DOI: 10.1111/cgf.14022
EUROGRAPHICS 2020
R. Mantiuk and V. Sundstedt
(Guest Editors)

Volume 39 (2020), Number 2
STAR – State of The Art Report

State of the Art on Neural Rendering

A. Tewari[1⋆] O. Fried[2⋆] J. Thies[3⋆] V. Sitzmann[2⋆] S. Lombardi[4] K. Sunkavalli[5] R. Martin-Brualla[6] T. Simon[4] J. Saragih[4] M. Nießner[3] R. Pandey[6] S. Fanello[6] G. Wetzstein[2] J.-Y. Zhu[5] C. Theobalt[1] M. Agrawala[2] E. Shechtman[5] D. B Goldman[6] M. Zollhöfer[4]

[1]MPI Informatics [2]Stanford University [3]Technical University of Munich [4]Facebook Reality Labs [5]Adobe Research [6]Google Inc ⋆Equal contribution.

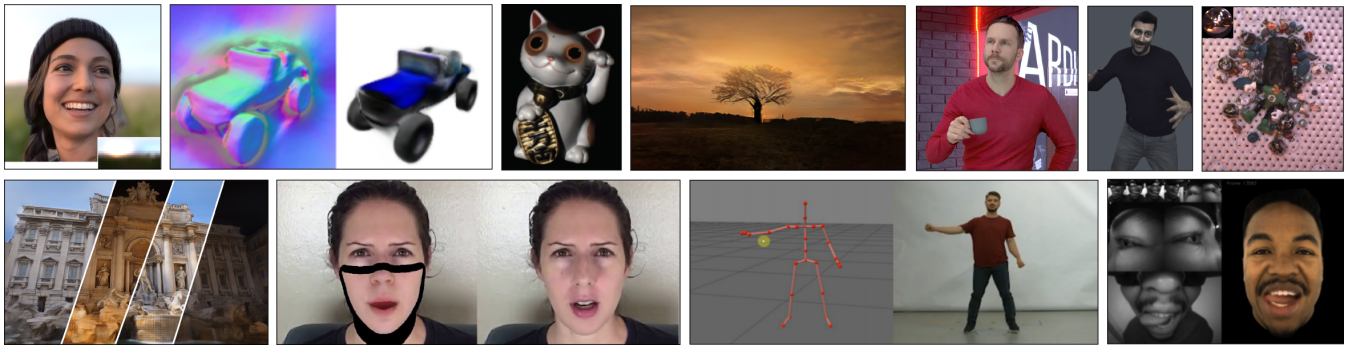


Figure 1: *Neural renderings of a large variety of scenes. See Section 6 for more details on the various methods. Images from [SBT*19, SZW19, XBS*19, KHM17, GLD*19, MBPY*18, XSHR18, MGK*19, FTZ*19, LXZ*19, WSS*19].*

Abstract

Efficient rendering of photo-realistic virtual worlds is a long standing effort of computer graphics. Modern graphics techniques have succeeded in synthesizing photo-realistic images from hand-crafted scene representations. However, the automatic generation of shape, materials, lighting, and other aspects of scenes remains a challenging problem that, if solved, would make photo-realistic computer graphics more widely accessible. Concurrently, progress in computer vision and machine learning have given rise to a new approach to image synthesis and editing, namely deep generative models. Neural rendering is a new and rapidly emerging field that combines generative machine learning techniques with physical knowledge from computer graphics, e.g., by the integration of differentiable rendering into network training. With a plethora of applications in computer graphics and vision, neural rendering is poised to become a new area in the graphics community, yet no survey of this emerging field exists. This state-of-the-art report summarizes the recent trends and applications of neural rendering. We focus on approaches that combine classic computer graphics techniques with deep generative models to obtain controllable and photo-realistic outputs. Starting with an overview of the underlying computer graphics and machine learning concepts, we discuss critical aspects of neural rendering approaches. Specifically, our emphasis is on the type of control, i.e., how the control is provided, which parts of the pipeline are learned, explicit vs. implicit control, generalization, and stochastic vs. deterministic synthesis. The second half of this state-of-the-art report is focused on the many important use cases for the described algorithms such as novel view synthesis, semantic photo manipulation, facial and body reenactment, relighting, free-viewpoint video, and the creation of photo-realistic avatars for virtual and augmented reality telepresence. Finally, we conclude with a discussion of the social implications of such technology and investigate open research problems.

1. Introduction

The creation of photo-realistic imagery of virtual worlds has been one of the primary driving forces for the development of sophisticated computer graphics techniques. Computer graphics approaches span the range from real-time rendering, which enables the latest generation of computer games, to sophisticated global illumination simulation for the creation of photo-realistic digital humans in feature films. In both cases, one of the main bottlenecks is content creation, i.e., that a vast amount of tedious and expensive manual work of skilled artists is required for the creation of the underlying scene representations in terms of surface geometry, appearance/material, light sources, and animations. Concurrently, powerful generative models have emerged in the computer vision and machine learning communities. The seminal work on *Generative Adversarial Neural Networks* (GANs) by Goodfellow et al. [GPAM*14] has evolved in recent years into

deep generative models for the creation of high resolution imagery [RMC16, KALL17, BDS19] and videos [VPT16, CDS19]. Here, control over the synthesized content can be achieved by conditioning [IZZE17, ZPIE17] the networks on control parameters or images from other domains. Very recently, the two areas have come together and have been explored as "neural rendering". One of the first publications that used the term neural rendering is *Generative Query Network* (GQN) [ERB*18]. It enables machines to learn to perceive their surroundings based on a representation and generation network. The authors argue that the network has an implicit notion of 3D due to the fact that it could take a varying number of images of the scene as input, and output arbitrary views with correct occlusion. Instead of an implicit notion of 3D, a variety of other methods followed that include this notion of 3D more explicitly, exploiting components of the graphics pipeline.

While classical computer graphics starts from the perspective of physics, by modeling for example geometry, surface properties and cameras, machine learning comes from a statistical perspective, i.e., learning from real world examples to generate new images. To this end, the quality of computer graphics generated imagery relies on the physical correctness of the employed models, while the quality of the machine learning approaches mostly relies on carefully-designed machine learning models and the quality of the used training data. Explicit reconstruction of scene properties is hard and error prone and leads to artifacts in the rendered content. To this end, image-based rendering methods try to overcome these issues, by using simple heuristics to combine captured imagery. But in complex scenery, these methods show artifacts like seams or ghosting. Neural rendering brings the promise of addressing both *reconstruction and rendering* by using deep networks to learn complex mappings from captured images to novel images. Neural rendering combines physical knowledge, e.g., mathematical models of projection, with learned components to yield new and powerful algorithms for controllable image generation. Neural rendering has not yet a clear definition in the literature. Here, we define *Neural Rendering* as:

> *Deep image or video generation approaches that enable explicit or implicit control of scene properties such as illumination, camera parameters, pose, geometry, appearance, and semantic structure.*

This state-of-the-art report defines and classifies the different types of neural rendering approaches. Our discussion focuses on methods that combine computer graphics and learning-based primitives to yield new and powerful algorithms for *controllable* image generation, since controllability in the image generation process is essential for many computer graphics applications. One central scheme around which we structure this report is the kind of control afforded by each approach. We start by discussing the fundamental concepts of computer graphics, vision, and machine learning that are prerequisites for neural rendering. Afterwards, we discuss critical aspects of neural rendering approaches, such as: type of control, how the control is provided, which parts of the pipeline are learned, explicit vs. implicit control, generalization, and stochastic vs. deterministic synthesis. Following, we discuss the landscape of applications that is enabled by neural rendering. The applications of neural rendering range from novel view synthesis, semantic photo manipulation, facial and body reenactment, relighting, free-viewpoint video, to the creation of photo-realistic avatars for virtual and augmented reality telepresence Since the creation and manipulation of images that are indistinguishable from real photos has many social implications, especially when humans are photographed, we also discuss these implications and the detectability of synthetic content. As the field of neural rendering is still rapidly evolving, we conclude with current open research problems.

2. Related Surveys and Course Notes

Deep Generative Models have been widely studied in the literature, with several surveys [Sal15, OE18, Ou18] and course notes [ope, Sta, IJC] describing them. Several reports focus on specific generative models, such as *Generative Adversarial Networks* (GANs) [WSW19, CWD*18, Goo16, CVPa, PYY*19] and *Variational Autoencoders* (VAEs) [Doe16, KW19]. Controllable image synthesis using classic computer graphics and vision techniques have also been studied extensively. Image-based rendering has been discussed in several survey reports [SK00, ZC04]. The book of Szeliski [Sze10] gives an excellent introduction to 3D reconstruction and image-based rendering techniques. Recent survey reports [EST*19, ZTG*18] discuss approaches for 3D reconstruction and controllable rendering of faces for various applications. Some aspects of neural rendering have been covered in tutorials and workshops of recent computer vision conferences. These include approaches for free viewpoint rendering and relighting of full body performances [ECCa, CVPb, CVPc], tutorials on neural rendering for face synthesis [ECCb] and 3D scene generation using neural networks [CVPd]. However, none of the above surveys and courses provide a structured and comprehensive look into neural rendering and all of its various applications.

3. Scope of this STAR

In this state-of-the-art report, we focus on novel approaches that combine classical computer graphics pipelines and learnable components. Specifically, we are discussing where and how classical rendering pipelines can be improved by machine learning and which data is required for training. To give a comprehensive overview, we also give a short introduction to the pertinent fundamentals of both fields, i.e., computer graphics and machine learning. The benefits of the current hybrids are shown, as well as their limitations. This report also discusses novel applications that are empowered by these techniques. We focus on techniques with the primary goal of generating controllable photo-realistic imagery via machine learning. We do not cover work on geometric and 3D deep learning [MON*19, SHN*19, QSMG16, CXG*16, PFS*19], which is more focused on 3D reconstruction and scene understanding. This branch of work is highly inspiring for many neural rendering approaches, especially ones that are based on 3D-structured scene representations, but goes beyond the scope of this survey. We are also not focused on techniques that employ machine learning for denoising raytraced imagery [CKS*17, KBS15].

4. Theoretical Fundamentals

In the following, we discuss theoretical fundamentals of work in the neural rendering space. First, we discuss image formation models

in computer graphics, followed by classic image synthesis methods. Next, we discuss approaches to generative models in deep learning.

4.1. Physical Image Formation

Classical computer graphics methods approximate the physical process of image formation in the real world: light sources emit photons that interact with the objects in the scene, as a function of their geometry and material properties, before being recorded by a camera. This process is known as light transport. Camera optics acquire and focus incoming light from an aperture onto a sensor or film plane inside the camera body. The sensor or film records the amount of incident light on that plane, sometimes in a nonlinear fashion. All the components of image formation—light sources, material properties, and camera sensors—are wavelength-dependent. Real films and sensors often record only one to three different wavelength distributions, tuned to the sensitivity of the human visual system. All the steps of this physical image formation are modelled in computer graphics: light sources, scene geometry, material properties, light transport, optics, and sensor behavior.

4.1.1. Scene Representations

To model objects in a scene, many different representations for scene geometry have been proposed. They can be classified into *explicit* and *implicit* representations. Explicit methods describe scenes as a collection of geometric primitives, such as triangles, point-like primitives, or higher-order parametric surfaces. Implicit representations include signed distance functions mapping from $\mathbb{R}^3 \rightarrow \mathbb{R}$, such that the surface is defined as the zero-crossing of the function (or any other level-set). In practice, most hardware and software renderers are tuned to work best on triangle meshes, and will convert other representations into triangles for rendering.

The interactions of light with scene surfaces depend on the material properties of the surfaces. Materials may be represented as bidirectional reflectance distribution functions (BRDFs) or bidirectional subsurface scattering reflectance distribution functions (BSSRDFs). A BRDF is a 5-dimensional function that describes how much light of a given wavelength incident on a surface point from each incoming ray direction is reflected toward each exiting ray direction. While a BRDF only models light interactions that happen at a single surface point, a BSSDRF models how light incident on one surface point is reflected at a different surface point, thus making it a 7-D function. BRDFs can be represented using analytical models [Pho75, CT82, ON95] or measured data [MPBM03]. When a BRDF changes across a surface, it is referred to as a spatially-varying BRDF (svBRDF). Spatially varying behavior across geometry may be represented by binding discrete materials to different geometric primitives, or via the use of texture mapping. A texture map defines a set of continuous values of a material parameter, such as diffuse albedo, from a 2- or 3-dimensional domain onto a surface. 3-dimensional textures represent the value throughout a bounded region of space and can be applied to either explicit or implicit geometry. 2-dimensional textures map from a 2-dimensional domain onto a parametric surface; thus, they are typically applicable only to explicit geometry.

Sources of light in a scene can be represented using parametric models; these include point or directional lights, or area sources that are represented by surfaces in the scene that emit light. Some methods account for continuously varying emission over a surface, defined by a texture map or function. Often environment maps are used to represent dense, distant scene lighting. These environment maps can be stored as non-parametric textures on a sphere or cube, or can be approximated by coefficients of a spherical harmonic basis [Mül66]. Any of the parameters of a scene might be modeled as varying over time, allowing both animation across successive frames, and simulations of motion blur within a single frame.

4.1.2. Camera Models

The most common camera model in computer graphics is the pinhole camera model, in which rays of light pass through a pinhole and hit a film plane (image plane). Such a camera can be parameterized by the pinhole's 3D location, the image plane, and a rectangular region in that plane representing the spatial extent of the sensor or film. The operation of such a camera can be represented compactly using projective geometry, which converts 3D geometric representations using homogeneous coordinates into the two-dimensional domain of the image plane. This is also known as a full perspective projection model. Approximations of this model such as the weak perspective projection are often used in computer vision to reduce complexity because of the non-linearity of the full perspective projection. More accurate projection models in computer graphics take into account the effects of non-ideal lenses, including distortion, aberration, vignetting, defocus blur, and even the inter-reflections between lens elements [SRT*11].

4.1.3. Classical Rendering

The process of transforming a scene definition including cameras, lights, surface geometry and material into a simulated camera image is known as rendering. The two most common approaches to rendering are rasterization and raytracing: *Rasterization* is a feed-forward process in which geometry is transformed into the image domain, sometimes in back-to-front order known as painter's algorithm. *Raytracing* is a process in which rays are cast backwards from the image pixels into a virtual scene, and reflections and refractions are simulated by recursively casting new rays from the intersections with the geometry [Whi80].

Hardware-accelerated rendering typically relies on rasterization, because it has good memory coherence. However, many real-world image effects such as global illumination and other forms of complex light transport, depth of field, motion blur, etc. are more easily simulated using raytracing, and recent GPUs now feature acceleration structures to enable certain uses of raytracing in real-time graphics pipelines (e.g., NVIDIA RTX or DirectX Raytracing [HAM19]). Although rasterization requires an explicit geometric representation, raytracing/raycasting can also be applied to implicit representations. In practice, implicit representations can also be converted to explicit forms for rasterization using the marching cubes algorithm [LC87] and other similar methods. Renderers can also use combinations of rasterization and raycasting to obtain high efficiency and physical realism at the same time (e.g., screen space ray-tracing [MM14]). The quality of images produced by a given rendering pipeline depends heavily on the accuracy of the different models in the pipeline. The components

must account for the discrete nature of computer simulation, such as the gaps between pixel centers, using careful application of sampling and signal reconstruction theory. The process of estimating the different model parameters (camera, geometry, material, light parameters) from real-world data, for the purpose of generating novel views, editing materials or illumination, or creating new animations is known as *inverse rendering*. Inverse rendering [Mar98, DAD*19, HMR19, DAD*18, LSC18], which has been explored in the context of both computer vision and computer graphics, is closely related to neural rendering. A drawback of inverse rendering is that the predefined physical model or data structures used in classical rendering don't always accurately reproduce all the features of real-world physical processes, due to either mathematical complexity or computational expense. In contrast, neural rendering introduces learned components into the rendering pipeline in place of such models. Deep neural nets can statistically approximate such physical processes, resulting in outputs that more closely match the training data, reproducing some real-world effects more accurately than inverse rendering.

Note that there are approaches at the intersection of inverse rendering and neural rendering. E.g., Li et al. [LXR*18] uses a neural renderer that approximates global illumination effects to efficiently train an inverse rendering method that predicts depth, normal, albedo and roughness maps. There are also approaches that use neural networks to enhance specific building blocks of the classical rendering pipeline, e.g., shaders. Rainer et al. [RJGW19] learn Bidirectional Texture Functions and Maximov et al. [MLTFR19] learn Appearance Maps.

4.1.4. Light Transport

Light transport considers all the possible paths of light from the emitting light sources, through a scene, and onto a camera. A well-known formulation of this problem is the classical rendering equation [Kaj86]:

$$L_o(\mathbf{p}, \omega_o, \lambda, t) = L_e(\mathbf{p}, \omega_o, \lambda, t) + L_r(\mathbf{p}, \omega_o, \lambda, t)$$

where L_o represents outgoing radiance from a surface as a function of location, ray direction, wavelength, and time. The term L_e represents direct surface emission, and the term L_r represents the interaction of incident light with surface reflectance:

$$L_r(\mathbf{p}, \omega_o, \lambda, t) = \int_\Omega f_r(\mathbf{p}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{p}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) \, \mathrm{d}\omega_i$$

Note that this formulation omits consideration of transparent objects and any effects of subsurface or volumetric scattering. The rendering equation is an integral equation, and cannot be solved in closed form for nontrivial scenes, because the incident radiance L_i appearing on the right hand side is the same as the outgoing radiance L_o from another surface on the same ray. Therefore, a vast number of approximations have been developed. The most accurate approximations employ *Monte Carlo* simulations [Vea98], sampling ray paths through a scene. Faster approximations might expand the right hand side one or two times and then truncate the recurrence, thereby simulating only a few "bounces" of light. Computer graphics artists may also simulate additional bounces by adding non-physically based light sources to the scene.

4.1.5. Image-based Rendering

In contrast to classical rendering, which projects 3D content to the 2D plane, image-based rendering techniques generate novel images by transforming an existing set of images, typically by warping and compositing them together. Image-based rendering can handle animation, as shown by Thies et al. [TZS*18], but the most common use-case is novel view synthesis of static objects, in which image content from captured views are warped into a novel view based on a proxy geometry and estimated camera poses [DYB98, GGSC96, HRDB16]. To generate a complete new image, multiple captured views have to be warped into the target view, requiring a blending stage. The resulting image quality depends on the quality of the geometry, the number and arrangement of input views, and the material properties of the scene, since some materials change appearance dramatically across viewpoints. Although heuristic methods for blending and the correction of view-dependent effects [HRDB16] show good results, recent research has substituted parts of these image-based rendering pipelines with learned components. Deep neural networks have successfully been employed to reduce both blending artifacts [HPP*18] and artifacts that stem from view-dependent effects [TZT*20] (Section 6.2.1).

