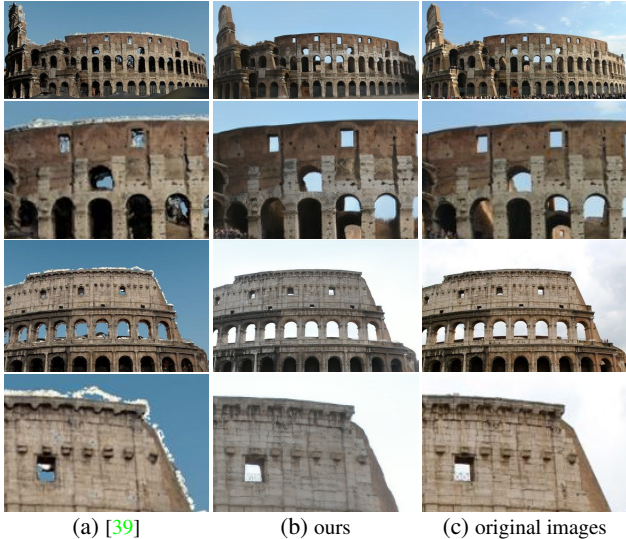


Figure 9: Comparison of [39] and our approach. Rows 1 & 3: original photos. Rows 2 & 4: detailed crops. Image credits: Graeme Churchard, Sarah-Rose (Creative Commons).

Comparison to 3D reconstruction methods We evaluated our technique against the one of Shan *et al.* [39] on the Colosseum, which contains 3K images, 10M color vertices and 48M triangles and was generated from Flickr, Google Street View, and aerial images. Their 3D representation is a dense vertex-colored mesh, where the albedo and vertex normals are jointly recovered together with a simple 8-dimensional lighting model (diffuse, plus directional lighting) for each image in the photo collection.

Figure 9 compares both methods and the original ground truth image. Their method suffers from floating white geometry on the top edge of the Colosseum, and has less detail, although it recovers the lighting better than our method, thanks to its explicit lighting reasoning. Note that both models are accessing the test image to compute lighting coefficients and appearance latent vectors, with dimension 8 in both cases, and that we use the predicted segmentation labels from B_i .

We ran a randomized user study on 20 random sets of output images that do not contain close-ups of people or cars, and were not in our training set. For each viewpoint, 200 participants chose “which image looks most real?” between an output of their system and ours (without seeing the original). Respondents preferred images generated by our system a 69.9% of the time, with our technique being preferred on all but one of the images. We show the 20 random sets of the user study in the supplementary material.

5. Discussion

Our system’s limitations are significantly different from those of traditional 3D reconstruction pipelines:

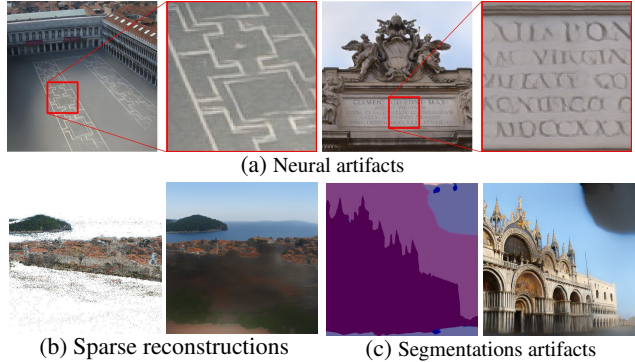


Figure 10: Limitations of the current system.

Segmentation Our model relies heavily on the segmentation mask to synthesize parts of the image not modeled in the proxy geometry, like the ground or sky regions. Thus our results are very sensitive to errors in the segmentation network, like in the sky region in Figure 10c or an appearing “ghost pole” artifact in San Marco (frame 40 of bottom row in Figure 7, best seen in video). Jointly training the neural renderer together with the segmentation network could reduce such artifacts.

Neural artifacts Neural networks are known to produce screendoor patterns [30] and other intriguing artifacts [29]. We observe such artifacts in repeated structures, like the patterns on the floor of San Marco, which in our renderings are misaligned as if hand-painted. Similarly, the inscription above the Trevi fountain is reproduced with a distorted font (see Figure 10a).

