Machine Intelligence Research
www.mi-research.net

20(6), December 2023, 799-821
DOI: 10.1007/s11633-022-1400-x

State of the Art on Deep Learning-enhanced Rendering Methods

Qi Wang[1] Zhihua Zhong[1] Yuchi Huo[2,1] Hujun Bao[1] Rui Wang[1]

[1] State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310058, China

[2] Zhejiang Laboratory, Hangzhou 311121, China

Abstract: Photorealistic rendering of the virtual world is an important and classic problem in the field of computer graphics. With the development of GPU hardware and continuous research on computer graphics, representing and rendering virtual scenes has become easier and more efficient. However, there are still unresolved challenges in efficiently rendering global illumination effects. At the same time, machine learning and computer vision provide real-world image analysis and synthesis methods, which can be exploited by computer graphics rendering pipelines. Deep learning-enhanced rendering combines techniques from deep learning and computer vision into the traditional graphics rendering pipeline to enhance existing rasterization or Monte Carlo integration renderers. This state-of-the-art report summarizes recent studies of deep learning-enhanced rendering in the computer graphics community. Specifically, we focus on works of renderers represented using neural networks, whether the scene is represented by neural networks or traditional scene files. These works are either for general scenes or specific scenes, which are differentiated by the need to retrain the network for new scenes.

Keywords: Neural rendering, computer graphics, scene representation, rendering, post-processing.

Citation: Q. Wang, Z. Zhong, Y. Huo, H. Bao, R. Wang. State of the art on deep learning-enhanced rendering methods. *Machine Intelligence Research*, vol.20, no.6, pp.799–821, 2023. http://doi.org/10.1007/s11633-022-1400-x

1 Introduction

For traditional computer graphics, generating photorealistic rendering results of a scene is an important research direction, and researchers have developed a variety of algorithms to solve this problem in recent decades, including modeling complex materials[1−3], sophisticated sampling methods[4, 5], acceleration of global illumination computation[6, 7], etc. These methods are mainly applied in two fields: the rasterization pipeline for real-time rendering that serves the latest video games and the ray tracing pipeline commonly used in the film industry for offline rendering that deals with global illumination effects. Regardless of the rendering pipeline, much time-consuming manual work by artists and programmers is essential, i.e., complex and sophisticated renderers or shaders must be written by experienced programmers, and scene construction, including geometry, material textures, lighting conditions, and animations are the responsibility of artists. These preliminary preparations greatly increase the time and capital cost of photorealistic rendering. However, with the rapid development of computer vision and deep learning research fields in recent years, the combination of traditional graphics rendering pipelines and deep learning provides a new direction to solve the above problems. A bunch of deep generative models of generating high-resolution styled[8] or high-fidelity 2D images have emerged, e.g., the seminal generative adversarial neural networks (GANs)[9] and their follow-ups[10, 11], also the variational auto encoder networks (VAEs)[12−14]. Reference [15] even realizes the control of the generated image through additional condition input.

Armed with powerful neural network-based image generation tools, researchers consider how to represent traditional scenes as data types that neural networks can handle and feed into generative networks to render scenes. The first seminal method that combines a deep neural network and a traditional rendering pipeline is the generative query network (GQN)[16]. The network takes several rendered images and the corresponding camera parameters as input to encode the complete scene information as a vector, and the vector is fed to a generative network to enable the rendering of the scene from any viewpoint (Section 4.4). Although the rendering results generated by their method are not realistic enough, they inspire a vast amount of subsequent work and create a new field of research: neural rendering. Compared with other deep learning research fields, the focus of neural rendering is not only on the delicate network structure design but also on the combination of physical and mathematical knowledge in traditional rendering. In fact, for a

Review
Manuscript received on July 11, 2022; accepted on November 29, 2022
Recommended by Associate Editor Jing-Yi Yu
Colored figures are available in the online version at https://link.springer.com/journal/11633

specific rendering task, the difficulty lies in embedding the corresponding domain knowledge into the network, e.g., for human skin rendering, how to embed the subsurface scattering process into the network to allow the neural generator to generate more realistic skin effects. Compared with the traditional rendering pipeline, the rendering quality of neural rendering is closely related to the quantity, distribution, and quality of the input dataset. Thus, how to render high-quality results with insufficient data is also an important concern of neural rendering.