4.2. Deep Generative Models

While traditional computer graphics methods focus on physically modeling scenes and simulating light transport to generate images, machine learning can be employed to tackle this problem from a statistical standpoint, by learning the distribution of real world imagery. Compared to classical image-based rendering, which historically has used small sets of images (e.g., hundreds), deep generative models can learn image priors from large-scale image collections.

Seminal work on deep generative models [AHS85, HS06, SH09] learned to generate random samples of simple digits and frontal faces. In these early results, both the quality and resolution was far from that achievable using physically-based rendering techniques. However, more recently, photo-realistic image synthesis has been demonstrated using *Generative Adversarial Networks* (GANs) [GPAM*14] and its extensions. Recent work can synthesize random high-resolution portraits that are often indistinguishable from real faces [KLA19].

Deep generative models excel at generating *random* realistic images with statistics resembling the training set. However, user control and interactivity play a key role in image synthesis and manipulation [BSFG09]. For example, concept artists want to create particular scenes that reflect their design ideas rather than random scenes. Therefore, for computer graphics applications, generative models need to be extended to a conditional setting to gain explicit control of the image synthesis process. Early work trained feed-forward neural networks with a per-pixel ℓ_p distance to generate images given conditional inputs [DTSB15]. However, the generated results are often blurry as ℓ_p distance in pixel space considers each pixel independently and ignores the complexity of visual structure [IZZE17, BM18]. Besides, it tends to average multiple possible outputs. To address the above issue, recent work proposes perceptual similarity distances [GEB16, DB16, JAFF16] to measure the discrepancy between synthesized results and ground

truth outputs in a high-level deep feature embedding space constructed by a pre-trained network. Applications include artistic stylization [GEB16, JAFF16], image generation and synthesis [DB16, CK17], and super-resolution [JAFF16, LTH*17]. Matching an output to its ground truth image does not guarantee that the output looks natural [BM18]. Instead of minimizing the distance between outputs and targets, *conditional GANs* (cGANs) aim to match the conditional distribution of outputs given inputs [MO14, IZZE17]. The results may not look the same as the ground truth images, but they look natural. Conditional GANs have been employed to bridge the gap between coarse computer graphics renderings and the corresponding real-world images [BSP*19b, ZPIE17], or to produce a realistic image given a user-specified semantic layout [IZZE17, PLWZ19b]. Below we provide more technical details for both network architectures and learning objectives.

4.2.1. Learning a Generator

We aim to learn a neural network G that can map a conditional input $x \in \mathcal{X}$ to an output $y \in \mathcal{Y}$. Here $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output domains. We call this neural network *generator*. The conditional input x can take on a variety of forms depending on the targeted application, such as a user-provided sketch image, camera parameters, lighting conditions, scene attributes, textual descriptions, among others. The output y can also vary, from an image, a video, to 3D data such as voxels or meshes. See Table 1 for a complete list of possible network inputs and outputs for each application.

Here we describe three commonly-used generator architectures. Readers are encouraged to check application-specific details in Section 6. (1) *Fully Convolutional Networks (FCNs)* [MBLD92, LSD15] are a family of models that can take an input image with arbitrary size and predict an output with the same size. Compared to popular image classification networks such as AlexNet [KSH12] and VGG [SZ15] that map an image into a vector, FCNs use fractionally-strided convolutions to preserve the spatial image resolution [ZKTF10]. Although originally designed for recognition tasks such as semantic segmentation and object detection, FCNs have been widely used for many image synthesis tasks. (2) *U-Net* [RFB15] is an FCN-based architecture with improved localization ability. The model adds so called "skip connections" from high-resolution feature maps at early layers to upsampled features in late-stage layers. These skip connections help to produce more detailed outputs, since high-frequency information from the input can be passed directly to the output. (3) *ResNet-based generators* use residual blocks [HZRS16] to pass the high-frequency information from input to output, and have been used in style transfer [JAFF16] and image super-resolution [LTH*17].

4.2.2. Learning using Perceptual Distance

Once we collect many input-output pairs and choose a generator architecture, how can we learn a generator to produce a *desired* output given an input? What would be an effective objective function for this learning problem? One straightforward way is to cast it as a regression problem, and to minimize the distance between the output $G(x)$ and its ground truth image y, as follows:

$$\mathcal{L}_{\text{recon}}(G) = \mathbb{E}_{x,y}||G(x) - y||_p, \tag{1}$$

where $\mathbb{E}$ denotes the expectation of the loss function over training pairs (x, y), and $||\cdot||_p$ denotes the p-norm. Common choices include ℓ_1- or ℓ_2-loss. Unfortunately, the learned generator tends to synthesize blurry images or average results over multiple plausible outputs. For example, in image colorization, the learned generator sometimes produces desaturated results due to the averaging effect [ZIE16]. In image super-resolution, the generator fails to synthesize structures and details as the p-norm looks at each pixel independently [JAFF16].

To design a learning objective that better aligns with human's perception of image similarity, recent work [GEB16, JAFF16, DB16] proposes measuring the distance between deep feature representations extracted by a pre-trained image classifier F (e.g., VGG network [SZ15]). Such a loss is advantageous over the ℓ_p-norm, as the deep representation summarizes an entire image holistically, while the ℓ_p-norm evaluates the quality of each pixel independently. Mathematically, a generator is trained to minimize the following feature matching objective.

$$\mathcal{L}_{\text{perc}}(G) = \mathbb{E}_{x,y} \sum_{t=1}^{T} \lambda_t \frac{1}{N_t} \left|\left|F^{(t)}(G(x)) - F^{(t)}(y)\right|\right|_1, \tag{2}$$

where $F^{(t)}$ denotes the feature extractor in the t-th layer of the pre-trained network F with T layers in total and N_t denoting the total number of features in layer t. The hyper-parameter λ_t denotes the weight for each layer. Though the above distance is often coined "perceptual distance", it is intriguing why matching statistics in multi-level deep feature space can match human's perception and help synthesize higher-quality results, as the networks were originally trained for image classification tasks rather than image synthesis tasks. A recent study [ZIE*18] suggests that rich features learned by strong classifiers also provide useful representations for human perceptual tasks, outperforming classic hand-crafted perceptual metrics [WBSS04, WSB03]

4.2.3. Learning with Conditional GANs

However, minimizing distances between output and ground truth does not guarantee realistic looking output, according to the work of Blau and Michaeli [BM18]. They also prove that the small distance and photorealism are at odds with each other. Therefore, instead of distance minimization, deep generative models focus on distribution matching, i.e., matching the distribution of generated results to the distribution of training data. Among many types of generative models, *Generative Adversarial Networks* (GANs) have shown promising results for many computer graphics tasks. In the original work of Goodfellow et al. [GPAM*14], a GAN generator $G : z \rightarrow y$ learns a mapping from a low-dimensional random vector z to an output image y. Typically, the input vector is sampled from a multivariate Gaussian or Uniform distribution. The generator G is trained to produce outputs that cannot be distinguished from "real" images by an adversarially trained discriminator, D. The discriminator is trained to detect synthetic images generated by the generator. While GANs trained for object categories like faces or vehicles learn to synthesize high-quality instances of the object, usually the synthesized background is of a lower quality [KLA19, KALL17]. Recent papers [SDM19, AW19] try to alleviate this problem by learning generative models of a complete scene.

To add conditional information as input, *conditional GANs* (cGANs) [MO14, IZZE17] learn a mapping $G : \{x,z\} \rightarrow y$ from an observed input x and a randomly sampled vector z to an output image y. The observed input x is also passed to the discriminator, which models whether image pairs $\{x,y\}$ are real or fake. As mentioned before, both input x and output y vary according to the targeted application. In class-conditional GANs [MO14], the input x is a categorical label that controls which object category a model should generate. In the case of image-conditional GANs such as *pix2pix* [IZZE17], the generator G aims to translate an input image x, for example a semantic label map, to a realistic-looking output image, while the discriminator D aims to distinguish real images from generated ones. The model is trained with paired dataset $\{x_i,y_i\}_{i=1}^N$ that consists of pairs of corresponding input x_i and output images y_i. cGANs match the conditional distribution of the output given an input via the following minimax game:

$$\min_G \max_D \mathcal{L}_{\text{cGAN}}(G,D) \quad . \tag{3}$$

Here, the objective function $\mathcal{L}_{cGAN}(G,D)$ is normally defined as:

$$\mathcal{L}_{\text{cGAN}}(G,D) = \mathbb{E}_{x,y}\big[\log D(x,y)\big] + \mathbb{E}_{x,z}\big[\log\big(1 - D(x,G(x,z))\big)\big]. \tag{4}$$

In early cGAN implementations [IZZE17, ZPIE17], no noise vector is injected, and the mapping is deterministic, as it tends to be ignored by the network during training. More recent work uses latent vectors z to enable multimodal image synthesis [ZZP*17, HLBK18, ARS*18]. To stabilize training, cGANs-based methods [IZZE17, WLZ*18b] also adopt per-pixel ℓ_1-loss $\mathcal{L}_{\text{recon}}(G)$ (Equation (1)) and perceptual distance loss $\mathcal{L}_{\text{perc}(G)}$ (Equation (2)).

During training, the discriminator D tries to improve its ability to tell *real* and *synthetic* images apart, while the generator G, at the same time, tries to improve its capability of fooling the discriminator. The pix2pix method adopts a U-Net [RFB15] as the architecture of the generator and a patch-based fully convolutional network (FCN) [LSD15] as the discriminator.

Conceptually, perceptual distance and Conditional GANs are related, as both of them use an auxiliary network (either F or D) to define an effective learning objective for learning a better generator G. In a high-level abstraction, an accurate computer vision model (F or D) for assessing the quality of synthesized results $G(x)$ can significantly help tackle neural rendering problems. However, there are two significant differences. First, perceptual distance aims to measure the discrepancy between an output instance and its ground truth, while conditional GANs measure the closeness of the conditional distributions of real and fake images. Second, for perceptual distance, the feature extractor F is pre-trained and fixed, while conditional GANs adapt its discriminator D on the fly according to the generator. In practice, the two methods are complementary, and many neural rendering applications use both losses simultaneously [WLZ*18b, SZUL18]. Besides GANs, many promising research directions have recently emerged including Variational Autoencoders (VAEs) [KW13], auto-regressive networks (e.g., PixelCNN [OKV*16], PixelRNN [OKK16, ODZ*16]), invertible density models [DSDB17, KD18], among others. StarGAN [CCK*18] enables training a single model for image-to-image translation based on multiple datasets with different domains. To keep the discussion concise, we focus on GANs here. We urge our readers to review tutorials [Doe16, KW19] and course notes [ope, Sta, IJC] for a complete picture of deep generative models.

4.2.4. Learning without Paired Data

Learning a generator with the above objectives requires hundreds to millions of paired training data. In many real-world applications, paired training data are difficult and expensive to collect. Different from labeling images for classification tasks, annotators have to label every single pixel for image synthesis tasks. For example, only a couple of small datasets exist for tasks like semantic segmentation. Obtaining input-output pairs for graphics tasks such as artistic stylization can be even more challenging since the desired output often requires artistic authoring and is sometimes not even well-defined. In this setting, the model is given a source set $\{x_i\}_{i=1}^N$ ($x_i \in \mathcal{X}$) and a target set $\{y_j\}_{j=1}$ ($y_j \in \mathcal{Y}$). All we know is which target *domain* the output $G(x)$ should come from: i.e., like an image from domain Y. But given a particular input, we do not know which target *image* the output should be. There could be infinitely many mappings to project an image from $\mathcal{X}$ to $\mathcal{Y}$. Thus, we need additional constraints. Several constraints have been proposed including cycle-consistency loss for enforcing a bijective mapping [ZPIE17, YZTG17, KCK*17], the distance preserving loss for encouraging that the output is close to the input image either in pixel space [SPT*17] or in feature embedding space [BSD*17, TPW17], the weight sharing strategy for learning shared representation across domains [LT16, LBK17, HLBK18], etc. The above methods broaden the application scope of conditional GANs and enable many graphics applications such as object transfiguration, domain transfer, and CG2real.

5. Neural Rendering

Given high-quality scene specifications, classic rendering methods can render photorealistic images for a variety of complex real-world phenomena. Moreover, rendering gives us explicit editing control over all the elements of the scene—camera viewpoint, lighting, geometry and materials. However, building high-quality scene models, especially directly from images, requires significant manual effort, and automated scene modeling from images is an open research problem. On the other hand, deep generative networks are now starting to produce visually compelling images and videos either from random noise, or conditioned on certain user specifications like scene segmentation and layout. However, they do not yet allow for fine-grained control over scene appearance and cannot always handle the complex, non-local, 3D interactions between scene properties. In contrast, neural rendering methods hold the promise of combining these approaches to enable controllable, high-quality synthesis of novel images from input images/videos. Neural rendering techniques are diverse, differing in the control they provide over scene appearance, the inputs they require, the outputs they produce, and the network structures they utilize. A typical neural rendering approach takes as input images corresponding to certain scene conditions (for example, viewpoint, lighting, layout, etc.), builds a "neural" scene representation from them, and "renders" this representation under novel scene properties to synthesize novel images. The learned scene representation is not restricted by simple scene modeling approximations and can be optimized for high quality

novel images. At the same time, neural rendering approaches incorporate ideas from classical graphics—in the form of input features, scene representations, and network architectures—to make the learning task easier, and the output more controllable.

We propose a taxonomy of neural rendering approaches along the axes that we consider the most important:

- *Control:* What do we want to control and how do we condition the rendering on the control signal?
- *CG Modules:* Which computer graphics modules are used and how are they integrated into a neural rendering pipeline?
- *Explicit or Implicit Control:* Does the method give explicit control over the parameters or is it done implicitly by showing an example of what we expect to get as output?
- *Multi-modal Synthesis:* Is the method trained to output multiple optional outputs, given a specific input?
- *Generality:* Is the rendering approach generalized over multiple scenes/objects?

In the following, we discuss these axes that we use to classify current state-of-the-art methods (see also Table 1).

5.1. Control

Neural rendering aims to render high-quality images under user-specified scene conditions. In the general case, this is an open research problem. Instead, current methods tackle specific sub-problems like novel view synthesis [HPP*18, TZT*20, STH*19, SZW19], relighting under novel lighting [XSHR18, GLD*19], and animating faces [KGT*18, TZN19, FTZ*19] and bodies [ASL*19, SZA*19, MBPY*18] under novel expressions and poses. A main axis in which these approaches differ is in how the control signal is provided to the network. One strategy is to directly pass the scene parameters as input to the first or an intermediate network layer [ERB*18]. Related strategies are to tile the scene parameters across all pixels of an input image, or concatenate them to the activations of an inner network layer [MHP*19, SBT*19]. Another approach is to rely on the spatial structure of images and employ an image-to-image translation network to map from a "guide image" or "conditioning image" to the output. For example, such approaches might learn to map from a semantic mask to the output image [KAEE16, PLWZ19b, WLZ*18b, ZKSE16, BSP*19a, BLRW17, CK17, IZZE17]. Another option, which we describe in the following, is to use the control parameters as input to a graphics layer.

5.2. Computer Graphics Modules

One emerging trend in neural rendering is the integration of computer graphics knowledge into the network design. Therefore, approaches might differ in the level of "classical" graphics knowledge that is embedded in the system. For example, directly mapping from the scene parameters to the output image does not make use of any graphics knowledge. One simple way to integrate graphics knowledge is a non-differentiable computer graphics module. Such a module can for example be used to render an image of the scene and pass it as dense conditioning input to the network [KGT*18, LXZ*19, FTZ*19, MBPY*18]. Many different channels could be provided as network inputs, such as a depth map, normal map, camera/world space position maps, albedo map, a diffuse rendering of the scene, and many more. This transforms the problem to an image-to-image translation task, which is a well researched setting, that can for example be tackled by a deep conditional generative model with skip connections. A deeper integration of graphics knowledge into the network is possible based on differentiable graphics modules. Such a differentiable module can for example implement a complete computer graphics renderer [LSS*19, SZW19], a 3D rotation [STH*19, NPLBY18, NLT*19], or an illumination model [SYH*17]. Such components add a physically inspired inductive bias to the network, while still allowing for end-to-end training via backpropagation. This can be used to analytically enforce a truth about the world in the network structure, frees up network capacity, and leads to better generalization, especially if only limited training data is available.

5.3. Explicit vs. Implicit Control

Another way to categorize neural rendering approaches is by the type of control. Some approaches allow for explicit control, i.e., a user can edit the scene parameters manually in a semantically meaningful manner. For example, current neural rendering approaches allow for explicit control over camera viewpoint [XBS*19, TZT*20, NLT*19, ERB*18, HPP*18, AUL19, MGK*19, NPLBY18, SZW19, STH*19], scene illumination [ZHSJ19, XSHR18, PGZ*19, MHP*19, SBT*19], facial pose and expression [LSSS18, TZN19, WSS*19, KGT*18, GSZ*18]. Other approaches only allow for implicit control by way of a representative sample. While they can copy the scene parameters from a reference image/video, one cannot manipulate these parameters explicitly. This includes methods that transfer human head motion from a reference video to a target person [ZSBL19], or methods which retarget full-body motion [ASL*19, CGZE18] Methods which allow for explicit control require training datasets with images/videos and their corresponding scene parameters.On the other hand, implicit control usually requires less supervision. These methods can be trained without explicit 3D scene parameters, only with weaker annotations. For example, while dense facial performance capture is required to train networks with explicit control for facial reenactment [KGT*18, TZN19], implicit control can be achieved by training just on videos with corresponding sparse 2D keypoints [ZSBL19].

5.4. Multi-modal Synthesis

Often times it is beneficial to have several different output options to choose from. For example, when only a subset of scene parameters are controlled, there potentially exists a large multi-modal output space with respect to the other scene parameters. Instead of being presented with one single output, the user can be presented with a gallery of several choices, which are visibly different from each other. Such a gallery helps the user better understand the output landscape and pick a result to their liking. To achieve various outputs which are significantly different from each other the network or control signals must have some stochasticity or structured variance. For example, variational auto-encoders [KW13, LSLW16] model processes with built-in variability, and can be used to achieve multi-modal synthesis [WDGH16, XWBF16, ZZP*17]. The latest example is Park et al. [PLWZ19b], which demonstrates one way to incorporate variability and surfaces it via a user interface: given the

same semantic map, strikingly different images are generated with the push of a button.