Incomplete reconstructions Sometimes an image contains partially reconstructed parts of the 3D model, creating large holes in the rendered B_i . This forces the network to hallucinate the incomplete regions, generally leading to blurry outputs (see Figure 10b).

Temporal artifacts When smoothly varying the viewpoint, sometimes the appearance of the scene can flicker considerably, especially under complex appearance, such as when the sun hits the Trevi Fountain, creating complex highlights and cast shadows. Please see the supplementary video for an example.

In summary, we present a first attempt at solving the total scene capture problem. Using unstructured internet photos, we can train a neural rendering network that is able to produce highly realistic scenes under different illumination conditions. We propose a novel staged training approach that better captures the appearance of the scene as seen in internet photos. Finally, we evaluate our system on five challenging datasets and against state-of-the-art 3D reconstruction methods.

Acknowledgements: We thank Gregory Blascovich for his help in conducting the user study, and Johannes Schönberger and True Price for their help generating datasets.

A. Supplementary Results

Appearance variation. Figure 12 shows additional results of diverse appearances modeled by our proposed staged training method on the San Marco dataset. As in Figure 6, it shows realistic renderings of five different scenes/viewpoints under four different appearances obtained from other photos.

Qualitative comparison. We evaluate our technique against Shan *et al.* [39] on the Colosseum. In Section 4, we report the result of a user study run on 20 randomly selected sets of output images that do not contain close-ups of people or cars, and were not in our training set. Figures 13, 14 show a side-by-side comparison of all 20 images used in the user study.

Quantitative evaluation with learned segmentations

To quantitatively evaluate rerendering using estimated segmentation masks, we generate semantic labelings for the validation set, as described in Section 3.3, and recompute the quantitative metrics, as in Table 1, for our proposed method. Note that estimated semantic maps will not perfectly match those of the ground truth validation images. For example, ground truth semantic maps could contain the segmentation of transient objects, like people or trees. So, it is not fair to compare reconstructions based on estimated segmentation maps to the ground truth validation images. While results in Table 2 show some performance drop as expected, we still get a reasonable performance compared to that in Table 1. In fact, we still perform better than the BicycleGAN baseline on the Trevi, Pantheon and Dubrovnik datasets, even though the BicycleGAN baseline uses ground truth segmentation masks.

Dataset	+Sem+StagedApp		
	VGG	L_1	PSNR
Sacre Coeur	67.74	28.66	16.45
Trevi	77.35	26.03	17.90
Pantheon	62.54	25.40	17.29
Dubrovnik	74.44	30.39	16.18
San Marco	75.58	26.69	17.34

Table 2: We evaluate our staged training approach using estimated segmentation masks, as opposed to Table 1 which uses segmentation masks computed from ground truth validation images.

B. Implementation Details

We use different networks for the staged training and the baseline mode. We obtain best results for each model with different networks. Below, we provide an overview of the different architectures used in the staged training and the baseline models. Code will be available at <https://bit.ly/2Uzy1Wj>.



Figure 11: Sample frames of the aligned dataset. Even though interior structures can be seen through the walls in the point cloud rendering (bottom), neural rerendering is able to reason about occlusion among the points and thereby avoid rerendering artifacts. Image credits: James Manners, Patrick Denker (Creative Commons).

B.1. Neural rerender network architecture

Our rerendering network is a symmetric encoder-decoder with skip connections. The generator is adopted from [18] without using progressive growing. Specifically, we extend the GAN architecture in [18] to a conditional GAN setting. The encoder/decoder operates at a 256×256 resolution, with 6 downsampling/upsampling blocks. Each block has a downsampling/upsampling layer followed by two single-strided 3×3 conv layers with a *leaky ReLu* ($\alpha = 0.2$) and *pixel-norm* [18] layers. We add skip connections between the encoder and decoder by concatenating feature maps at the beginning of each decoder block. We use 64 feature maps at the first encoder and double the size of feature maps after each downsampling layer until it reaches size 512.

B.2. Appearance encoder architecture

We implement the appearance encoder architecture used in [25] except that we add *pixel-norm* [18] layers after each downsampling block. We observe that adding a pixel-wise normalization layer stabilizes the training while at the same time avoids mixing information between different pixels as in *instance norm* or *batch norm*. We use a latent appearance vector $z^a \in \mathbb{R}^8$. The latent vector is injected at the bottleneck between the encoder and decoder in the rendering network. We tile z^a to match the dimension of feature maps at the bottleneck and concatenate it to the feature maps channel-wise.