This state-of-the-art report (STAR) summarizes and classifies the different types of deep learning-enhanced rendering approaches. It should be noted that our work is different from another review on neural rendering[17]. We only focus on approaches that are integrated into the traditional rendering pipelines with neural networks, i.e., the forwards subset of neural rendering which assumes known input scenes (geometry, lighting, material, viewpoint) and does not concern the specific representation. However, the concept of "rendering" in their review is broader, including a series of GAN-based 2D image generation works and image-based rendering[18]. At the same time, our classification of deep learning-enhanced rendering methods is more in line with traditional graphics research and our demonstration of each work is more detailed. The central scheme around which we structure this report is the generality and application scenarios of each approach which is essential for most kinds of graphic applications. Novel view synthesis and relighting are commonly achieved in the following methods. Thus, we do not classify the approaches by them. We start by clarifying the scope of our report. Then we discuss the theoretical fundamentals of physically-based rendering and deep generative networks to provide readers with a better understanding of the methods described below. Then we discuss the landscape of applications that is enabled by deep learning-enhanced rendering. Finally, we summarize the entire report.

2 Scope of this STAR

In this state-of-the-art report, we focus on the classic and latest applications that combine deep neural networks with renderer components in the computer graphics rendering pipeline (Fig. 1). Specifically, we discuss how neural networks can replace or enhance the work of renderers in traditional rendering pipelines, and the advantages and disadvantages of the combinations. We categorize the deep learning-based rendering techniques and representative works that appeared in this survey in Table 1. For a clearer understanding, we first introduce the fundamentals of traditional physically based rendering and deep neural networks that relate to image synthesis. Then, we discuss some classic and up-to-date works, based on several aspects.

We do not cover any work based on neural radiance fields (NeRF)[70–78], which is not related to traditional rendering pipelines. NeRF is a novel view synthesis and 3D reconstruction method with implicit scene representation (density field) combined with the ray marching algorithm that draw great attention in the field of computer vision. Please refer to [80] for a deep comprehension. Similar to NeRF, the signed distance function (SDF) also utilizes implicit scene representation (signed distance field) which achieves better 3D reconstruction results. Although the SDF-based approach is somewhat different from the traditional rendering pipeline, this method utilizes the volume rendering algorithm, which is an