5.5. Generality

Neural rendering methods differ in their object specificity. Some methods aim to train a general purpose model once, and apply it to all instances of the task at hand [XBS*19, SZW19, NPLBY18, NLT*19, HPP*18, ERB*18, BSP*19a, PLWZ19b, ZKSE16, BLRW17, ZSBL19, IZZE17, KAEE16, CK17, WLZ*18b]. For example, if the method operates on human heads, it will aim to be applicable to all humans. Conversely, other methods are instance-specific [CGZE18, LXZ*19, LSSS18, WSS*19, ASL*19, STH*19, LSS*19, KGT*18, FTZ*19, TZT*20, AUL19, MGK*19, SZW19]. Continuing our human head example, these networks will operate on a single person (with a specific set of clothes, in a specific location) and a new network will have to be retrained for each new subject. For many tasks, object specific approaches are currently producing higher quality results, at the cost of lengthy training times for each object instance. For real-world applications such training times are prohibitive—improving general models is an open problem and an exciting research direction.

6. Applications of Neural Rendering

Neural rendering has many important use cases such as semantic photo manipulation, novel view synthesis, relighting, free-viewpoint video, as well as facial and body reenactment. Table 1 provides an overview of various applications discussed in this survey. For each, we report the following attributes:

- ***Required Data.*** All the data that is required for the system. This does not include derived data, e.g., automatically computed facial landmarks, but instead can be thought of as the minimal amount of data a person would have to acquire in order to be able to reproduce the system.
- ***Network Inputs.*** The data that is directly fed into the learned part of the system, i.e., the part of the system through which the gradients flow during backpropagation.
- ***Network Outputs.*** Everything produced by the learned parts of the system. This is the last part of the pipeline in which supervision is provided.

  Possible values for *Required Data*, *Network Inputs* and *Network Outputs*: Images, Videos, Meshes, Noise, Text, Camera, Lighting, 2D Joint positions, Renders, Semantic labels, 2D Keypoints, volumE, teXtures, Depth (for images or video).

- ***Contents.*** The types of objects and environments that the system is designed to handle as input and output. Possible values: Head, Person, Room, outdoor Environment, Single object (of any category).
- ***Controllable Parameters.*** The parameters of the scene that can be modified. Possible values: Camera, Pose, Lighting, coloR, Texture, Semantics, Expression, speecH.
- ***Explicit control.*** Refers to systems in which the user is given interpretable parameters that, when changed, influence the generated output in a predictable way. Possible values: ✗ uninterpretable or uncontrollable, ✓ interpretable controllable parameters.
- ***CG module.*** The level of "classical" graphics knowledge embedded in the system. Possible values: ✗ no CG module, Non-differentiable CG module, Differentiable CG module.

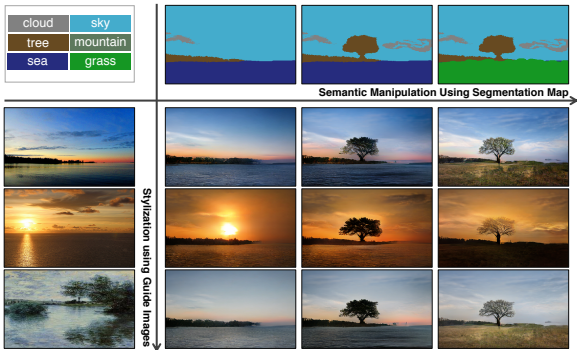


Figure 2: GauGAN *[PLWZ19b, PLWZ19a] enables image synthesis with both semantic and style control. Please see the SIGGRAPH 2019 Real-Time Live for more details. Images taken from Park et al. [PLWZ19b].*

- ***Generality.*** General systems are trained once and can be applied to multiple different input instances. E.g. a system that synthesizes humans, but has to be retrained for each new person, does not have this property. Possible values: ✗ instance specific, ✓ general.
- ***Multi-modal synthesis.*** Systems that, as presented, allow on-demand generation of multiple outputs which are significantly different from each other, based on the same input. Possible values: ✗ single output, ✓ on-demand multiple outputs.
- ***Temporal coherence.*** Specifies whether temporal coherence is explicitly enforced during training of the approach. Possible values: ✗ not enforced, ✓ enforced (e.g. in loss function).

The following is a detailed discussion of various neural rendering applications.

6.1. Semantic Photo Synthesis and Manipulation

Semantic photo synthesis and manipulation enable interactive image editing tools for controlling and modifying the appearance of a photograph in a semantically meaningful way. The seminal work *Image Analogies* [HJO*01] creates new texture given a semantic layout and a reference image, using patch-based texture synthesis [EL99, EF01]. Such single-image patch-based methods [HJO*01, WSI07, SCSI08, BSFG09] enable image reshuffling, retargeting, and inpainting, but they cannot allow high-level operations such as adding a new object or synthesizing an image from scratch. Data-driven graphics systems create new imagery by compositing multiple image regions [PGB03] from images retrieved from a large-scale photo collection [LHE*07, CCT*09, JBS*06, HE07, MEA09]. These methods allow the user to specify a desired scene layout using inputs such as a sketch [CCT*09] or a semantic label map [JBS*06]. The latest development is *OpenShapes* [BSR19], which composes regions by matching scene context, shapes, and parts. While achieving appealing results, these systems are often slow as they search in a large image database. In addition, undesired artifacts can be sometimes spotted due to visual inconsistency between different images.

6.1.1. Semantic Photo Synthesis

In contrast to previous non-parametric approaches, recent work has trained fully convolutional networks [LSD15] with a condi-

Method	Required Data	Network Inputs	Network Outputs	Contents	Controllable Parameters	Explicit Control	CG Module	Generality	Multi-modal Synthesis	Temporal Coherence	
Bau et al. [BSP*19a]	IS	IS	I	RE	S	✗	✗	✓	✗	✗	Semantic Photo Synthesis (Section 6.1)
Brock et al. [BLRW17]	I	N	I	S	R	✓	✗	✓	✗	✗	
Chen and Koltun [CK17]	IS	S	I	RE	S	✗	✗	✓	✓	✗	
Isola et al. [IZZE17]	IS	S	I	ES	S	✗	✗	✓	✗	✗	
Karacan et al. [KAEE16]	IS	S	I	E	S	✗	✗	✓	✓	✗	
Park et al. [PLWZ19b]	IS	S	I	RE	S	✗	✗	✓	✓	✗	
Wang et al. [WLZ*18b]	IS	S	I	RES	S	✗	✗	✓	✓	✗	
Zhu et al. [ZKSE16]	I	N	I	ES	RT	✓	✗	✓	✓	✗	
Aliev et al. [AUL19]	ID	R	I	RS	C	✓	N	✗	✗	✗	Novel View Synthesis (Section 6.2)
Eslami et al. [ERB*18]	IC	IC	I	RS	C	✓	✗	✓	✗	✗	
Hedman et al. [HPP*18]	V	I	I	RES	C	✓	N	✓	✗	✗	
Meshry et al. [MGK*19]	I	IL	I	RE	CL	✓	N	✗	mark	✗	
Nguyen-Phuoc et al. [NPLBY18]	ICL	E	I	S	CL	✓	N	✓	✗	✗	
Nguyen-Phuoc et al. [NLT*19]	I	NC	I	S	C	✓	✗	✓	✓	✗	
Sitzmann et al. [STH*19]	V	IC	I	S	C	✓	D	✗	✗	✗	
Sitzmann et al. [SZW19]	IC	IC	I	S	C	✓	D	✓	✗	✗	
Thies et al. [TZT*20]	V	IRC	I	S	C	✓	N	✗	✗	✗	
Xu et al. [XBS*19]	IC	IC	I	S	C	✓	✗	✓	✗	✗	
Lombardi et al. [LSS*19]	VC	IC	I	HPS	C	✓	D	✗	✗	✗	Free Viewpoint Video (Section 6.3)
Martin-Brualla et al. [MBPY*18]	VDC	R	V	P	C	✓	N	✓	✗	✓	
Pandey et al. [PTY*19]	VDI	IDC	I	P	C	✓	✗	✓	✗	✗	
Shysheya et al. [SZA*19]	V	R	I	P	CP	✓	✗	✓	✗	✗	
Meka et al. [MHP*19]	IL	IL	I	H	L	✓	✗	✓	✗	✗	Relighting (Section 6.4)
Philip et al. [PGZ*19]	I	IL	I	E	L	✓	N	✓	✗	✗	
Sun et al. [SBT*19]	IL	IL	IL	H	L	✓	✗	✓	✗	✗	
Xu et al. [XSHR18]	IL	IL	I	S	L	✓	✗	✓	✗	✗	
Zhou et al. [ZHSJ19]	IL	IL	IL	H	L	✓	✗	✓	✗	✗	
Fried et al. [FTZ*19]	VT	VR	V	H	H	✓	N	✗	✗	✓	Facial Reenactment (Section 6.5)
Kim et al. [KGT*18]	V	R	V	H	PE	✓	N	✗	✗	✓	
Lombardi et al. [LSSS18]	VC	IMC	MX	H	CP	✓	N	✗	✗	✗	
Thies et al. [TZN19]	V	IRC	I	HS	CE	✓	D	✗	✗	✗	
Wei et al. [WSS*19]	VC	I	MX	H	CP	✓	D	✗	✗	✗	
Zakharov et al. [ZSBL19]	I	IK	I	H	PE	✗	✗	✓	✗	✗	
Aberman et al. [ASL*19]	V	J	V	P	P	✗	✗	✗	✗	✓	Body Reenactment (Section 6.5)
Chan et al. [CGZE18]	V	J	V	P	P	✗	✗	✗	✗	✓	
Liu et al. [LXZ*19]	VM	R	V	P	P	✓	N	✗	✗	✓	
	Inputs and Outputs				Control		Misc				

Table 1: *Selected methods presented in this survey. See Section 6 for explanation of attributes in the table and their possible values.*

tional GANs objective [MO14, IZZE17] to directly map a user-specified semantic layout to a photo-realistic image [IZZE17, KAEE16, LBK17, ZPIE17, YZTG17, HLBK18, WLZ*18b]. Other types of user inputs such as color, sketch, and texture have also been supported [SLF*17, ISSI16, ZIE16, XSA*18]. Among these, *pix2pix* [IZZE17] and the method of Karacan et al. [KAEE16] present the first learning-based methods for semantic image synthesis including generating street view and natural scene images. To increase the image resolution, *Cascaded refinement networks* [CK17] learn a coarse-to-fine generator, trained with a perceptual loss [GEB16]. The results are high-resolution, but lack high frequency texture and details. To synthesize richer details, *pix2pixHD* [WLZ*18b] proposes a conditional GAN that can generate 2048×1024 results with realistic texture. The key extensions compared to *pix2pix* [IZZE17] include a coarse-to-fine generator similar to CRN [CK17], multi-scale discriminators that capture local image statistics at different scales, and a multi-scale discriminator-based feature matching objective, that resembles perceptual distance [GEB16], but uses an adaptive discriminator to extract task-specific features instead. Notably, the multi-scale pipeline, a decades-old scheme in vision and graphics [BA83, BL*03], is still highly effective for deep image synthesis. Both *pix2pixHD* and *BicycleGAN* [ZZP*17] can synthesize multiple possible outputs given the same user input, allowing a user to choose different styles. Subsequent systems [WLZ*18a, BMRS18, BUS18] extend to the video domain, allowing a user to control the semantics of a video. Semi-parametric systems [QCJK18, BSR19] combine classic data-driven image compositing [LHE*07] and feed-forward networks [LSD15].

Most recently, *GauGAN* [PLWZ19b, PLWZ19a] uses a SPatially-Adaptive (DE)normalization layer (SPADE) to better preserve semantic information in the generator. While previous conditional models [IZZE17, WLZ*18b] process a semantic layout through multiple normalization layers (e.g., InstanceNorm [UVL16]), the channel-wise normalization layers tend to "wash away" semantic information, especially for uniform and flat input layout regions. Instead, the GauGAN generator takes a random latent vector as an image style code, and employs multiple ResNet blocks with spatially-adaptive normalization layers (SPADE), to produce the final output. As shown in Figure 2, this design not only produces visually appealing results, but also enables better user control over style and semantics. The adaptive normalization layers have also been found to be effective for stylization [HB17] and super-resolution [WYDCL18].

6.1.2. Semantic Image Manipulation

The above image synthesis systems excel at creating new visual content, given user controls as inputs. However, *semantic image manipulation* of a user provided image with deep generative models remains challenging for two reasons. First, editing an input image requires accurately reconstructing it with the generator, which is a difficult task even with recent GANs. Second, once the controls are applied, the newly synthesized content might not be compatible with the input photo. To address these issues, *iGAN* [ZKSE16] proposes using an unconditional GAN as a natural image prior for image editing tasks. The method first optimizes a low-dimensional latent vector such that the GAN can faithfully reproduce an input photo. The reconstruction method combines quasi-Newton optimization with encoder-based initialization. The system then modifies the appearance of the generated image using color, sketch, and warping tools. To render the result, they transfer the edits from the generated image to the original photo using guided image filtering [HST12]. Subsequent work on *Neural Photo Editing* [BLRW17] uses a VAE-GAN [LSLW16] to encode an image into a latent vector and generates an output by blending the modified content and the original pixels. The system allows semantic editing of faces, such as adding a beard. Several works [PVDWRÁ16, YHZ*18, HYHL18] train an encoder together with the generator. They deploy a second encoder to predict additional image attributes (e.g., semantics, 3D information, facial attributes) and allow a user to modify these attributes. This idea of using GANs as a deep image prior was later used in image inpainting [YCYL*17] and deblurring [ASA18]. The above systems work well on a low-resolution image with a single object or of a certain class and often require post-processing (e.g., filtering and blending) as the direct GANs' results are not realistic enough. To overcome these challenges, *GANPaint* [BSP*19a] adapts a pre-trained GAN model to a particular image. The learned image-specific GAN combines the prior learned from the entire image collection and image statistics of that particular image. Similar to prior work [ZKSE16, BLRW17], the method first projects an input image into a latent vector. The reconstruction from the vector is close to the input, but many visual details are missing. The method then slightly changes the network's internal parameters to reconstruct more precisely the input image. During test time, GANPaint modifies intermediate representations of GANs [BZS*19] according to user inputs. Instead of training a randomly initialized CNN on a single image as done in *Deep Image Prior* [UVL18], *GANPaint* leverages the prior learned from a pre-trained generative model and fine-tunes it for each input image. As shown in Figure 3, this enables addition and removal of certain objects in a realistic manner. Learning distribution priors via pre-training, followed by fine-tuning on limited data, is useful for many One-shot and Few-shot synthesis scenarios [BW18, LHM*19].

Figure 3: GANPaint *[BSP*19a] enables a few high-level image editing operations. A user can add, remove, or alter an object in an image with simple brush tools. A deep generative model will then satisfy user's constraint while preserving natural image statistics. Images taken from Bau et al. [BSP*19a].

6.1.3. Improving the Realism of Synthetic Renderings

The methods discussed above use deep generative models to either synthesize images from user-specified semantic layouts, or modify a given input image in a semantically meaningful manner. As noted before, rendering methods in computer graphics have been devel-

oped for the exact same goal—generating photorealistic images from scene specifications. However, the visual quality of computer rendered images depends on the fidelity of the scene modeling; using low-quality scene models and/or rendering methods results in images that look obviously synthetic. Johnson et al. [JDA*11] addressed this issue by improving the realism of synthetic renderings using content from similar, real photographs retrieved from a large-scale photo collection. However, this approach is restricted by the size of the database and the simplicity of the matching metric. Bi et al. [BSP*19b] propose using deep generative models to accomplish this task. They train a conditional generative model to translate a low-quality rendered image (along with auxiliary information like scene normals and diffuse albedo) to a high-quality photorealistic image. They propose performing this translation on an albedo-shading decomposition (instead of image pixels) to ensure that textures are preserved. Shrivastava et al. [SPT*17] learn to improve the realism of renderings of the human eyes based on unlabeled real images and Mueller et al. [MBS*18] employs a similar approach for human hands. Hoffman et al. [HTP*18] extends CycleGAN [ZPIE17] with feature matching to improve the realism of street view rendering for domain adaptation. Along similar lines, Nalbach et al. [NAM*17] propose using deep convolutional networks to convert shading buffers such as per-pixel positions, normals, and material parameters into complex shading effects like ambient occlusion, global illumination, and depth-of-field, thus significantly speeding up the rendering process. The idea of using coarse renderings in conjunction with deep generative models to generate high-quality images has also been used in approaches for facial editing [KGT*18, FTZ*19].

6.2. Novel View Synthesis for Objects and Scenes

Novel view synthesis is the problem of generating novel camera perspectives of a scene given a fixed set of images of the same scene. Novel view synthesis methods thus deal with image and video synthesis conditioned on camera pose. Key challenges underlying novel view synthesis are inferring the scene's 3D structure given sparse observations, as well as inpainting of occluded and unseen parts of the scene. In classical computer vision, image-based rendering (IBR) methods [DYB98, CDSHD13] typically rely on optimization-based multi-view stereo methods to reconstruct scene geometry and warp observations into the coordinate frame of the novel view. However, if only few observations are available, the scene contains view-dependent effects, or a large part of the novel perspective is not covered by the observations, IBR may fail, leading to results with ghosting-like artifacts and holes. Neural rendering approaches have been proposed to generate higher quality results. In Neural Image-based Rendering [HPP*18, MGK*19], previously hand-crafted parts of the IBR pipeline are replaced or augmented by learning-based methods. Other approaches reconstruct a learned representation of the scene from the observations, learning it end-to-end with a differentiable renderer. This enables learning of priors on geometry, appearance and other scene properties in a learned feature space. Such neural scene representation-based approaches range from prescribing little structure on the representation and the renderer [ERB*18], to proposing 3D-structured representations such as voxel grids of features [SZW19, NLT*19], to explicit 3D disentanglement of voxels and texture [ZZZ*18], point clouds [MGK*19], multi-plane images [XBS*19, FBD*19], or implicit functions [SHN*19, SZW19] which equip the network with inductive biases on image formation and geometry. Neural rendering approaches have made significant progress in previously open challenges such as the generation of view-dependent effects [TZT*20, XBS*19] or learning priors on shape and appearance from extremely sparse observations [SZW19, XBS*19]. While neural rendering shows better results compared to classical approaches, it still has limitations. I.e., they are restricted to a specific use-case and are limited by the training data. Especially, view-dependent effects such as reflections are still challenging.