B.3. Baseline architecture

We use a faithful Tensorflow implementation of the encoder-decoder network and appearance encoder in [25] using their PyTorch released code as a guideline. We adapt their training pipeline to the single-domain supervised setup as described in Section 3.2 in our paper.

B.4. Aligned datasets

Figure 11 shows sample frames from aligned datasets we generate as described in Section 3.1.

B.5. Latent space visualization

Figure 15 visualizes the latent space learned by the appearance encoder, E^a , after appearance pretraining and finetuning in our staged training, as well as training E^a with the BicycleGAN baseline. The embedding learned during the appearance pretraining stage shows meaningful clusters, but has lower quality than the one learned after finetuning, which is comparable to the one of the BicycleGAN baseline.



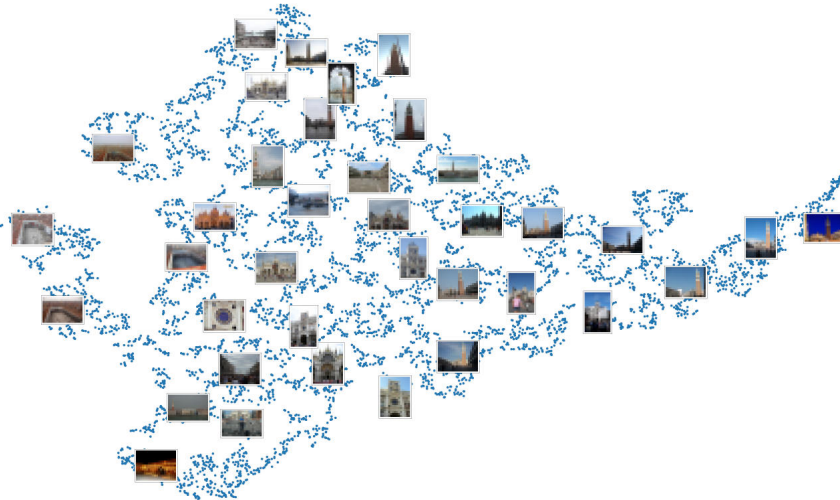
Figure 12: We capture the appearance of the original images in the first row, and re-render several viewpoints under them. The first column shows the rendered point cloud images used as input to the re-renderer. Image credits: Michael Pate, Jeremy Thompson, Patrick Denker, Rob Young (Creative Commons).



Figure 13: Comparison with Shan *et al.* [39] – set 1 of 2. First and third columns show the result of Shan *et al.* [39]. Second and fourth columns show our result.



Figure 14: Comparison with Shan *et al.* [39] – set 2 of 2. First and third columns show the result of Shan *et al.* [39]. Second and fourth columns show our result.



(a) Our staged training: After appearance pretraining.



(b) Our staged training: After finetuning.



(c) BicycleGAN baseline.

Figure 15: t-SNE plots for the latent appearance space learned by the appearance encoder (a) after appearance pretraining in our staged training, (b) after finetuning in our staged training, and (c) using the BicycleGAN baseline.