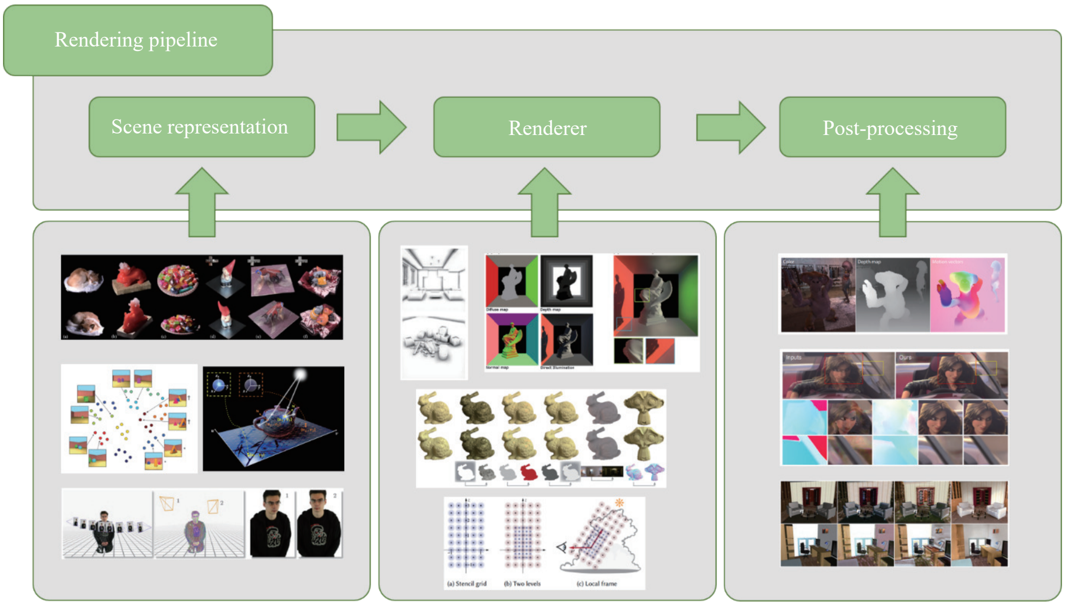


Fig. 1 Traditional rendering pipeline and corresponding learning-based methods. Each method replaces a (or more) step (steps) in the traditional rendering pipeline.

Table 1 Categories of deep learning-based rendering techniques and representative works in each category

Technique category		Papers	
		Generalize	Specific
Scene representation	Voxel-based scene representation	[19–21]	[22]
	Vector-based scene representation	[16, 23]	[24, 25]
	Mesh-based scene representation	[26]	[27–29]
	Point-based scene representation		[30–33]
	Network-based scene representation	[34, 35]	[36–39]
Global illumination	Ambient occlusion	[40–43]	
	Direct illumination	[44]	
	Indirect illumination	[45–47]	[48–50]
	Volume and subsurface	[51–56]	
	Human-related rendering	[57–60]	[61–64]
	Post-processing	[65–69]	
	NeRF (not discussed)	[70–78]	
	Denoising (not discussed)	[79]	

important algorithm in traditional graphics. In view of the fact that there is no review on SDF, we introduce one of the most popular SDF methods[36] (Section 5.4). Please refer to [81, 82] for more related work.

Although deep learning-based Monte Carlo denoising methods that aim to reconstruct denoised results from synthetic images generated by low sample per pixel (SPP) have made significant progress in recent years, and these processes are typically used as a post-processing stage in a traditional ray tracing pipeline, we will not discuss them because they have been well studied by a recent survey[79].

3 Theoretical foundation

3.1 Physically based rendering

Traditional graphics pipelines model image formation as a physical process in the real world: the photons emitted by the light source interact with objects in the scene as a bidirectional scattering distribution function (BSDF) determined by geometry and material properties, which are then recorded by the camera. This process is known as light transport and can be formulated by an equation, the classical rendering equation[83]:

$$L_0\left(\boldsymbol{x}, \boldsymbol{\omega}_0, \boldsymbol{\lambda}, \boldsymbol{t}\right) = L_\varepsilon\left(\boldsymbol{x}, \boldsymbol{\omega}_0, \boldsymbol{\lambda}, \boldsymbol{t}\right) + \int_\Omega f_r\left(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_0, \boldsymbol{\lambda}, \boldsymbol{t}\right) L_i\left(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\lambda}, \boldsymbol{t}\right)\left(\boldsymbol{\omega}_\mathrm{i} \cdot \boldsymbol{n}\right) \mathrm{d}\boldsymbol{\omega}_i \quad (1)$$

where L_o represents the outgoing radiance from a surface, $\boldsymbol{x}$ denotes the surface position, $\boldsymbol{\omega}_o$ denotes the outgoing direction of the light path, $\boldsymbol{\lambda}$ denotes the light wavelength, $\boldsymbol{t}$ denotes the moment of interaction, $\boldsymbol{n}$ denotes the surface normal, $\boldsymbol{\omega}_i$ denotes the incident direction of the light path, L_i denotes the incident radiance, f_r denotes the BSDF function, and $\boldsymbol{\Omega}$ denotes the hemisphere around the surface point. This equation omits consideration of transparent objects and any effects of subsurface or volumetric scattering. The most classic solver for this integral is Monte Carlo simulations[84]. In practice, the film only records three different wavelengths corresponding to the R, G, and B spectrum. The BSDF function is usually obtained by fitting the actual measured data of different materials. For more discussion on modelling lighting, materials, cameras, and geometry, please refer to [85].