6.2.1. Neural Image-based Rendering

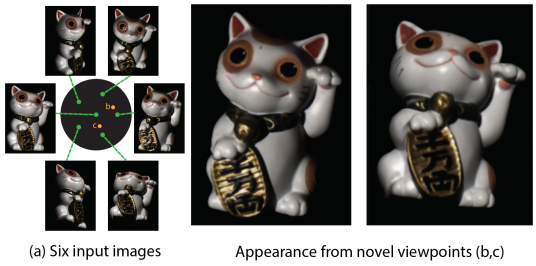
Neural Image-based Rendering (N-IBR) is a hybrid between classical image-based rendering and deep neural networks that replaces hand-crafted heuristics with learned components. A classical IBR method uses a set of captured images and a proxy geometry to create new images, e.g., from a different viewpoint. The proxy geometry is used to reproject image content from the captured images to the new target image domain. In the target image domain, the projections from the source images are blended to composite the final image. This simplified process gives accurate results only for diffuse objects with precise geometry reconstructed with a sufficient number of captured views. However, artifacts such as ghosting, blur, holes, or seams can arise due to view-dependent effects, imperfect proxy geometry or too few source images. To address these issues, N-IBR methods replace the heuristics often found in classical IBR methods with learned blending functions or corrections that take into account view-dependent effects. *DeepBlending* [HPP*18] proposes a generalized network to predict blending weights of the projected source images for compositing in the target image space. They show impressive results on indoor scenes with fewer blending artifacts than classical IBR methods. In *Image-guided Neural Object Rendering* [TZT*20], a scene specific network is trained to predict view-dependent effects with a network called *EffectsNet*. It is used to remove specular highlights from the source images to produce diffuse-only images, which can be projected into a target view without copying the source views' highlights. This EffectsNet is trained in a Siamese fashion on two different views at the same time, enforcing a multi-view photo consistency loss. In the target view, new view-dependent effects are reapplied and the images are blended using a U-Net-like architecture. As a result, this method demonstrates novel view point synthesis on objects and small scenes including view-dependent effects.

6.2.2. Neural Rerendering

Neural Rerendering combines classical 3D representation and renderer with deep neural networks that rerender the classical render into a more complete and realistic views. In contrast to Neural Image-based Rendering (N-IBR), neural rerendering does not use input views at runtime, and instead relies on the deep neural network to recover the missing details. *Neural Rerendering in the Wild* [MGK*19] uses neural rerendering to synthesize realistic views of tourist landmarks under various lighting conditions, see Figure 4. The authors cast the problem as a multi-modal image synthesis problem that takes as input a rendered deep buffer, containing depth and color channels, together with an appearance

(a) *Input deep buffer* **(b)** *Output renderings*

Figure 4: Neural Rerendering in the Wild *[MGK*19] reconstructs a proxy 3D model from a large-scale internet photo collection. This model is rendered into a deep buffer of depth, color and semantic labels (a). A neural rerendering network translates these buffers into realistic renderings under multiple appearances (b). Images taken from Meshry et al. [MGK*19].*



(a) Six input images Appearance from novel viewpoints (b,c)

Figure 5: *Xu et al. [XBS*19] render scene appearance from a novel viewpoint, given only six sparse, wide baseline views. Images taken from Xu et al. [XBS*19].*

code, and outputs realistic views of the scene. The system reconstructs a dense colored point cloud from internet photos using Structure-from-Motion and Multi-View Stereo, and for each input photo, renders the recovered point cloud into the estimated camera. Using pairs of real photos and corresponding rendered deep buffers, a multi-modal image synthesis pipeline learns an implicit model of appearance, that represents time of day, weather conditions and other properties not present in the 3D model. To prevent the model from synthesizing transient objects, like pedestrians or cars, the authors propose conditioning the rerenderer with a semantic labeling of the expected image. At inference time, this semantic labeling can be constructed to omit any such transient objects. Pittaluga et al. [PKKS19] use a neural rerendering technique to invert Structure-from-Motion reconstructions and highlight the privacy risks of Structure-from-Motion 3D reconstructions, that typically contain color and SIFT features. The authors show how sparse point clouds can be inverted and generate realistic novel views from them. In order to handle very sparse inputs, they propose a visibility network that classifies points as visible or not and is trained with ground-truth correspondences from 3D reconstructions.

6.2.3. Novel View Synthesis with Multiplane Images

Given a sparse set of input views of an object, Xu et al. [XBS*19] also address the problem of rendering the object from novel viewpoints (see Figure 5). Unlike previous view interpolation methods that work with images captured under natural illumination and at a small baseline, they aim to capture the light transport of the scene, including view-dependent effects like specularities. Moreover they attempt to do this from a sparse set of images captured at large baselines, in order to make the capture process more light-weight. They capture six images of the scene under point illumination in a cone of about 60° and render any novel viewpoint within this cone. The input images are used to construct a plane sweeping volume aligned with the novel viewpoint [FNPS16]. This volume is processed by 3D CNNs to reconstruct both scene depth and appearance. To handle the occlusions caused by the large baseline, they propose predicting attention maps that capture the visibility of the input viewpoints at different pixels. These attention maps are used to modulate the appearance plane sweep volume and remove inconsistent content. The network is trained on synthetically rendered data with supervision on both geometry and appearance; at test time it is able to synthesize photo-realistic results of real scenes featuring high-frequency light transport effects such as shadows and specularities. *DeepView* [FBD*19] is a technique to visualize light fields under novel views. The view synthesis is based on multiplane images [ZTF*18] that are estimated by a learned gradient descent method given a sparse set of input views. Similar to image-based rendering, the image planes can be warped to new views and are rendered back-to-front into the target image.

6.2.4. Neural Scene Representation and Rendering

While neural rendering methods based on multi-plane images and image-based rendering have enabled some impressive results, they prescribe the model's internal representation of the scene as a point cloud, a multi-plane image, or a mesh, and do not allow the model to learn an optimal representation of the scene's geometry and appearance. A recent line in novel view synthesis is thus to build models with neural scene representations: learned, feature-based representations of scene properties. The Generative Query Network [ERB*18] is a framework for learning a low-dimensional feature embedding of a scene, explicitly modeling the stochastic nature of such a neural scene representation due to incomplete observations. A scene is represented by a collection of observations, where each observation is a tuple of an image and its respective camera pose. Conditioned on a set of context observations and a target camera pose, the GQN parameterizes a distribution over frames observed at the target camera pose, consistent with the context observations. The GQN is trained by maximizing the log-likelihood of each observation given other observations of the same scene as context. Given several context observations of a single scene, a convolutional encoder encodes each of them into a low-dimensional latent vector. These latent vectors are aggregated to a single representation r via a sum. A convolutional Long-Short Term Memory network (ConvLSTM) parameterizes an auto-regressive prior distribution over latent variables z. At every timestep, the hidden state of the ConvLSTM is decoded into a residual update to a canvas u that represents the sampled observation. To make the optimization problem tractable, the GQN uses an approximate posterior at training time. The authors demonstrate the capability of the GQN to learn a rich feature representation of the scene on novel view synthesis, control of a simulated robotic arm, and the exploration of a labyrinth environment. The probabilistic formulation of the GQN allows the model to sample different frames all consistent

with context observations, capturing, for instance, the uncertainty about parts of the scene that were occluded in context observations.

6.2.5. Voxel-based Novel View Synthesis Methods

While learned, unstructured neural scene representations are an attractive alternative to hand-crafted scene representations, they come with a number of drawbacks. First and foremost, they disregard the natural 3D structure of scenes. As a result, they fail to discover multi-view and perspective geometry in regimes of limited training data. Inspired by recent progress in geometric deep learning [KHM17, CXG*16, JREM*16, HMR19], a line of neural rendering approaches has emerged that instead proposes to represent the scene as a voxel grid, thus enforcing 3D structure.

RenderNet [NPLBY18] proposes a convolutional neural network architecture that implements differentiable rendering from a scene explicitly represented as a 3D voxel grid. The model is retrained for each class of objects and requires tuples of images with labeled camera pose. RenderNet enables novel view synthesis, texture editing, relighting, and shading. Using the camera pose, the voxel grid is first transformed to camera coordinates. A set of 3D convolutions extracts 3D features. The 3D voxel grid of features is translated to a 2D feature map via a subnetwork called the "projection unit." The projection unit first collapses the final two channels of the 3D feature voxel grid and subsequently reduces the number of channels via 1x1 2D convolutions. The 1x1 convolutions have access to all features along a single camera ray, enabling them to perform projection and visibility computations of a typical classical renderer. Finally, a 2D up-convolutional neural network upsamples the 2D feature map and computes the final output. The authors demonstrate that RenderNet learns to render high-resolution images from low-resolution voxel grids. RenderNet can further learn to apply varying textures and shaders, enabling scene relighting and novel view synthesis of the manipulated scene. They further demonstrate that RenderNet may be used to recover a 3D voxel grid representation of a scene from single images via an iterative reconstruction algorithm, enabling subsequent manipulation of the representation.

DeepVoxels [STH*19] enables joint reconstruction of geometry and appearance of a scene and subsequent novel view synthesis. DeepVoxels is trained on a specific scene, given only images as well as their extrinsic and intrinsic camera parameters – no explicit scene geometry is required. This is achieved by representing a scene as a Cartesian 3D grid of embedded features, combined with a network architecture that explicitly implements image formation using multi-view and projective geometry operators. Features are first extracted from 2D observations. 2D features are then un-projected by replicating them along their respective camera rays, and integrated into the voxel grid by a small 3D U-net. To render the scene with given camera extrinsic and intrinsic parameters, a virtual camera is positioned in world coordinates. Using the intrinsic camera parameters, the voxel grid is resampled into a canonical view volume. To reason about occlusions, the authors propose an occlusion reasoning module. The occlusion module is implemented as a 3D U-Net that receives as input all the features along a camera ray as well as their depth, and produces as output a visibility score for each feature along the ray, where the scores along each ray sum to one. The final projected feature is then computed as a weighted sum of features along each ray. Finally, the resulting 2D feature map is translated to an image using a small U-Net. As a side-effect of the occlusion reasoning module, DeepVoxels produces depth maps in an unsupervised fashion. The model is fully differentiable from end to end, and is supervised only by a 2D re-rendering loss enforced over the training set. The paper shows wide-baseline novel view synthesis on several challenging scenes, both synthetic and real, and outperforms baselines that do not use 3D structure by a wide margin.

Visual Object Networks (VONs) [ZZZ*18] is a 3D-aware generative model for synthesizing the appearance of objects with a disentangled 3D representation. Inspired by classic rendering pipelines, VON decomposes the neural image formation model into three factors—viewpoint, shape, and texture. The model is trained with an end-to-end adversarial learning framework that jointly learns the distribution of 2D images and 3D shapes through a differentiable projection module. During test time, VON can synthesize a 3D shape, its intermediate 2.5D depth representation, and a final 2D image all at once. This 3D disentanglement allows users to manipulate the shape, viewpoint, and texture of an object independently.

HoloGAN [NLT*19] builds on top of the learned projection unit of RenderNet to build an unconditional generative model that allows explicit viewpoint changes. It implements an explicit affine transformation layer that directly applies view manipulations to learnt 3D features. As in DeepVoxels, the network learns a 3D feature space, but more bias about the 3D object / scene is introduced by transforming these deep voxels with a random latent vector z. In this way, an unconditional GAN that natively supports viewpoint changes can be trained in an unsupervised fashion. Notably, HoloGAN requires neither pose labels and intrinsic camera information nor multiple views of an object.

6.2.6. Implicit-function based Approaches

While 3D voxel grids have demonstrated that a 3D-structured scene representation benefits multi-view consistent scene modeling, their memory requirement scales cubically with spatial resolution, and they do not parameterize surfaces smoothly, requiring a neural network to learn priors on shape as joint probabilities of neighboring voxels. As a result, they cannot parameterize large scenes at a sufficient spatial resolution, and have so far failed to generalize shape and appearance across scenes, which would allow applications such as reconstruction of scene geometry from only few observations. In geometric deep learning, recent work alleviated these problems by modeling geometry as the level set of a neural network [PFS*19, MON*19]. Recent neural rendering work generalizes these approaches to allow rendering of full color images. In addition to parameterizing surface geometry via an implicit function, *Pixel-Aligned Implicit Functions* [SHN*19] represent object color via an implicit function. An image is first encoded into a pixel-wise feature map via a convolutional neural network. A fully connected neural network then takes as input the feature at a specific pixel location as well as a depth value z, and classifies the depth as inside/outside the object. The same architecture is used to encode color. The model is trained end-to-end, supervised with images and 3D geometry. The authors demonstrate single- and multi-shot 3D reconstruction and novel view synthesis of clothed humans. *Scene Representation Networks* (SRNs) [SZW19] encodes both scene geometry and appearance in a single fully connected neural network,

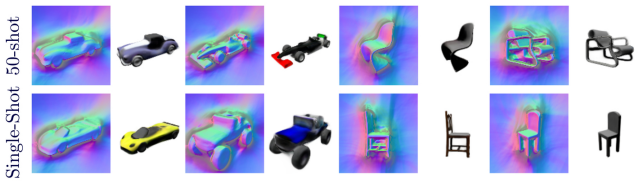


Figure 6: Scene Representation Networks *[SZW19] allow reconstruction of scene geometry and appearance from images via a continuous, 3D-structure-aware neural scene representation, and subsequent, multi-view-consistent view synthesis. By learning strong priors, they allow full 3D reconstruction from only a single image (bottom row, surface normals and color render). Images taken from Sitzmann et al. [SZW19].*

the SRN, that maps world coordinates to a feature representation of local scene properties. A differentiable, learned neural ray-marcher is trained end-to-end given only images and their extrinsic and intrinsic camera parameters—no ground-truth shape information is required. The SRN takes as input (x, y, z) world coordinates and computes a feature embedding. To render an image, camera rays are traced to their intersections with scene geometry (if any) via a differentiable, learned raymarcher, which computes the length of the next step based on the feature returned by the SRN at the current intersection estimate. The SRN is then sampled at ray intersections, yielding a feature for every pixel. This 2D feature map is translated to an image by a per-pixel fully connected network. Similarly to DeepSDF [PFS*19], SRNs generalize across scenes in the same class by representing each scene by a code vector $\mathbf{z}$. The code vectors $\mathbf{z}$ are mapped to the parameters of a SRN via a fully connected neural network, a so-called hypernetwork. The parameters of the hypernetwork are jointly optimized with the code vectors and the parameters of the pixel generator. The authors demonstrate single-image reconstruction of geometry and appearance of objects in the ShapeNet dataset (Figure 6), as well as multi-view consistent view synthesis. Due to their per-pixel formulation, SRNs generalize to completely unseen camera poses like zoom or camera roll.

6.3. Free Viewpoint Videos

Free Viewpoint Videos, also known as *Volumetric Performance Capture*, rely on multi-camera setups to acquire the 3D shape and texture of performers. The topic has gained a lot of interest in the research community starting from the early work of Tanco and Hilton [TH00] and reached compelling high quality results with the works of Collet et al. [CCS*15] and its real-time counterpart by Dou et al. [DKD*16, DDF*17]. Despite the efforts, these systems lack photorealism due to missing high frequency details [OERF*16] or baked in texture [CCS*15], which does not allow for accurate and convincing re-lighting of these models in arbitrary scenes. Indeed, volumetric performance capture methods lack view dependent effects (e.g. specular highlights); moreover, imperfections in the estimated geometry usually lead to blurred texture maps. Finally, creating temporally consistent 3D models [CCS*15] is very challenging in many real world cases (e.g. hair, translucent materials). A recent work on human performance capture by Guo et al. [GLD*19] overcomes many of these limitations by combining traditional image based relighting methods [DHT*00] with recent advances in high-

Figure 7: The Relightables *system by Guo et al. [GLD*19] for free viewpoint capture of humans with realistic re-lighting. From left to right: a performer captured in the Lightstage, the estimated geometry and albedo maps, examples of relightable volumetric videos. Images taken from Guo et al. [GLD*19].*

speed and accurate depth sensing [KRF*18, TSF*18]. In particular, this system uses 58 12.4MP RGB cameras combined with 32 12.4MP active IR sensors to recover very accurate geometry. During the capture, the system interleaves two different lighting conditions based on spherical gradient illumination [FWD09]. This produces an unprecedented level of photorealism for a volumetric capture pipeline (Figure 7). Despite steady progress and encouraging results obtained by these 3D capture systems, they still face important challenges and limitations. Translucent and transparent objects cannot be easily captured; reconstructing thin structures (e.g. hair) is still very challenging even with high resolution depth sensors. Nevertheless, these multi-view setups provide the foundation for machine learning methods [MBPY*18,LSSS18,LSS*19,PTY*19], which heavily rely on training data to synthesize high quality humans in arbitrary views and poses.

6.3.1. LookinGood with Neural Rerendering

The *LookinGood* system by Martin-Brualla et al. [MBPY*18] introduced the concept of neural rerendering for performance capture of human actors. The framework relies on a volumetric performance capture system [DDF*17], which reconstructs the performer in real-time. These models can then be rendered from arbitrary viewpoints using the known geometry. Due to real-time constraints, the reconstruction quality suffers from many artifacts such as missing depth, low resolution texture and oversmooth geometry. Martin-Brualla et al. propose to add "witness cameras", which are high resolution RGB sensors (12MP) that are not used in the capture system, but can provide training data for a deep learning architecture to re-render the output of the geometrical pipeline. The authors show that this enables high quality re-rendered results for arbitrary viewpoints, poses and subjects in real-time. The problem at hand is very interesting since it is tackling denoising, in-painting and super-resolution simultaneously and in real-time. In order to solve this, authors cast the task into an image-to-image translation problem [IZZE17] and they introduce a semantic aware loss function that uses the semantic information (available at training time) to increase the weights of the pixels belonging to salient areas of the image and to retrieve precise silhouettes ignoring the contribution of the background. The overall system is shown in Figure 8.