References

- [1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV*, 2009. 1, 2
- [2] A. Almahairi, S. Rajeshwar, A. Sordoni, P. Bachman, and A. Courville. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In *ICML*, 2018. 3
- [3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. PAMI*, 2018. 5
- [4] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH*, 1996. 2
- [5] H. Dong, S. Yu, C. Wu, and Y. Guo. Semantic image synthesis via adversarial learning. In *ICCV*, 2017. 2
- [6] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, et al. Neural scene representation and rendering. *Science*, 2018. 2
- [7] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. PAMI*, 2010. 2
- [8] R. Garg, H. Du, S. M. Seitz, and N. Snavely. The dimensionality of scene appearance. In *ICCV*, 2009. 2
- [9] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2, 4
- [10] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. SIGGRAPH*, 1996. 2
- [11] D. Hauagge, S. Wehrwein, P. Upchurch, K. Bala, and N. Snavely. Reasoning about photo collections using models of outdoor illumination. In *BMVC*, 2014. 2
- [12] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow. Deep blending for free-viewpoint image-based rendering. In *Proc. SIGGRAPH*, 2018. 2
- [13] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *CVPR*, 2009. 2
- [14] Y. Hold-Geoffroy, K. Sunkavalli, S. Hadap, E. Gambaretto, and J.-F. Lalonde. Deep outdoor illumination estimation. In *CVPR*, 2017. 2
- [15] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2, 3, 4
- [16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 3
- [17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5, 6
- [18] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 3, 5, 9
- [19] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. Eurographics Symposium on Geometry Processing*, 2006. 3
- [20] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt. Deep video portraits. In *Proc. SIGGRAPH*, 2018. 2, 3
- [21] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 2017. 3
- [22] P.-Y. Laffont, A. Bousseau, S. Paris, F. Durand, and G. Drettakis. Coherent intrinsic images from photo collections. In *Proc. SIGGRAPH Asia*, 2012. 2
- [23] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. In *Proc. SIGGRAPH*, 2014. 2
- [24] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017. 2
- [25] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018. 2, 3, 4, 5, 9
- [26] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017. 3
- [27] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 5
- [28] R. Martin-Brualla, R. Pandey, S. Yang, P. Pidlypenskyi, J. Taylor, J. Valentin, S. Khamsi, P. Davidson, A. Tkach, P. Lincoln, A. Kowdle, C. Rhemann, D. B. Goldman, C. Keskin, S. Seitz, S. Izadi, and S. Fanello. LookinGood: Enhancing performance capture with real-time neural re-rendering. In *Proc. SIGGRAPH Asia*, 2018. 2, 3
- [29] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*, 2015. 8
- [30] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 8
- [31] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg. Transformation-grounded image generation network for novel 3D view synthesis. In *CVPR*, 2017. 1
- [32] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2
- [33] F. Radenović, J. L. Schönberger, D. Ji, J.-M. Frahm, O. Chum, and J. Matas. From dusk till dawn: Modeling in the dark. In *CVPR*, 2016. 2
- [34] H. Rhodin, M. Salzmann, and P. Fua. Unsupervised geometry-aware representation learning for 3D human pose estimation. In *ECCV*, 2018. 1
- [35] T. Saito and T. Takahashi. Comprehensible rendering of 3-D shapes. In *Proc. SIGGRAPH*, 1990. 2, 3
- [36] J. L. Schönberger. Colmap. <http://colmap.github.io>, 2016. 3, 5
- [37] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 3
- [38] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2, 3

- [39] Q. Shan, R. Adams, B. Curless, Y. Furukawa, and S. M. Seitz. The Visual Turing Test for scene reconstruction. In *Proc. 3DV*, 2013. [1](#), [2](#), [3](#), [8](#), [9](#), [12](#), [13](#)
- [40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. [5](#)
- [41] N. Snavely, S. M. Seitz, and R. Szeliski. Photo Tourism: Exploring photo collections in 3D. In *Proc. SIGGRAPH*, 2006. [1](#)
- [42] K. Sunkavalli, W. Matusik, H. Pfister, and S. Rusinkiewicz. Factored time-lapse video. In *Proc. SIGGRAPH*, 2007. [2](#)
- [43] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Multi-view 3D models from single images with a convolutional network. In *ECCV*, 2016. [1](#)
- [44] J. Thies, M. Zollhöfer, C. Theobalt, M. Stamminger, and M. Nießner. IGNOR: Image-guided neural object rendering. *arXiv 2018*, 2018. [2](#), [3](#)
- [45] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! Large-scale texturing of 3D reconstructions. In *ECCV*, 2014. [3](#)
- [46] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. [3](#), [5](#)
- [47] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, N. Yakovenko, A. Tao, J. Kautz, and B. Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. [2](#), [3](#)
- [48] T. Y. Wang, T. Ritschel, and N. J. Mitra. Joint material and illumination estimation from photo sets in the wild. In *Proc. 3DV*, 2018. [2](#)
- [49] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *ECCV*, 2016. [2](#)
- [50] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR*, 2017. [2](#)
- [51] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. [5](#)
- [52] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. [3](#)
- [53] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. [2](#), [3](#), [4](#), [5](#), [6](#)