3.2 Deep generative network

Traditional generative adversarial networks (GANs)[9] synthesize virtual images with statistics resembling the training set from a sampled random vector. The specific content of the generated pictures cannot be controlled. However, this is far from sufficient for scene rendering, as generating a random image for a specific scene is meaningless. To address this problem, feed-forward neural networks are trained with a distance to generate images giving conditional inputs[86]. However, these networks usually suffer from blurry results caused by the distance that only counts for individual pixels in image space and ignores the complex visual structure[87]. Later work proposed perceptual similarity distances[88–90] computed by pre-trained networks (usually VGGnet) to measure the distance between generated image and ground truth in high-dimensional feature space. Additionally, the structural similarity index measure (SSIM)[91, 92] distance is considered to improve the prediction quality. Although pairwise supervised training might achieve better metrics, the generated images may look unnatural. The conditional GANs (cGANs)[45] and StyleGAN[93] aim to generate images matching the conditional distribution of outputs given inputs that are indistinguishable from the human visual system. Although the generation can be controlled by the condition, the network cannot yet achieve explicit scene-level control.

4 General methods

There are many applications of deep learning-enhanced rendering, including surface rendering, subsurface rendering, volume rendering and novel view synthesis, relighting, photorealistic human appearance rendering, etc. Here, we categorize these applications into general methods and specific methods because the ability to use a trained network to different input scenes is important for rendering. Under each category, we detail each application by the renderer′s input and output types. Instead of classifying by novel view synthesis and relighting[17], we focus on the overall pipeline of the application as it is closer to the traditional computer graphics process.

General methods only need to be trained once and can then be applied to a range of scenes without retraining. The applications described below are general methods by default, and no additional explanations will be given.

4.1 Ambient occlusion generation methods

Ambient occlusion (AO) is a typical screen-space effect that is usually used in a real-time rendering pipeline that simulates the occlusion effect of objects in the scene. High-quality AO is usually generated by offline rendering while there is still work to generate inaccurate AO in real-time[94]. Deep shading[40] presents a novel technique to generate several rendering effects, including AO utilizing deferred shading buffers and convolutional neural networks. This is an early classic work using neural networks as renderers, using a U-shaped network that takes deferred shading buffers as input to generate specific rendering results. Erra et al.[41] introduce another method to generate AO, different from [40], the input to their network is not deferred shading buffers but sampled normals in the object space. And they use OpenGL Shading language to implement network inference which enables direct integration into the real-time rendering pipeline. Similar to [40], Zhang et al.[42] use a similar U-shape network structure and deferred shading buffer to generate AO. They also implement a Compute Shader Library to integrate the network into a real-time rendering pipeline. AOGAN[43] is the latest approach to generate screen-space AO. Different from all the above methods, they build a GAN-based neural network with a self-attention module with position and normal shading buffer as input. They also combined the perceptual loss of the VGG structure with the adversarial loss of the GAN structure to train the generator and discriminator jointly. Benefitting from their advanced network structure, they generate results close to offline rendering in real-time, see Fig. 2.