6.3.2. Neural Volumes

Neural Volumes [LSS*19] addresses the problem of automatically creating, rendering, and animating high-quality object models from

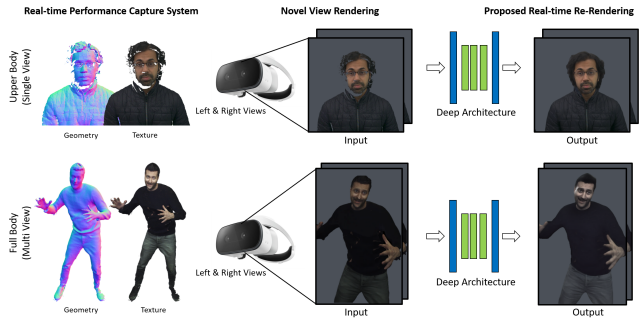


Figure 8: *The* LookinGood *system [MBPY*18] uses real-time neural re-rendering to enhance performance capture systems. Images taken from Martin-Brualla et al. [MBPY*18].*

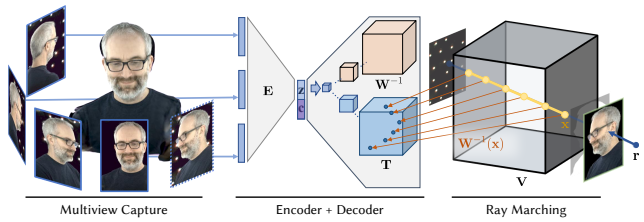


Figure 9: *Pipeline for* Neural Volumes *[LSS*19]. Multi-view capture is input to an encoder to produce a latent code* **z**. **z** *is decoded to a volume that stores RGBα values, as well as a warp field. Differentiable ray marching renders the volume into an image, allowing the system to be trained by minimizing the difference between rendered and target images. Images taken from Lombardi et al. [LSS*19].*

multi-view video data (see Figure 9). The method trains a neural network to encode frames of a multi-view video sequence into a compact latent code which is decoded into a semi-transparent volume containing RGB and opacity values at each (x,y,z) location. The volume is rendered by raymarching from the camera through the volume, accumulating color and opacity to form an output image and alpha matte. Formulating the problem in 3D rather than in screen space has several benefits: viewpoint interpolation is improved because the object must be representable as a 3D shape, and the method can be easily combined with traditional triangle-mesh rendering. The method produces high-quality models despite using a low-resolution voxel grid (128^3) by introducing a learned warp field that not only helps to model the motion of the scene but also reduces blocky voxel grid artifacts by deforming voxels to better match the geometry of the scene and allows the system to shift voxels to make better use of the voxel resolution available. The warp field is modeled as a spatially-weighted mixture of affine warp fields, which can naturally model piecewise deformations. By virtue of the semi-transparent volumetric representation, the method can reconstruct challenging objects such as moving hair, fuzzy toys, and smoke all from only 2D multi-view video with no explicit tracking required. The latent space encoding enables animation by generating new latent space trajectories or by conditioning the decoder on some information like head pose.

6.3.3. Free Viewpoint Videos from a Single Sensor

The availability of multi-view images at training and test time is one of the key elements for the success of free viewpoint systems. However this capture technology is still far from being accessible to a typical consumer who, at best, may own a single RGBD sensor such as a Kinect. Therefore, parallel efforts [PTY*19] try to make the capture technology accessible through consumer hardware by dropping the infrastructure requirements through deep learning. Reconstructing performers from a single image is very related to the topic of synthesizing humans in unseen poses [ZWC*17, BZD*18, SWWT18, MSG*18, MJS*17, NGK18, CGZE18]. Differently from the other approaches, the recent work of Pandey et al. [PTY*19] synthesizes performers in *unseen poses* and from *arbitrary viewpoints*, mimicking the behavior of volumetric capture systems. The task at hand is much more challenging because it requires disentangling pose, texture, background and viewpoint. Pandey et al. propose to solve this problem by leveraging a semi-parametric model. In particular they assume that a short calibration sequence of the user is available: e.g. the user rotates in front of the camera before the system starts. Multiple deep learning stages learn to combine the current viewpoint (which contains the correct user pose and expression) with the pre-recorded calibration images (which contain the correct viewpoint but wrong poses and expressions). The results are compelling given the substantial reduction in the infrastructure required.

6.4. Learning to Relight

Photo-realistically rendering of a scene under novel illumination—a procedure known as "relighting"—is a fundamental component of a number of graphics applications including compositing, augmented reality and visual effects. An effective way to accomplish this task is to use image-based relighting methods that take as input images of the scene captured under different lighting conditions (also known as a "reflectance field"), and combine them to render the scene's appearance under novel illumination [DHT*00]. Image-based relighting can produce high-quality, photo-realistic results and has even been used for visual effects in Hollywood productions. However, these methods require slow data acquisition with expensive, custom hardware, precluding the applicability of such methods to settings like dynamic performance and "in-the-wild" capture. Recent methods address these limitations by using synthetically rendered or real, captured reflectance field data to train deep neural networks that can relight scenes from just a few images.

6.4.1. Deep Image-based Relighting from Sparse Samples

Xu et al. [XSHR18] propose an image-based relighting method that can relight a scene from a sparse set of five images captured under learned, optimal light directions. Their method uses a deep convolutional neural network to regress a relit image under an arbitrary directional light from these five images. Traditional image-based relighting methods rely on the *linear* superposition property of lighting, and thus require tens to hundreds of images for high-quality results. Instead, by training a *non-linear* neural relighting network, this method is able to accomplish relighting from sparse images. The relighting quality depends on the input light directions, and the authors propose combining a custom-designed sampling network with the relighting network, in an end-to-end fashion, to jointly learn both the optimal input light directions and the relighting function. The entire system is trained on a large synthetic

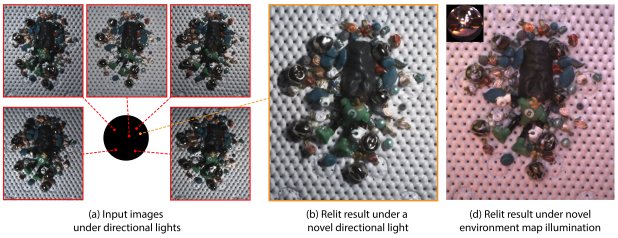


Figure 10: *Xu et al. [XSHR18] are able to generate relit versions of a scene under novel directional and environment map illumination from only five images captured under specified directional lights. Images taken from Xu et al. [XSHR18].*

dataset comprised of procedurally generated shapes rendered with complex, spatially-varying reflectances. At test time, the method is able to relight real scenes and reproduce complex, high-frequency lighting effects like specularities and cast shadows.

6.4.2. Multi-view Scene Relighting

Given multiple views of a large-scale outdoor scene captured under uncontrolled natural illumination, Philip et al. [PGZ*19] can render the scene under novel outdoor lighting (parameterized by the sun position and cloudiness level).The input views are used to reconstruct the 3D geometry of the scene; this geometry is coarse and erroneous and directly relighting it would produce poor results. Instead, the authors propose using this geometry to construct intermediate buffers—normals, reflection features, and RGB shadow maps—as auxiliary inputs to guide a neural network-based relighting method. The method also uses a shadow refinement network to improve the removal and addition of shadows that are an important cue in outdoor images. While the entire method is trained on a synthetically rendered dataset, it generalizes to real scenes, producing high-quality results for applications like the creation of time-lapse effects from multiple (or single) images and relighting scenes in traditional image-based rendering pipelines.

6.4.3. Deep Reflectance Fields

Deep Reflectance Fields [MHP*19] presents a novel technique to relight images of human faces by learning a model of facial reflectance from a database of 4D reflectance field data of several subjects in a variety of expressions and viewpoints. Using a learned model, a face can be relit in arbitrary illumination environments using only two original images recorded under spherical color gradient illumination [FWD09]. The high-quality results of the method indicate that the color gradient images contain the information needed to estimate the full 4D reflectance field, including specular reflections and high frequency details. While capturing images under spherical color gradient illumination still requires a special lighting setup, reducing the capture requirements to just two illumination conditions, as compared to previous methods that require hundreds of images [DHT*00], allows the technique to be applied to *dynamic* facial performance capture (Figure 11).

6.4.4. Single Image Portrait Relighting

A particularly useful application of relighting methods is to change the lighting of a portrait image captured in the wild, i.e., with off-the-shelf (possibly cellphone) cameras under natural unconstrained lighting. While the input in this case is only a single (uncontrolled) image, recent methods have demonstrated state-of-the-art results using deep neural networks [SBT*19, ZHSJ19]. The relighting model in these methods consists of a deep neural network that has been trained to take a single RGB image as input and produce as output a relit version of the portrait image under an arbitrary user-specified environment map. Additionally, the model also predicts an estimation of the current lighting conditions and, in the case of Sun et al. [SBT*19], can run on mobile devices in ~160ms, see Figure 12. Sun et al. represent the target illumination as an environment map and train their network using captured reflectance field data. On the other hand, Zhou et al. [ZHSJ19] use a spherical harmonics representation for the target lighting and train the network with a synthetic dataset created by relighting single portrait images using a traditional ratio image-based method. Instead of having an explicit inverse rendering step for estimating geometry and reflectance [BM14, SYH*17, SKCJ18], these methods directly regress to the final relit image from an input image and a "target" illumination. In doing so, they bypass restrictive assumptions like Lambertian reflectance and low-dimensional shape spaces that are made in traditional face relighting methods, and are able to generalize to full portrait image relighting including hair and accessories.

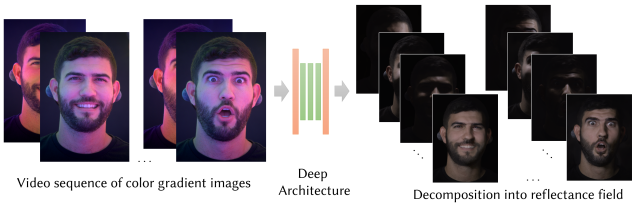


Figure 11: *Meka et al. [MHP*19] decompose the full reflectance fields by training a convolutional neural network that maps two spherical gradient images to any one-light-at-a-time image. Images taken from Meka et al. [MHP*19].*

Figure 12: *Given a single portrait image captured with a standard camera, portrait relighting methods [SBT*19] can generate images of the subject under novel lighting environments. Images taken from Sun et al. [SBT*19].*

6.5. Facial Reenactment

Facial reenactment aims to modify scene properties beyond those of viewpoint (Section 6.3) and lighting (Section 6.4), for example by generating new head pose motion, facial expressions, or speech. Early methods were based on classical computer graphics techniques. While some of these approaches only allow implicit control, i.e., retargeting facial expressions from a source to

a target sequence [TZS*16], explicit control has also been explored [BBPV03]. These approaches usually involve reconstruction of a 3D face model from the input, followed by editing and rendering of the model to synthesize the edited result. Neural rendering techniques overcome the limitations of classical approaches by better dealing with inaccurate 3D reconstruction and tracking, as well as better photorealistic appearance rendering. Early neural rendering approaches, such as that of Kim et al. [KGT*18], use a conditional GAN to refine the outputs estimated by classical methods. In addition to more photo-realistic results compared to classical techniques, neural rendering methods allow for the control of head pose in addition to facial expressions [KGT*18, ZSBL19, NSX*18, WKZ18]. Most neural rendering approaches for facial reenactment are trained separately for each identity. Only recently, methods which generalize over multiple identities have been explored [ZSBL19, WKZ18, NSX*18, GSZ*18].

6.5.1. Deep Video Portraits

Deep Video Portraits [KGT*18] is a system for full head reenactment of portrait videos. The head pose, facial expressions and eye motions of the person in a video are transferred from another reference video. A facial performance capture method is used to compute 3D face reconstructions for both reference and target videos. This reconstruction is represented using a low-dimensional semantic representation which includes identity, expression, pose, eye motion, and illumination parameters. Then, a rendering-to-video translation network, based on U-Nets, is trained to convert classical computer graphics renderings of the 3D models to photo-realistic images. The network adds photo-realistic details on top of the imperfect face renderings, in addition to completing the scene by adding hair, body, and background. The training data consists of pairs of training frames, and their corresponding 3D reconstructions. Training is identity and scene specific, with only a few minutes (typically 5-10) of training data needed. At test time, semantic dimensions which are relevant for reenactment, i.e., expressions, eye motion and rigid pose are transferred from a source to a different target 3D model. The translation network subsequently converts the new 3D sequence into a photo-realistic output sequence. Such a framework allows for interactive control of a portrait video.

6.5.2. Editing Video by Editing Text

Text-based Editing of Talking-head Video [FTZ*19] takes as input a one-hour long video of a person speaking, and the transcript of that video. The editor changes the transcript in a text editor, and the system synthesizes a new video in which the speaker appears to be speaking the revised transcript (Figure 13). The system supports cut, copy and paste operations, and is also able to generate new words that were never spoken in the input video. The first part of the pipeline is not learning-based. Given a new phrase to synthesize, the system finds snippets in the original input video that, if combined, will appear to be saying the new phrase. To combine the snippets, a parameterized head model is used, similarly to Deep Video Portraits [KGT*18]. Each video frame is converted to a low-dimensional representation, in which expression parameters (i.e. what the person is saying and how they are saying it) are decoupled from all other properties of the scene (e.g. head pose and global illumination). The snippets and parameterized model are

Figure 13: Text-based Editing of Talking-head Video *[FTZ*19]. An editor changes the text transcript to create a new video in which the subject appears to be saying the new phrase. In each pair, left: composites containing real pixels, rendered pixels and transition regions (in black); right: photo-realistic neural-rendering results. Images taken from Fried et al. [FTZ*19].

then used to synthesize a low-fidelity render of the desired output. Neural rendering is used to convert the low-fidelity render into a photo-realistic frame. A GAN-based encoder-decoder, again similar to Deep Video Portraits, is trained to add high frequency details (e.g., skin pores) and hole-fill to produce the final result. The neural network is person specific, learning a person's appearance variation in a given environment. In contrast to Kim et al. [KGT*18], this network can deal with dynamic backgrounds.

6.5.3. Image Synthesis using Neural Textures

Deferred Neural Rendering [TZN19] enables novel-view point synthesis as well as scene-editing in 3D (geometry deformation, removal, copy-move). It is trained for a specific scene or object. Besides ground truth color images, it requires a coarse reconstructed and tracked 3D mesh including a texture parametrization. Instead of a classical texture as used by Kim et al. [KGT*18], the approach learns a neural texture, a texture that contains neural feature descriptors per surface point. A classical computer graphics rasterizer is used to sample from these neural textures, given the 3D geometry and view-point, resulting in a projection of the neural feature descriptors onto the image plane. The final output image is generated from the rendered feature descriptors using a small U-Net, which is trained in conjunction with the neural texture. The paper shows several applications based on color video inputs, including novel-view point synthesis, scene editing and animation synthesis of portrait videos. The learned neural feature descriptors and the decoder network compensate for the coarseness of the underlying geometry, as well as for tracking errors, while the classical rendering step ensures 3D-consistent image formation. Similar to the usage of neural textures on meshes, Aliev et al. [AUL19] propose to use vertex-located feature descriptors and point based rendering to project these to the image plane. Given the splatted features as input, a U-Net architecture is used for image generation. They show results on objects and room scenes.

6.5.4. Neural Talking Head Models

The facial reenactment approaches we have discussed so far [KGT*18, FTZ*19, TZN19] are person-specific, i.e., a different network has to be trained for each identity. In contrast, a *generalized* face reenactment approach was proposed by Zakharov et al. [ZSBL19]. The authors train a common network to control faces of any identity using sparse 2D keypoints. The network consists of an embedder network to extract pose-independent identity information. The output of this network is then fed to

the generator network, which learns to transform given input keypoints into photo-realistic frames of the person. A large video dataset [CNZ18] consisting of talking videos of a large number of identities is used to train the network. At test time, few-shot learning is used to refine the network for an unseen identity, similar to Liu et al. [LHM*19]. While the approach allows for control over unseen identities, it does not allow for explicit 3D control of scene parameters such as pose and expressions. It needs a reference video to extract the keypoints used as input for the network.

6.5.5. Deep Appearance Models

Deep Appearance Models [LSSS18] model facial geometry and appearance with a conditional variational autoencoder. The VAE compresses input mesh vertices and a texture map into a small latent encoding of facial expression. Importantly, this VAE is conditioned on the viewpoint of the camera used to render the face. This enables the decoder network to correct geometric tracking errors by decoding a texture map that reprojects to the correct place in the rendered image. The result is that the method produces high-quality, high-resolution viewpoint-dependent renderings of the face that runs at 90Hz in virtual reality. The second part of this method is a system for animating the learned face model from cameras mounted on a virtual reality headset. Two cameras are placed inside the headset looking at the eyes and one is mounted at the bottom looking at the mouth. To generate correspondence between the VR headset camera images and the multi-view capture stage images, the multi-view capture stage images are re-rendered using image-based rendering from the point-of-view of the VR headset cameras, and a single conditional variational autoencoder is used to learn a common encoding of the multi-view capture images and VR headset images, which can be regressed to the latent facial code learned in the first part. Wei et al. [WSS*19] improve the facial animation system by solving for the latent facial code that decodes an avatar such that, when rendered from the perspective of the headset, most resembles a set of VR headset camera images, see Figure 14. To make this possible, the method uses a "training" headset, that includes an additional 6 cameras looking at the eyes and mouth, to better condition the analysis-by-synthesis formulation. The domain gap between rendered avatars and headset images is closed by using unsupervised image-to-image translation techniques. This system is able to more precisely match lip shapes and better reproduce complex facial expressions. While neural rendering approaches for facial reenactment achieve impressive results, many challenges still remain to be solved. Full head reenactment, including control over the head pose, is very challenging in dynamic environments. Many of the methods discussed do not preserve high-frequency details in a temporally coherent manner. In addition, photorealistic synthesis and editing of hair motion and mouth interior including tongue motion is challenging. Generalization across different identities without any degradation in quality is still an open problem.