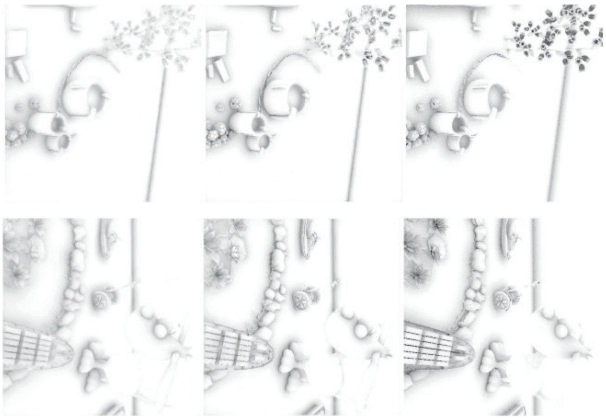
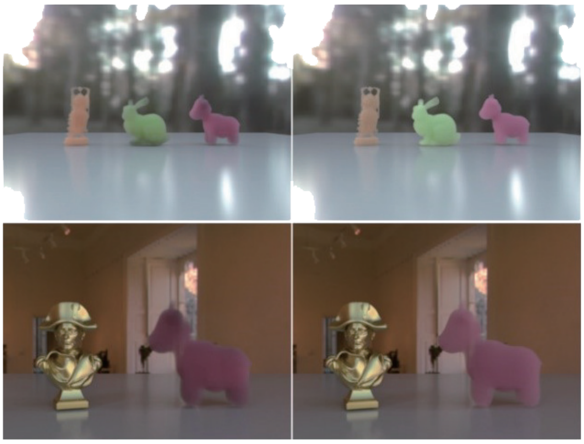


Fig. 2 Left to right: Results from Deep shading, AOGAN, and ground truth, respectively. Images taken from Ren and Song[43].

4.2 Volume and subsurface rendering

The techniques of rendering participate medium are a crucial part of traditional computer graphics, which can represent effects such as clouds, smoke, flames, waxes, multiple liquids, skin, etc. These rendering techniques can be divided into two types, one is the volume rendering of the medium as particles, including homogeneous volume rendering[95] and heterogeneous volume rendering[96, 97], and the other is the approximation of internal scattering, i.e., bidirectional scattering surface reflectance distribution function (BSSRDF) based methods[3, 98] commonly used to render high density, high albedo medium such as wax, skin, marble, etc. Although the rendering of the participating medium is relatively mature, there are still problems with time-consuming volume rendering and the inaccuracy of BSSRDF-based methods, so deep learning-based volume and subsurface rendering techniques have emerged in recent years.

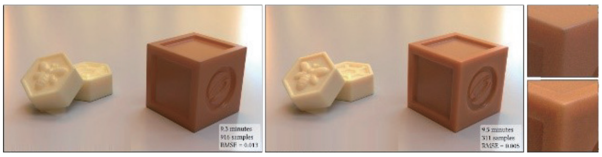
Deep scattering[51] proposed a method to synthesize multi-scattered illumination in clouds using deep radiance-predicting neural networks (RPNN), which efficiently synthesize the in-scattered radiance to replace the costly evaluation of Monte Carlo (MC) integration. Instead of predicting the full radiance directly, they opt for only multi-scattered transport and employ MC integration for the rest of the transport. Their method achieves an up to 4 000× speedup over path tracing and the bias is visually acceptable. Panin and Nikolenko[52] improved Deep scattering by proposing the Faster RPNN which is 2–3 times faster than the RPNN. They decrease the RPNN network size by using a baking network for baking light of a single directional light source and decrease the descriptor by passing in the rendering network a much smaller cloud descriptor, thus saving time both on inference and collection while obtaining a lower bias compared to RPNN. Abbas and Babahenini[53] introduce the latest method of rendering forest fog using a method similar to [40]. Although fog is a cloud-like medium, instead of considering a specific scattering process, they simply generate forest fog rendering results utilizing shading buffers (normal map, depth map, albedo map, RGB color map without fog) and a U-shape-based generative adversarial neural network. Their ground truth is images of forest fog rendered by traditional rendering pipelines. Zheng et al.[54] proposed a new method for rendering heterogeneous volumes that utilizes three neural networks to predict visibility, single scattering, and multiple scattering. Different from [51], this work not only predicts multiple scattering but also predicts all scatter processes at once, and they present multiple scattering by spherical harmonic (SH) basis functions. Thus, the network only needs to predict the coefficients of SH. Fig. 3 shows their rendering results of translucent materials and opaque materials.