6.6. Body Reenactment

Neural pose-guided image [SSLS18, MJS*17, ESO18] and video [LXZ*19, CGZE18, ASL*19] generation enables the control of the position, rotation, and body pose of a person in a target image/video (see Figure 15). The problem of generating realistic images of the full human body is challenging, due to the large non-linear motion space of humans. Full body performance cloning approaches [LXZ*19, CGZE18, ASL*19] transfer the motion in a source video to a target video. These approaches are commonly trained in a person-specific manner, i.e., they require a several-minutes-long video (often with a static background) as training data for each new target person. In the first step, the motion of the target is reconstructed based on sparse or dense human performance capture techniques. This is required to obtain the paired training corpus (pose and corresponding output image) for supervised training of the underlying neural rendering approach. Current approaches cast the problem of performance cloning as learning a conditional generative mapping based on image-to-image translation networks. The inputs are either joint heatmaps [ASL*19], the rendered skeleton model [CGZE18], or a rendered mesh of the human [LXZ*19]. The approach of Chan et al. [CGZE18] predicts two consecutive frames of the output video and employs a space-time discriminator for more temporal coherence. For better generalization, the approach of Aberman et al. [ASL*19] employs a network with two separate branches that is trained in a hybrid manner based on a mixed training corpus of paired and unpaired data. The paired branch employs paired training data extracted from a reference video to directly supervise image generation based on a reconstruction loss. The unpaired branch is trained with unpaired data based on an adversarial identity loss and a temporal coherence loss. *Textured Neural Avatars* [SZA*19] predict dense texture coordinates based on rendered skeletons to sample a learnable, but static, RGB texture. Thus, they remove the need of an explicit geometry at training and

Figure 14: *Correspondence found by the system of Wei et al. [WSS*19]. Cameras placed inside a virtual reality head-mounted display (HMD) produce a set of infrared images. Inverse rendering is used to find the latent code of a Deep Appearance Model [LSSS18] corresponding to the images from the HMD, enabling a full-face image to be rendered. Images taken from Wei et al. [WSS*19].*

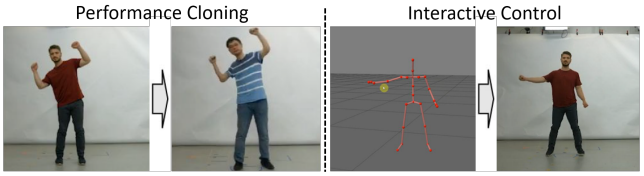


Figure 15: *Neural pose-guided image and video generation enables the control of the position, rotation, and body pose of a person in a target video. For human performance cloning the motion information is extracted from a source video clip (left). Interactive user control is possible by modifying the underlying skeleton model (right). Images taken from Liu et al. [LXZ*19].*

test time by mapping multiple 2D views to a common texture map. Effectively, this maps a 3D geometry into a global 2D space that is used to re-render an arbitrary view at test-time using a deep network. The system is able to infer novel viewpoints by conditioning the network on the desired 3D pose of the subject. Besides texture coordinates, the approach also predicts a foreground mask of the body. To ensure convergence, the authors prove the need of a pre-trained DensePose model [GNK18] to initialize the common texture map, at the same time they show how their training procedure improves the accuracy of the 2D correspondences and sharpens the texture map by recovering high frequency details. Given new skeleton input images they can also drive the learned pipeline. This method shows consistently improved generalization compared to standard image-to-image translation approaches. On the other hand, the network is trained per-subject and cannot easily generalize to unseen scales. Since the problem of human performance cloning is highly challenging, none of the existing methods obtain artifact-free results. Remaining artifacts range from incoherently moving surface detail to partly missing limbs.

7. Open Challenges

As this survey shows, significant progress on neural rendering has been made over the last few years and it had a high impact on a vast number of application domains. Nevertheless, we are still just at the beginning of this novel paradigm of learned image-generation approaches, which leaves us with many open challenges, but also incredible opportunities for further advancing this field. In the following, we describe open research problems and suggest next steps.

Generalization. Many of the first neural rendering approaches have been based on overfitting to a small set of images or a particular video depicting a single person, object, or scene. This is the best case scenario for a learning based approach, since the variation that has to be learned is limited, but it also restricts generalization capabilities. In a way, these approaches learn to interpolate between the training examples. As is true for any machine learning approach, they might fail if tested on input that is outside the span of the training samples. For example, learned reenactment approaches might fail for unseen poses [KGT*18, LXZ*19, CGZE18, ASL*19]. Nevertheless, the neural rendering paradigm already empowers many applications in which the data distribution at test and training time is similar. One solution to achieve better generalization is to explicitly add the failure cases to the training corpus, but this comes at the expense of network capacity and all failures cases might not be known a priori. Moreover, if many of the scene parameters have to be controlled, the curse of dimensionality makes capturing all potential scenarios infeasible. Even worse, if a solution should work for arbitrary people, we cannot realistically gather training data for all potential users, and even if we could it is unclear whether such training will be successful. Thus, one of the grand challenges for the future is true generalization to unseen settings. For examples, first successful steps have been taken to generalize 3D-structured neural scene representations [SZW19, NLT*19, NPLBY18] across object categories. One possibility to improve generalization is to explicitly build a physically inspired inductive bias into the network. Such an inductive bias can for example be a differentiable camera model or an explicit 3D-structured latent space. This analytically enforces a truth about the world in the network structure and frees up network capacity. Together, this enables better generalization, especially if only limited training data is available. Another interesting direction is to explore how additional information at test time can be employed to improve generalization, e.g., a set of calibration images [PTY*19] or a memory bank.

Scalability. So far, a lot of the effort has focused on very specific applications that are constrained in the complexity and size of the scenes they can handle. For example, work on (re-)rendering faces has primarily focused on processing a single person in a short video clip. Similarly, neural scene representations have been successful in representing individual objects or small environments of limited complexity. While network generalization may be able to address a larger diversity of objects or simple scenes, scalability is additionally needed to successfully process complex, cluttered, and large scenes, for example to enable dynamic crowds, city- or global-scale scenes to be efficiently processed. Part of such an effort is certainly software engineering and improved use of available computational resources, but one other possible direction that could allow neural rendering techniques to scale is to let the network reason about compositionality. A complex scene can be understood as the sum of its parts. For a network to efficiently model this intuition, it has to be able to segment a scene into objects, understand local coordinate systems, and robustly process observations with partial occlusions or missing parts. Yet, compositionality is just one step towards scalable neural rendering techniques and other improvements in neural network architectures and steps towards unsupervised learning strategies have to be developed.

Editability. Traditional computer graphics pipelines are not only optimized for modeling and rendering capabilities, but they also allow all aspects of a scene to be edited either manually or through simulation. Neural rendering approaches today do not always offer this flexibility. Those techniques that combine learned parameters with traditional parts of the pipeline, such as neural textures, certainly allow the traditional part (i.e., the mesh) to be edited but it is not always intuitive how to edit the learned parameters (i.e., the neural texture). Achieving an intuitive way to edit abstract feature-based representations does not seem straightforward, but it is certainly worth considering how to set up neural rendering architectures to allow artists to edit as many parts of the pipeline as possible. Moreover, it is important to understand and reason about the network output as well. Even if explicit control may not be available in some cases, it may be useful to reason about failure cases.

Multimodal Neural Scene Representations. This report primarily focuses on rendering applications and, as such, most of the applications we discuss revolve around using images and videos as inputs and outputs to a network. A few of these applications also incorporate sound, for example to enable lip synchronization in a video clip when the audio is edited [FTZ*19]. Yet, a network that uses both visual and audio as input may learn useful ways to process the additional input modalities. Similarly, immersive virtual and augmented reality experiences and other applications may demand multimodal output of a neural rendering algorithm that incorporates spatial audio, tactile and haptic experiences, or perhaps olfactory signals. Extending neural rendering techniques to include other senses could be a fruitful direction of future research.

8. Social Implications

In this paper, we present a multitude of neural rendering approaches, with various applications and target domains. While some applications are mostly irreproachable, others, while having legitimate and extremely useful use cases, can also be used in a nefarious manner (e.g., talking-head synthesis). Methods for image and video manipulation are as old as the media themselves, and are common, for example, in the movie industry. However, neural rendering approaches have the potential to lower the barrier for entry, making manipulation technology accessible to non-experts with limited resources. While we believe that all the methods discussed in this paper were developed with the best of intentions, and indeed have the potential to positively influence the world via better communication, content creation and storytelling, we must not be complacent. It is important to proactively discuss and devise a plan to limit misuse. We believe it is critical that synthesized images and videos clearly present themselves as synthetic. We also believe that it is essential to obtain permission from the content owner and/or performers for any alteration before sharing a resulting video. Also, it is important that we as a community continue to develop forensics, fingerprinting and verification techniques (digital and non-digital) to identify manipulated video (Section 8.1). Such safeguarding measures would reduce the potential for misuse while allowing creative uses of video editing technologies. Researchers must also employ responsible disclosure when appropriate, carefully considering how and to whom a new system is released. In one recent example in the field of natural language processing [RWC*19], the authors adopted a "staged release" approach [SBC*19], refraining from releasing the full model immediately, instead releasing increasingly more powerful versions of the implementation over a full year. The authors also partnered with security researchers and policymakers, granting early access to the full model. Learning from this example, we believe researchers must make disclosure strategies a key part of any system with a potential for misuse, and not an afterthought. We hope that repeated demonstrations of image and video synthesis will teach people to think more critically about the media they consume, especially if there is no proof of origin. We also hope that publication of the details of such systems can spread awareness and knowledge regarding their inner workings, sparking and enabling associated research into the aforementioned forgery detection, watermarking and verification systems. Finally, we believe that a robust public conversation is necessary to create a set of appropriate regulations and laws that would balance the risks of misuse of these tools against the importance of creative, consensual use cases. For an in-depth analysis of security and safety considerations of AI systems, we refer the reader to [BAC*18]. While most measures described in this section involve law, policy and educational efforts, one measure — media forensics — is a *technical* challenge, as we describe next.

8.1. Forgery Detection

Integrity of digital content is of paramount importance nowadays. The verification of the integrity of an image can be done using a pro-active protection method, like digital signatures and watermarking, or a passive forensic analysis. An interesting concept is the 'Secure Digital Camera' [BF04] which not only introduces a watermark but also stores a biometric identifier of the person who took the photograph. While watermarking for forensic applications is explored in the literature, camera manufactures have so far failed to implement such methods in camera hardware [BF04, YP17, KBD15, KM18]. Thus, automatic passive detection of synthetic or manipulated imagery gains more and more importance. There is a large corpus of digital media forensic literature which splits up in manipulation-specific and manipulation-independent methods. *Manipulation-specific detection methods* learn to detect the artifacts produced by a specific manipulation method. FaceForensics++ [RCV*19] offers a large-scale dataset of different image synthesis and manipulation methods, suited to train deep neural networks in a supervised fashion. It is now the largest forensics dataset for detecting facial manipulations with over 4 million images. In addition, they show that they can train state-of-the-art neural networks to achieve high detection rates even under different level of image compression. Similar, Wang et al. [WWO*19] scripted photoshop to later detect photoshopped faces. The disadvantage of such manipulation-specific detection methods is the need of a large-scale training corpus per manipulation method. In ForensicTransfer [CTR*18], the authors propose a few-shot learning approach. Based on a few samples of a previously unseen manipulation method, high detection rates can be achieved even without a large (labeled) training corpus. In the scenario where no knowledge or samples of a manipulation method (i.e., "in the wild" manipulations) are available, *manipulation-independent methods* are required. These approaches concentrate on image plausibility. Physical and statistical information have to be consistent all over the image or video (e.g., shadows or JPEG compression). This consistency can for example be determined in a patch-based fashion [CV18, HLOE18], where one patch is compared to another part of the image. Using such a strategy, a detector can be trained on real data only using patches of different images as negative samples.

9. Conclusion

Neural rendering has raised a lot of interest in the past few years. This state-of-the-art report reflects the immense increase of research in this field. It is not bound to a specific application but spans a variety of use-cases that range from novel-view synthesis, semantic image editing, free viewpoint videos, relighting, face and body reenactment to digital avatars. Neural rendering has already enabled applications that were previously intractable, such as rendering of digital avatars without any manual modeling. We believe that neural rendering will have a profound impact in making complex photo and video editing tasks accessible to a much broader audience. We hope that this survey will introduce neural rendering to a large research community, which in turn will help to develop the next generation of neural rendering and graphics applications.

Acknowledgements G.W. was supported by an Okawa Research Grant, a Sloan Fellowship, NSF Awards IIS 1553333 and CMMI 1839974, and a PECASE by the ARL. V.S. was supported by a Stanford Graduate Fellowship. C.T. was supported by the ERC Consolidator Grant 4DRepLy (770784). M.N. was supported by Google, Sony, a TUM-IAS Rudolf Mößbauer Fellowship, the ERC Starting Grant Scan2CAD (804724), and a Google Faculty Award. M.A. and O.F. were supported by the Brown Institute for Media Innovation. We thank David Bau, Richard Zhang, Taesung Park, and Phillip Isola for proofreading.

References

[AHS85] ACKLEY D. H., HINTON G. E., SEJNOWSKI T. J.: A learning algorithm for boltzmann machines. *Cognitive science 9*, 1 (1985), 147–169. 4

[ARS*18] ALMAHAIRI A., RAJESWAR S., SORDONI A., BACHMAN P., COURVILLE A.: Augmented cyclegan: Learning many-to-many mappings from unpaired data. In *arXiv preprint arXiv:1802.10151* (2018). 6

[ASA18] ASIM M., SHAMSHAD F., AHMED A.: Blind image deconvolution using deep generative priors. *arXiv preprint arXiv:1802.04073* (2018). 10

[ASL*19] ABERMAN K., SHI M., LIAO J., LISCHINSKI D., CHEN B., COHEN-OR D.: Deep video-based performance cloning. *Comput. Graph. Forum 38*, 2 (2019), 219–233. URL: https://doi.org/10.1111/cgf.13632, doi:10.1111/cgf.13632. 7, 8, 9, 18, 19

[AUL19] ALIEV K.-A., ULYANOV D., LEMPITSKY V.: Neural point-based graphics. *arXiv* (2019). 7, 8, 9, 17

[AW19] ASHUAL O., WOLF L.: Specifying object attributes and relations in interactive scene generation. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019). 5

[BA83] BURT P., ADELSON E.: The laplacian pyramid as a compact image code. *IEEE Transactions on communications 31*, 4 (1983), 532–540. 10

[BAC*18] BRUNDAGE M., AVIN S., CLARK J., TONER H., ECKERSLEY P., GARFINKEL B., DAFOE A., SCHARRE P., ZEITZOFF T., FILAR B., ET AL.: The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. *arXiv preprint arXiv:1802.07228* (2018). 20

[BBPV03] BLANZ V., BASSO C., POGGIO T., VETTER T.: Reanimating faces in images and video. In *Computer graphics forum* (2003), vol. 22, Wiley Online Library, pp. 641–650. 17

[BDS19] BROCK A., DONAHUE J., SIMONYAN K.: Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)* (2019). 2

[BF04] BLYTHE P. A., FRIDRICH J. J.: Secure digital camera. 20

[BL*03] BROWN M., LOWE D. G., ET AL.: Recognising panoramas. In *International Conference on Computer Vision (ICCV)* (2003). 10

[BLRW17] BROCK A., LIM T., RITCHIE J. M., WESTON N.: Neural photo editing with introspective adversarial networks. In *International Conference on Learning Representations (ICLR)* (2017). 7, 8, 9, 10

[BM14] BARRON J. T., MALIK J.: Shape, illumination, and reflectance from shading. *IEEE transactions on pattern analysis and machine intelligence 37*, 8 (2014), 1670–1687. 16

[BM18] BLAU Y., MICHAELI T.: The perception-distortion tradeoff. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 4, 5

[BMRS18] BANSAL A., MA S., RAMANAN D., SHEIKH Y.: Recycle-gan: Unsupervised video retargeting. In *European Conference on Computer Vision (ECCV)* (2018), pp. 119–135. 10

[BSD*17] BOUSMALIS K., SILBERMAN N., DOHAN D., ERHAN D., KRISHNAN D.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 6

[BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D. B.: Patchmatch: A randomized correspondence algorithm for structural image editing. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2009). 4, 8

[BSP*19a] BAU D., STROBELT H., PEEBLES W., WULFF J., ZHOU B., ZHU J., TORRALBA A.: Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH) 38*, 4 (2019). 7, 8, 9, 10

[BSP*19b] BI S., SUNKAVALLI K., PERAZZI F., SHECHTMAN E., KIM V., RAMAMOORTHI R.: Deep cg2real: Synthetic-to-real translation via image disentanglement. In *International Conference on Computer Vision (ICCV)* (2019). 5, 11

[BSR19] BANSAL A., SHEIKH Y., RAMANAN D.: Shapes and context: In-the-wild image synthesis & manipulation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 8, 10

[BUS18] BASHKIROVA D., USMAN B., SAENKO K.: Unsupervised video-to-video translation. *arXiv preprint arXiv:1806.03698* (2018). 10

[BW18] BENAIM S., WOLF L.: One-shot unsupervised cross domain translation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2018), NIPS'18, Curran Associates Inc., p. 2108–2118. 10

[BZD*18] BALAKRISHNAN G., ZHAO A., DALCA A. V., DURAND F., GUTTAG J. V.: Synthesizing images of humans in unseen poses. *CVPR* (2018). 15

[BZS*19] BAU D., ZHU J.-Y., STROBELT H., ZHOU B., TENENBAUM J. B., FREEMAN W. T., TORRALBA A.: Gan dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations (ICLR)* (2019). 10

[CCK*18] CHOI Y., CHOI M., KIM M., HA J., KIM S., CHOO J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings - 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018* (12 2018), Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, pp. 8789–8797. doi:10.1109/CVPR.2018.00916. 6

[CCS*15] COLLET A., CHUANG M., SWEENEY P., GILLETT D., EVSEEV D., CALABRESE D., HOPPE H., KIRK A., SULLIVAN S.: High-quality streamable free-viewpoint video. *ACM TOG* (2015). 14

[CCT*09] CHEN T., CHENG M.-M., TAN P., SHAMIR A., HU S.-M.: Sketch2photo: internet image montage. *ACM Transactions on Graphics (TOG) 28*, 5 (2009), 124. 8

[CDS19] CLARK A., DONAHUE J., SIMONYAN K.: Efficient video generation on complex datasets. *ArXiv abs/1907.06571* (2019). 2

[CDSHD13] CHAURASIA G., DUCHENE S., SORKINE-HORNUNG O., DRETTAKIS G.: Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph. 32*, 3 (July 2013), 30:1–30:12. URL: http://doi.acm.org/10.1145/2487228.2487238, doi:10.1145/2487228.2487238. 11

[CGZE18] CHAN C., GINOSAR S., ZHOU T., EFROS A. A.: Everybody dance now. *CoRR abs/1808.07371* (2018). URL: http://arxiv.org/abs/1808.07371, arXiv:1808.07371. 7, 8, 9, 15, 18, 19

[CK17] CHEN Q., KOLTUN V.: Photographic image synthesis with cascaded refinement networks. In *International Conference on Computer Vision (ICCV)* (2017). 5, 7, 8, 9, 10

[CKS*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Trans. Graph. 36*, 4 (July 2017), 98:1–98:12. URL: http://doi.acm.org/10.1145/3072959.3073601, doi:10.1145/3072959.3073601. 2

[CNZ18] CHUNG J. S., NAGRANI A., ZISSERMAN A.: Voxceleb2: Deep speaker recognition. In *INTERSPEECH* (2018). 18

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Trans. Graph. 1*, 1 (January 1982), 7–24. 3

[CTR*18] COZZOLINO D., THIES J., RÖSSLER A., RIESS C., NIESSNER M., VERDOLIVA L.: Forensictransfer: Weakly-supervised domain adaptation for forgery detection. *arXiv* (2018). 20

[CV18] COZZOLINO D., VERDOLIVA L.: Noiseprint: A cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security 15* (2018), 144–159. 20

[CVPa] https://sites.google.com/view/cvpr2018tutorialongans/. 2

[CVPb] https://augmentedperception.github.io/cvpr18/. 2

[CVPc] https://augmentedperception.github.io/cvpr19/. 2

[CVPd] https://3dscenegen.github.io/. 2

[CWD*18] CRESWELL A., WHITE T., DUMOULIN V., ARULKUMARAN K., SENGUPTA B., BHARATH A. A.: Generative adversarial networks: An overview. *IEEE Signal Processing Magazine 35*, 1 (2018), 53–65. 2

[CXG*16] CHOY C. B., XU D., GWAK J., CHEN K., SAVARESE S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)* (2016). 2, 13

[DAD*18] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 128. 4

[DAD*19] DESCHAINTRE V., AITTALA M., DURAND F., DRETTAKIS G., BOUSSEAU A.: Flexible svbrdf capture with a multi-image deep network. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering) 38*, 4 (July 2019). URL: http://www-sop.inria.fr/reves/Basilic/2019/DADDB19. 4

[DB16] DOSOVITSKIY A., BROX T.: Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems* (2016). 4, 5

[DDF*17] DOU M., DAVIDSON P., FANELLO S. R., KHAMIS S., KOWDLE A., RHEMANN C., TANKOVICH V., IZADI S.: Motion2fusion: Real-time volumetric performance capture. *SIGGRAPH Asia* (2017). 14

[DHT*00] DEBEVEC P., HAWKINS T., TCHOU C., DUIKER H.-P., SAROKIN W., SAGAR M.: Acquiring the reflectance field of a human face. In *SIGGRAPH* (2000). 14, 15, 16

[DKD*16] DOU M., KHAMIS S., DEGTYAREV Y., DAVIDSON P., FANELLO S. R., KOWDLE A., ESCOLANO S. O., RHEMANN C., KIM D., TAYLOR J., KOHLI P., TANKOVICH V., IZADI S.: Fusion4d: Real-time performance capture of challenging scenes. *SIGGRAPH* (2016). 14

[Doe16] DOERSCH C.: Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908* (2016). 2, 6

[DSDB17] DINH L., SOHL-DICKSTEIN J., BENGIO S.: Density estimation using real nvp. In *International Conference on Learning Representations (ICLR)* (2017). 6

[DTSB15] DOSOVITSKIY A., TOBIAS SPRINGENBERG J., BROX T.: Learning to generate chairs with convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 4

[DYB98] DEBEVEC P., YU Y., BOSHOKOV G.: Efficient view-dependent IBR with projective texture-mapping. EG Rendering Workshop. 4, 11

[ECCa] https://augmentedperception.github.io/eccv18/. 2

[ECCb] https://sites.google.com/view/eccv2018tutorialonfacetracking. 2

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2001). 8

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *ICCV* (1999). 8

[ERB*18] ESLAMI S. A., REZENDE D. J., BESSE F., VIOLA F., MORCOS A. S., GARNELO M., RUDERMAN A., RUSU A. A., DANIHELKA I., GREGOR K., ET AL.: Neural scene representation and rendering. *Science 360*, 6394 (2018), 1204–1210. 2, 7, 8, 9, 11, 12

[ESO18] ESSER P., SUTTER E., OMMER B.: A variational u-net for conditional appearance and shape generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 18

[EST*19] EGGER B., SMITH W. A. P., TEWARI A., WUHRER S., ZOLLHOEFER M., BEELER T., BERNARD F., BOLKART T., KORTYLEWSKI A., ROMDHANI S., THEOBALT C., BLANZ V., VETTER T.: 3d morphable face models – past, present and future, 2019. arXiv:1909.01815. 2

[FBD*19] FLYNN J., BROXTON M., DEBEVEC P., DUVALL M., FYFFE G., OVERBECK R. S., SNAVELY N., TUCKER R.: Deepview: High-quality view synthesis by learned gradient descent. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). URL: https://arxiv.org/abs/1906.07316. 11, 12

[FNPS16] FLYNN J., NEULANDER I., PHILBIN J., SNAVELY N.: Deepstereo: Learning to predict new views from the world's imagery. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 12

[FTZ*19] FRIED O., TEWARI A., ZOLLHÖFER M., FINKELSTEIN A., SHECHTMAN E., GOLDMAN D. B., GENOVA K., JIN Z., THEOBALT C., AGRAWALA M.: Text-based editing of talking-head video. *ACM Trans. Graph. 38*, 4 (July 2019), 68:1–68:14. doi:10.1145/3306346.3323028. 1, 7, 8, 9, 11, 17, 19

[FWD09] FYFFE G., WILSON C. A., DEBEVEC P.: Cosine lobe based relighting from gradient illumination photographs. In *2009 Conference for Visual Media Production* (Nov 2009), pp. 100–108. doi:10.1109/CVMP.2009.18. 14, 16

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 4, 5, 10

[GGSC96] 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* (1996), SIGGRAPH '96. 4

[GLD*19] GUO K., LINCOLN P., DAVIDSON P., BUSCH J., YU X., WHALEN M., HARVEY G., ORTS-ESCOLANO S., PANDEY R., DOURGARIAN J., TANG D., TKACH A., KOWDLE A., COOPER E., DOU M., FANELLO S., FYFFE G., RHEMANN C., TAYLOR J., DEBEVEC P., IZADI S.: The relightables: Volumetric performance capture of humans with realistic relighting. *ACM Trans. Graph. 38*, 6 (November 2019). URL: https://doi.org/10.1145/3355089.3356571, doi:10.1145/3355089.3356571. 1, 7, 14

[GNK18] GÃIJLER R. A., NEVEROVA N., KOKKINOS I.: Densepose: Dense human pose estimation in the wild. In *CVPR* (2018). 19

[Goo16] GOODFELLOW I.: Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160* (2016). 2

[GPAM*14] GOODFELLOW I., POUGET-ABADIE J., MIRZA M., XU B., WARDE-FARLEY D., OZAIR S., COURVILLE A., BENGIO Y.: Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, Ghahramani Z., Welling M., Cortes C., Lawrence N. D., Weinberger K. Q., (Eds.). Curran Associates, Inc., 2014, pp. 2672–2680. 1, 4, 5

[GSZ*18] GENG J., SHAO T., ZHENG Y., WENG Y., ZHOU K.: Warp-guided gans for single-photo facial animation. *ACM Trans. Graph. 37*, 6 (December 2018), 231:1–231:12. URL: http://doi.acm.org/10.1145/3272127.3275043, doi:10.1145/3272127.3275043. 7, 17

[HAM19] HAINES E., AKENINE-MÖLLER T. (Eds.): *Ray Tracing Gems*. Apress, 2019. http://raytracinggems.com. 3

[HB17] HUANG X., BELONGIE S.: Arbitrary style transfer in real-time with adaptive instance normalization. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 10

[HE07] HAYS J., EFROS A. A.: Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG) 26*, 3 (2007), 4. 8

[HJO*01] HERTZMANN A., JACOBS C. E., OLIVER N., CURLESS B., SALESIN D. H.: Image analogies. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2001). 8

[HLBK18] HUANG X., LIU M.-Y., BELONGIE S., KAUTZ J.: Multimodal unsupervised image-to-image translation. *European Conference on Computer Vision (ECCV)* (2018). 6, 10

[HLOE18] HUH M., LIU A., OWENS A., EFROS A.: Fighting fake news: Image splice detection via learned self-consistency. In *ECCV* (2018). 20

[HMR19] HENZLER P., MITRA N., RITSCHEL T.: Escaping plato's cave using adversarial training: 3d shape from unstructured 2d image collections. 4, 13

[HPP*18] 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 (December 2018), 257:1–257:15. URL: http://doi.acm.org/10.1145/3272127.3275084, doi:10.1145/3272127.3275084. 4, 7, 8, 9, 11

[HRDB16] HEDMAN P., RITSCHEL T., DRETTAKIS G., BROSTOW G.: Scalable Inside-Out Image-Based Rendering. 231:1–231:11. 4

[HS06] HINTON G. E., SALAKHUTDINOV R. R.: Reducing the dimensionality of data with neural networks. *science 313*, 5786 (2006), 504–507. 4

[HST12] HE K., SUN J., TANG X.: Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence 35*, 6 (2012), 1397–1409. 10

[HTP*18] HOFFMAN J., TZENG E., PARK T., ZHU J.-Y., ISOLA P., SAENKO K., EFROS A. A., DARRELL T.: Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)* (2018). 11

[HYHL18] HONG S., YAN X., HUANG T. S., LEE H.: Learning hierarchical semantic image manipulation through structured representations. In *Advances in Neural Information Processing Systems* (2018). 10

[HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2016). 5

[IJC] https://ermongroup.github.io/generative-models/. 2, 6

[ISSI16] IIZUKA S., SIMO-SERRA E., ISHIKAWA H.: Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 110. 10

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 2, 4, 5, 6, 7, 8, 9, 10, 14

[JAFF16] JOHNSON J., ALAHI A., FEI-FEI L.: Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)* (2016). 4, 5

[JBS*06] JOHNSON M., BROSTOW G. J., SHOTTON J., ARANDJELOVIC O., KWATRA V., CIPOLLA R.: Semantic photo synthesis. *Computer Graphics Forum 25*, 3 (2006), 407–413. 8

[JDA*11] JOHNSON M. K., DALE K., AVIDAN S., PFISTER H., FREEMAN W. T., MATUSIK W.: Cg2real: Improving the realism of computer generated images using a large collection of photographs. *IEEE Trans. Vis. Comput. Graph. 17* (2011), 1273–1285. 11

[JREM*16] JIMENEZ REZENDE D., ESLAMI S. M. A., MOHAMED S., BATTAGLIA P., JADERBERG M., HEESS N.: Unsupervised learning of 3d structure from images. In *Proc. NIPS*. 2016. 13

[KAEE16] KARACAN L., AKATA Z., ERDEM A., ERDEM E.: Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215* (2016). 7, 8, 9, 10

[Kaj86] KAJIYA J. T.: The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques* (1986), SIGGRAPH '86, pp. 143–150. 4

[KALL17] KARRAS T., AILA T., LAINE S., LEHTINEN J.: Progressive growing of gans for improved quality, stability, and variation. *ArXiv abs/1710.10196* (2017). 2, 5

[KBD15] KORUS P., BIALAS J., DZIECH A.: Towards practical self-embedding for jpeg-compressed digital images. *IEEE Transactions on Multimedia 17* (2015), 157–170. 20

[KBS15] KALANTARI N. K., BAKO S., SEN P.: A Machine Learning Approach for Filtering Monte Carlo Noise. *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2015) 34*, 4 (2015). 2

[KCK*17] KIM T., CHA M., KIM H., LEE J. K., KIM J.: Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)* (2017). 6

[KD18] KINGMA D. P., DHARIWAL P.: Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems* (2018). 6

[KGT*18] KIM H., GARRIDO P., TEWARI A., XU W., THIES J., NIESSNER M., PÉREZ P., RICHARDT C., ZOLLHÖFER M., THEOBALT C.: Deep video portraits. *ACM Trans. Graph. 37*, 4 (July 2018), 163:1–163:14. URL: http://doi.acm.org/10.1145/3197517.3201283, doi:10.1145/3197517.3201283. 7, 8, 9, 11, 17, 19

[KHM17] KAR A., HÄNE C., MALIK J.: Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems 30* (2017), pp. 365–376. 1, 13

[KLA19] KARRAS T., LAINE S., AILA T.: A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 4, 5

[KM18] KORUS P., MEMON N. D.: Content authentication for neural imaging pipelines: End-to-end optimization of photo provenance in complex distribution channels. In *CVPR* (2018). 20

[KRF*18] KOWDLE A., RHEMANN C., FANELLO S., TAGLIASACCHI A., TAYLOR J., DAVIDSON P., DOU M., GUO K., KESKIN C., KHAMIS S., KIM D., TANG D., TANKOVICH V., VALENTIN J., IZADI S.: The need 4 speed in real-time dense visual tracking. *SIGGRAPH Asia* (2018). 14

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (2012). 5

[KW13] KINGMA D. P., WELLING M.: Auto-encoding variational bayes. *CoRR abs/1312.6114* (2013). 6, 7

[KW19] KINGMA D. P., WELLING M.: An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691* (2019). 2, 6

[LBK17] LIU M.-Y., BREUEL T., KAUTZ J.: Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems* (2017). 6, 10

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1987), SIGGRAPH 1987, ACM, pp. 163–169. URL: http://doi.acm.org/10.1145/37401.37422, doi:10.1145/37401.37422. 3

[LHE*07] LALONDE J.-F., HOIEM D., EFROS A. A., ROTHER C., WINN J., CRIMINISI A.: Photo clip art. In *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)* (2007). 8, 10

[LHM*19] LIU M.-Y., HUANG X., MALLYA A., KARRAS T., AILA T., LEHTINEN J., KAUTZ J.: Few-shot unsupervised image-to-image translation. In *International Conference on Computer Vision (ICCV)* (2019). 10, 18

[LSC18] LI Z., SUNKAVALLI K., CHANDRAKER M.: Materials for masses: Svbrdf acquisition with a single mobile phone image. In *ECCV* (2018), pp. 72–87.

[LSD15] LONG J., SHELHAMER E., DARRELL T.: Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2015). 5, 6, 8, 10

[LSLW16] LARSEN A. B. L., SØNDERBY S. K., LAROCHELLE H., WINTHER O.: Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of the International Conference on Machine Learning (ICML)* (2016). 7, 10

[LSS*19] 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 (July 2019), 65:1–65:14. URL: http://doi.acm.org/10.1145/3306346.3323020, doi:10.1145/3306346.3323020. 7, 8, 9, 14, 15

[LSSS18] LOMBARDI S., SARAGIH J., SIMON T., SHEIKH Y.: Deep appearance models for face rendering. *ACM Trans. Graph. 37*, 4 (July 2018), 68:1–68:13. URL: http://doi.acm.org/10.1145/3197517.3201401, doi:10.1145/3197517.3201401. 7, 8, 9, 14, 18

[LT16] LIU M.-Y., TUZEL O.: Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems* (2016). 6

[LTH*17] LEDIG C., THEIS L., HUSZÁR F., CABALLERO J., CUNNINGHAM A., ACOSTA A., AITKEN A., TEJANI A., TOTZ J., WANG Z., ET AL.: Photo-realistic single image super-resolution using a generative adversarial network. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 5

[LXR*18] LI Z., XU Z., RAMAMOORTHI R., SUNKAVALLI K., CHANDRAKER M.: Learning to reconstruct shape and spatially-varying reflectance from a single image. In *SIGGRAPH Asia 2018 Technical Papers* (2018), ACM, p. 269. 4

[LXZ*19] LIU L., XU W., ZOLLHÖFER M., KIM H., BERNARD F., HABERMANN M., WANG W., THEOBALT C.: Neural rendering and reenactment of human actor videos. *ACM Trans. Graph. 38*, 5 (October 2019). URL: https://doi.org/10.1145/3333002, doi:10.1145/3333002. 1, 7, 8, 9, 18, 19

[Mar98] MARSCHNER S. R.: *Inverse rendering for computer graphics*. Citeseer, 1998. 4

[MBLD92] MATAN O., BURGES C. J., LECUN Y., DENKER J. S.: Multi-digit recognition using a space displacement neural network. In *Advances in Neural Information Processing Systems* (1992). 5

[MBPY*18] MARTIN-BRUALLA R., PANDEY R., YANG S., PIDLYPENSKYI P., TAYLOR J., VALENTIN J., KHAMIS S., DAVIDSON P., TKACH A., LINCOLN P., KOWDLE A., RHEMANN C., GOLDMAN D. B., KESKIN C., SEITZ S., IZADI S., FANELLO S.: Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph. 37*, 6 (December 2018), 255:1–255:14. URL: http://doi.acm.org/10.1145/3272127.3275099, doi:10.1145/3272127.3275099. 1, 7, 9, 14, 15

[MBS*18] MUELLER F., BERNARD F., SOTNYCHENKO O., MEHTA D., SRIDHAR S., CASAS D., THEOBALT C.: Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)* (June 2018). URL: https://handtracker.mpi-inf.mpg.de/projects/GANeratedHands/. 11

[MEA09] MATHIAS EITZ KRISTIAN HILDEBRAND T. B., ALEXA M.: Photosketch: A sketch based image query and compositing system. In *ACM SIGGRAPH 2009 Talk Program* (2009). 8

[MGK*19] MESHRY M., GOLDMAN D. B., KHAMIS S., HOPPE H., PANDEY R., SNAVELY N., MARTIN-BRUALLA R.: Neural rerendering in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 1, 7, 8, 9, 11, 12

[MHP*19] MEKA A., HÄNE C., PANDEY R., ZOLLHÖFER M., FANELLO S., FYFFE G., KOWDLE A., YU X., BUSCH J., DOURGARIAN J., DENNY P., BOUAZIZ S., LINCOLN P., WHALEN M., HARVEY G., TAYLOR J., IZADI S., TAGLIASACCHI A., DEBEVEC P., THEOBALT C., VALENTIN J., RHEMANN C.: Deep reflectance fields: High-quality facial reflectance field inference from color gradient illumination. *ACM Trans. Graph. 38*, 4 (July 2019). URL: https://doi.org/10.1145/3306346.3323027, doi:10.1145/3306346.3323027. 7, 9, 16

[MJS*17] MA L., JIA X., SUN Q., SCHIELE B., TUYTELAARS T., VAN GOOL L.: Pose guided person image generation. In *NIPS* (2017). 15, 18

[MLTFR19] MAXIMOV M., LEAL-TAIXE L., FRITZ M., RITSCHEL T.: Deep appearance maps. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019). 4

[MM14] MCGUIRE M., MARA M.: Efficient GPU screen-space ray tracing. *Journal of Computer Graphics Techniques (JCGT) 3*, 4 (December 2014), 73–85. URL: http://jcgt.org/published/0003/04/04/. 3

[MO14] MIRZA M., OSINDERO S.: Conditional generative adversarial nets. arXiv:1411.1784, 2014. URL: https://arxiv.org/abs/1411.1784. 5, 6, 10

[MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3d reconstruction in function space. In *CVPR* (2019), pp. 4460–4470. 2, 13

[MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Trans. Graph. 22*, 3 (July 2003), 759–769. 3

[MSG*18] MA L., SUN Q., GEORGOULIS S., GOOL L. V., SCHIELE B., FRITZ M.: Disentangled person image generation. *CVPR* (2018). 15

[Mül66] MÜLLER C.: *Spherical harmonics*. Springer, 1966. 3

[NAM*17] NALBACH O., ARABADZHIYSKA E., MEHTA D., SEIDEL H.-P., RITSCHEL T.: Deep shading: Convolutional neural networks for screen space shading. *Comput. Graph. Forum 36*, 4 (July 2017), 65–78. URL: https://doi.org/10.1111/cgf.13225, doi:10.1111/cgf.13225. 11

[NGK18] NEVEROVA N., GÜLER R. A., KOKKINOS I.: Dense pose transfer. *ECCV* (2018). 15

[NLT*19] NGUYEN PHUOC T., LI C., THEIS L., RICHARDT C., YANG Y.: Hologan: Unsupervised learning of 3d representations from natural images. International Conference on Computer Vision 2019 ; Conference date: 27-10-2019 Through 02-11-2019. 7, 8, 9, 11, 13, 19

[NPLBY18] NGUYEN-PHUOC T. H., LI C., BALABAN S., YANG Y.: Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. In *NIPS* (2018), pp. 7891–7901. 7, 8, 9, 13, 19

[NSX*18] NAGANO K., SEO J., XING J., WEI L., LI Z., SAITO S., AGARWAL A., FURSUND J., LI H., ROBERTS R., ET AL.: pagan: real-time avatars using dynamic textures. *ACM Trans. Graph. 37*, 6 (2018), 258–1. 17

[ODZ*16] OORD A. V. D., DIELEMAN S., ZEN H., SIMONYAN K., VINYALS O., GRAVES A., KALCHBRENNER N., SENIOR A., KAVUKCUOGLU K.: Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016). 6

[OE18] OUSSIDI A., ELHASSOUNY A.: Deep generative models: Survey. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)* (April 2018), pp. 1–8. doi:10.1109/ISACV.2018.8354080. 2

[OERF*16] ORTS-ESCOLANO S., RHEMANN C., FANELLO S., CHANG W., KOWDLE A., DEGTYAREV Y., KIM D., DAVIDSON P. L., KHAMIS S., DOU M., TANKOVICH V., LOOP C., CAI Q., CHOU P. A., MENNICKEN S., VALENTIN J., PRADEEP V., WANG S., KANG S. B., KOHLI P., LUTCHYN Y., KESKIN C., IZADI S.: Holoportation: Virtual 3d teleportation in real-time. In *UIST* (2016). 14

[OKK16] OORD A. V. D., KALCHBRENNER N., KAVUKCUOGLU K.: Pixel recurrent neural networks. 6

[OKV*16] OORD A. V. D., KALCHBRENNER N., VINYALS O., ESPEHOLT L., GRAVES A., KAVUKCUOGLU K.: Conditional image generation with pixelcnn decoders. In *Proc. NIPS* (2016), pp. 4797–4805. 6

[ON95] OREN M., NAYAR S. K.: Generalization of the lambertian model and implications for machine vision. *Int. J. Comput. Vision 14*, 3 (April 1995), 227–251. 3

[ope] https://openai.com/blog/generative-models/. 2, 6

[Ou18] OU Z.: A review of learning with deep generative models from perspective of graphical modeling, 2018. arXiv:1808.01630. 2

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR* (2019), pp. 165–174. 2, 13, 14

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on graphics (TOG) 22*, 3 (2003), 313–318. 8

[PGZ*19] PHILIP J., GHARBI M., ZHOU T., EFROS A. A., DRETTAKIS G.: Multi-view relighting using a geometry-aware network. *ACM Trans. Graph. 38*, 4 (July 2019), 78:1–78:14. URL: http://doi.acm.org/10.1145/3306346.3323013, doi:10.1145/3306346.3323013. 7, 9, 16

[Pho75] PHONG B. T.: Illumination for computer generated pictures. *Commun. ACM 18*, 6 (June 1975), 311–317. 3

[PKKS19] PITTALUGA F., KOPPAL S. J., KANG S. B., SINHA S. N.: Revealing scenes by inverting structure from motion reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 145–154. 12

[PLWZ19a] PARK T., LIU M.-Y., WANG T.-C., ZHU J.-Y.: Gaugan: Semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!* (2019). 8, 10

[PLWZ19b] PARK T., LIU M.-Y., WANG T.-C., ZHU J.-Y.: Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019). 5, 7, 8, 9, 10

[PTY*19] PANDEY R., TKACH A., YANG S., PIDLYPENSKYI P., TAYLOR J., MARTIN-BRUALLA R., TAGLIASACCHI A., PAPANDREOU G., DAVIDSON P., KESKIN C., IZADI S., FANELLO S.: Volumetric capture of humans with a single rgbd camera via semi-parametric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019). 9, 14, 15, 19

[PVDWRÁ16] PERARNAU G., VAN DE WEIJER J., RADUCANU B., ÁLVAREZ J. M.: Invertible conditional gans for image editing. *NIPS 2016 Workshop on Adversarial Training* (2016). 10

[PYY*19] PAN Z., YU W., YI X., KHAN A., YUAN F., ZHENG Y.: Recent progress on generative adversarial networks (gans): A survey. *IEEE Access 7* (2019), 36322–36333. doi:10.1109/ACCESS.2019.2905015. 2

[QCJK18] QI X., CHEN Q., JIA J., KOLTUN V.: Semi-parametric image synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 10

[QSMG16] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593* (2016). 2

[RCV*19] RÖSSLER A., COZZOLINO D., VERDOLIVA L., RIESS C., THIES J., NIESSNER M.: Faceforensics++: Learning to detect manipulated facial images. In *ICCV 2019* (2019). 20

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (Cham, 2015), Navab N., Hornegger J., Wells W. M., Frangi A. F., (Eds.), Springer International Publishing, pp. 234–241. 5, 6

[RJGW19] RAINER G., JAKOB W., GHOSH A., WEYRICH T.: Neural BTF compression and interpolation. *Computer Graphics Forum (Proc. Eurographics) 38*, 2 (2019), 1–10. 4

[RMC16] RADFORD A., METZ L., CHINTALA S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)* (2016). 2

[RWC*19] RADFORD A., WU J., CHILD R., LUAN D., AMODEI D., SUTSKEVER I.: Language models are unsupervised multitask learners. *OpenAI Blog 1*, 8 (2019). 20

[Sal15] SALAKHUTDINOV R.: Learning deep generative models. *Annual Review of Statistics and Its Application 2* (2015), 361–385. 2

[SBC*19] SOLAIMAN I., BRUNDAGE M., CLARK J., ASKELL A., HERBERT-VOSS A., WU J., RADFORD A., WANG J.: Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203* (2019). 20

[SBT*19] SUN T., BARRON J. T., TSAI Y.-T., XU Z., YU X., FYFFE G., RHEMANN C., BUSCH J., DEBEVEC P., RAMAMOORTHI R.: Single image portrait relighting. *ACM Trans. Graph. 38*, 4 (July 2019). URL: https://doi.org/10.1145/3306346.3323008, doi:10.1145/3306346.3323008. 1, 7, 9, 16

[SCSI08] SIMAKOV D., CASPI Y., SHECHTMAN E., IRANI M.: Summarizing visual data using bidirectional similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008* (june 2008), pp. 1–8. 8

[SDM19] SHAHAM T. R., DEKEL T., MICHAELI T.: Singan: Learning a generative model from a single natural image. In *The IEEE International Conference on Computer Vision (ICCV)* (October 2019). 5

[SH09] SALAKHUTDINOV R., HINTON G.: Deep boltzmann machines. In *Artificial intelligence and statistics* (2009), pp. 448–455. 4

[SHN*19] SAITO S., HUANG Z., NATSUME R., MORISHIMA S., KANAZAWA A., LI H.: Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *ICCV* (2019). 2, 11, 13

[SK00] SHUM H., KANG S. B.: Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000* (2000), vol. 4067, International Society for Optics and Photonics, pp. 2–13. 2

[SKCJ18] SENGUPTA S., KANAZAWA A., CASTILLO C. D., JACOBS D. W.: Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 6296–6305. 16

[SLF*17] SANGKLOY P., LU J., FANG C., YU F., HAYS J.: Scribbler: Controlling deep image synthesis with sketch and color. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 10

[SPT*17] SHRIVASTAVA A., PFISTER T., TUZEL O., SUSSKIND J., WANG W., WEBB R.: Learning from simulated and unsupervised images through adversarial training. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 6, 11

[SRT*11] STURM P., RAMALINGAM S., TARDIF J.-P., GASPARINI S., BARRETO J. A.: Camera models and fundamental concepts used in geometric computer vision. *Found. Trends. Comput. Graph. Vis. 6* (January 2011), 1–183. 3

[SSLS18] SIAROHIN A., SANGINETO E., LATHUILIERE S., SEBE N.: Deformable GANs for pose-based human image generation. In *CVPR 2018* (2018). 18

[Sta] https://deepgenerativemodels.github.io/. 2, 6

[STH*19] SITZMANN V., THIES J., HEIDE F., NIESSNER M., WETZSTEIN G., ZOLLHÖFER M.: Deepvoxels: Learning persistent 3d feature embeddings. In *CVPR* (2019). 7, 8, 9, 13

[SWWT18] SI C., WANG W., WANG L., TAN T.: Multistage adversarial losses for pose-based human image synthesis. In *CVPR* (2018). 15

[SYH*17] SHU Z., YUMER E., HADAP S., SUNKAVALLI K., SHECHTMAN E., SAMARAS D.: Neural face editing with intrinsic image disentangling. In *Computer Vision and Pattern Recognition, 2017. CVPR 2017. IEEE Conference on* (2017), IEEE, pp. –. 7, 16

[SZ15] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)* (2015). 5

[SZA*19] SHYSHEYA A., ZAKHAROV E., ALIEV K.-A., BASHIROV R. S., BURKOV E., ISKAKOV K., IVAKHNENKO A., MALKOV Y.,

PASECHNIK I. M., ULYANOV D., VAKHITOV A., LEMPITSKY V. S.: Textured neural avatars. *CVPR* (2019). 7, 9, 18

[Sze10] SZELISKI R.: *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2

[SZUL18] SUNGATULLINA D., ZAKHAROV E., ULYANOV D., LEMPITSKY V.: Image manipulation with perceptual discriminators. In *European Conference on Computer Vision (ECCV)* (2018). 6

[SZW19] SITZMANN V., ZOLLHÖFER M., WETZSTEIN G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS* (2019). 1, 7, 8, 9, 11, 13, 14, 19

[TH00] TANCO L. M., HILTON A.: Realistic synthesis of novel human movements from a database of motion capture examples. In *Proceedings Workshop on Human Motion* (2000). 14

[TPW17] TAIGMAN Y., POLYAK A., WOLF L.: Unsupervised cross-domain image generation. In *International Conference on Learning Representations (ICLR)* (2017). 6

[TSF*18] TANKOVICH V., SCHOENBERG M., FANELLO S. R., KOWDLE A., RHEMANN C., DZITSIUK M., SCHMIDT M., VALENTIN J., IZADI S.: Sos: Stereo matching in o(1) with slanted support windows. *IROS* (2018). 14

[TZN19] THIES J., ZOLLHÖFER M., NIESSNER M.: Deferred neural rendering: Image synthesis using neural textures. *ACM Trans. Graph. 38*, 4 (July 2019), 66:1–66:12. URL: http://doi.acm.org/10.1145/3306346.3323035, doi:10.1145/3306346.3323035. 7, 9, 17

[TZS*16] THIES J., ZOLLHÖFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: Face2Face: Real-time Face Capture and Reenactment of RGB Videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE* (2016). 17

[TZS*18] THIES J., ZOLLHÖFER M., STAMMINGER M., THEOBALT C., NIESSNER M.: Headon: Real-time reenactment of human portrait videos. *ACM Transactions on Graphics 2018 (TOG)* (2018). 4

[TZT*20] THIES J., ZOLLHÖFER M., THEOBALT C., STAMMINGER M., NIESSNER M.: Image-guided neural object rendering. *ICLR 2020* (2020). 4, 7, 8, 9, 11

[UVL16] ULYANOV D., VEDALDI A., LEMPITSKY V.: Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022* (2016). 10

[UVL18] ULYANOV D., VEDALDI A., LEMPITSKY V.: Deep image prior. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 10

[Vea98] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford, CA, USA, 1998. AAI9837162. 4

[VPT16] VONDRICK C., PIRSIAVASH H., TORRALBA A.: Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems* (2016). 2

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing 13*, 4 (2004), 600–612. 5

[WDGH16] WALKER J., DOERSCH C., GUPTA A., HEBERT M.: An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision (ECCV)* (2016). 7

[Whi80] WHITTED T.: An improved illumination model for shaded display. *Commun. ACM 23*, 6 (June 1980), 343–349. URL: http://doi.acm.org/10.1145/358876.358882, doi:10.1145/358876.358882. 3

[WKZ18] WILES O., KOEPKE A., ZISSERMAN A.: X2face: A network for controlling face generation by using images, audio, and pose codes. In *European Conference on Computer Vision* (2018). 17

[WLZ*18a] WANG T.-C., LIU M.-Y., ZHU J.-Y., LIU G., TAO A., KAUTZ J., CATANZARO B.: Video-to-video synthesis. In *Advances in Neural Information Processing Systems* (2018). 10

[WLZ*18b] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 6, 7, 8, 9, 10

[WSB03] WANG Z., SIMONCELLI E. P., BOVIK A. C.: Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003* (2003), vol. 2, Ieee, pp. 1398–1402. 5

[WSI07] WEXLER Y., SHECHTMAN E., IRANI M.: Space-time completion of video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 3 (march 2007), 463–476. 8

[WSS*19] WEI S.-E., SARAGIH J., SIMON T., HARLEY A. W., LOMBARDI S., PERDOCH M., HYPES A., WANG D., BADINO H., SHEIKH Y.: Vr facial animation via multiview image translation. *ACM Trans. Graph. 38*, 4 (July 2019), 67:1–67:16. URL: http://doi.acm.org/10.1145/3306346.3323030, doi:10.1145/3306346.3323030. 1, 7, 8, 9, 18

[WSW19] WANG Z., SHE Q., WARD T. E.: Generative adversarial networks: A survey and taxonomy. *arXiv preprint arXiv:1906.01529* (2019). 2

[WWO*19] WANG S.-Y., WANG O., OWENS A., ZHANG R., EFROS A. A.: Detecting photoshopped faces by scripting photoshop. *arXiv preprint arXiv:1906.05856* (2019). 20

[WYDCL18] WANG X., YU K., DONG C., CHANGE LOY C.: Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 10

[XBS*19] XU Z., BI S., SUNKAVALLI K., HADAP S., SU H., RAMAMOORTHI R.: Deep view synthesis from sparse photometric images. *ACM Trans. Graph. 38*, 4 (July 2019), 76:1–76:13. URL: http://doi.acm.org/10.1145/3306346.3323007, doi:10.1145/3306346.3323007. 1, 7, 8, 9, 11, 12

[XSA*18] XIAN W., SANGKLOY P., AGRAWAL V., RAJ A., LU J., FANG C., YU F., HAYS J.: Texturegan: Controlling deep image synthesis with texture patches. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 10

[XSHR18] XU Z., SUNKAVALLI K., HADAP S., RAMAMOORTHI R.: Deep image-based relighting from optimal sparse samples. *ACM Trans. Graph. 37*, 4 (July 2018), 126:1–126:13. URL: http://doi.acm.org/10.1145/3197517.3201313, doi:10.1145/3197517.3201313. 1, 7, 9, 15, 16

[XWBF16] XUE T., WU J., BOUMAN K., FREEMAN B.: Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems* (2016). 7

[YCYL*17] YEH R. A., CHEN C., YIAN LIM T., SCHWING A. G., HASEGAWA-JOHNSON M., DO M. N.: Semantic image inpainting with deep generative models. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 10

[YHZ*18] YAO S., HSU T. M., ZHU J.-Y., WU J., TORRALBA A., FREEMAN B., TENENBAUM J.: 3d-aware scene manipulation via inverse graphics. In *Advances in Neural Information Processing Systems* (2018). 10

[YP17] YAN C.-P., PUN C.-M.: Multi-scale difference map fusion for tamper localization using binary ranking hashing. *IEEE Transactions on Information Forensics and Security 12* (2017), 2144–2158. 20

[YZTG17] YI Z., ZHANG H., TAN P., GONG M.: Dualgan: Unsupervised dual learning for image-to-image translation. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2017). 6, 10

[ZC04] ZHANG C., CHEN T.: A survey on image-based renderingâĂŤrepresentation, sampling and compression. *Signal Processing: Image Communication 19*, 1 (2004), 1–28. 2

[ZHSJ19] ZHOU H., HADAP S., SUNKAVALLI K., JACOBS D.: Deep single image portrait relighting. In *Proc. International Conference on Computer Vision (ICCV)* (2019). 7, 9, 16

[ZIE16] ZHANG R., ISOLA P., EFROS A. A.: Colorful image colorization. In *European Conference on Computer Vision (ECCV)* (2016). 5, 10

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 5

[ZKSE16] ZHU J.-Y., KRÄHENBÜHL P., SHECHTMAN E., EFROS A. A.: Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)* (2016). 7, 8, 9, 10

[ZKTF10] ZEILER M. D., KRISHNAN D., TAYLOR G. W., FERGUS R.: Deconvolutional networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)* (2010). 5

[ZPIE17] ZHU J.-Y., PARK T., ISOLA P., EFROS A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)* (2017). 2, 5, 6, 10, 11

[ZSBL19] ZAKHAROV E., SHYSHEYA A., BURKOV E., LEMPITSKY V.: Few-shot adversarial learning of realistic neural talking head models. *arXiv preprint arXiv:1905.08233* (2019). 7, 8, 9, 17

[ZTF*18] ZHOU T., TUCKER R., FLYNN J., FYFFE G., SNAVELY N.: Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. 37*, 4 (July 2018), 65:1–65:12. 12

[ZTG*18] ZOLLHÖFER M., THIES J., GARRIDO P., BRADLEY D., BEELER T., PÉREZ P., STAMMINGER M., NIESSNER M., THEOBALT C.: State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications. *Computer Graphics Forum (Eurographics State of the Art Reports 2018) 37*, 2 (2018). 2

[ZWC*17] ZHAO B., WU X., CHENG Z., LIU H., FENG J.: Multi-view image generation from a single-view. *CoRR* (2017). 15

[ZZP*17] ZHU J.-Y., ZHANG R., PATHAK D., DARRELL T., EFROS A. A., WANG O., SHECHTMAN E.: Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems* (2017). 6, 7, 10

[ZZZ*18] ZHU J.-Y., ZHANG Z., ZHANG C., WU J., TORRALBA A., TENENBAUM J., FREEMAN B.: Visual object networks: image generation with disentangled 3d representations. In *Proc. NIPS* (2018), pp. 118–129. 11, 13