Ours GT

Fig. 3 Rendering results and ground truth of translucent material and opaque material under different environment maps. Images taken from Zheng et al.[54]

In addition to the volume rendering method, there are also works related to subsurface scattering. Hermosilla et al.[55] introduced a deep learning-based method to learn the latent space of light transport from a 3D point cloud to represent the ambient occlusion, global illumination, and subsurface scattering shading effects. Compared to screen-space methods, their method represents the 3D scene as an unstructured 3D point cloud, which is later projected to the 2D output image during rendering. For each effect, they trained a network individually. For the subsurface scattering network, its input is the position, normal, albedo, direct illumination, and scattering coefficients corresponding to each point of the point cloud, and the final output is the rendering result of the point. The state-of-the-art approach for subsurface scattering rendering was proposed by Vicini et al.[56] To address the error caused by the semi-infinite plane assumption and diffusion-based approximation in the traditional BSSRDF method, they abandon the idea of diffusion approximation and use a neural network to predict the sampling point directly and their contributions while rendering. By fitting the scene surface to a quadratic polynomial, their network can handle arbitrarily shaped inputs. Their work produces a more realistic appearance and lower error compared to a photon beam diffusion path-traced reference (see Fig. 4).



(a) Photon beam diffusion (b) Shape adaptive BSSRDF

Fig. 4 Rendering of translucent soap blocks with photon beam diffusion and their work. Images taken from Vicini et al.[56]

4.3 Voxel-based scene representation rendering methods

Regardless of the rendering method, the representation of the scene determines the input form of the rendering neural network. The traditional renderer takes a scene representation file as input. However, this representation cannot be represented as a tensor, so it cannot be used as an input to a neural network. Inspired by recent progress in computer vision, many approaches that represent the scene as a voxel grid have emerged.

Visual object networks (VON)[19] presents a novel generative model, synthesizing natural images of objects with a disentangled 3D representation. Inspired by classic graphics rendering pipelines, they decomposed the generation model into three independent factors-shape, viewpoint, and texture. They first learn a shape-generative adversarial network that maps a randomly sampled shape code to a voxel grid. Then they project the voxel grid to 2.5D sketches with their differentiable projection module under a sampled viewpoint. Finally, they trained a texture network to add realistic, diverse textures to 2.5D sketches to generate 2D images that cannot be distinguished from real images by an image discriminator. The whole model is trained end-to-end on both 2D and 3D data. Their scenes, although randomly generated, are still represented using a voxel grid to represent three-dimensional structures. RenderNet[20] proposed a differentiable rendering convolutional network with a projection unit that can render 2D images from 3D shapes represented by a voxel grid. Benefitting from their differentiable renderer, their work enables relighting, different kinds of shading (phong, contour line, cartoon, ambient occlusion), novel view synthesis, and shape reconstruction from images. RenderNet passes a voxel grid, camera pose, and light position as input, and applies a view-projection transformation to convert the voxel grid to the camera coordinate system. After trilinear sampling, the transformed voxel grid is sent to a 3D convolution network with a projection unit to produce 2D feature maps that are sent to a 2D convolution network to compute shading. The network can alternatively produce normal maps of the 3D input. They also demonstrate their ability to iteratively recover a 3D voxel grid representation of a scene from a single image utilizing the differentiable renderer. Neural voxel renderer NVR[21] presents a deep learning-based rendering method that maps a voxelized scene into a high-quality image. Their method allows control of the scene that is similar to a classic graphics pipeline, including geometric and appearance modifications, lighting condition modification, and camera position modification. They demonstrate the effectiveness of their method by rendering scenes with varying scene settings. Their main contribution is presenting a novel neural network model that takes a voxel representation of the scene as input and learns how to render it. Two neural renderers: NVR and NVR+ are designed to render the scene. However, NVR generates a blurry and artifact result when the color pattern of the input voxels forms a high frequency and irregular texture. RenderNet[20] is the backbone of the NVR network. As the input of the NVR network, the voxel is first sent to the 3D encoder, which contains a series of 3D convolutions, and then passed through the reshape unit to become a 2D feature. These features are finally subjected to a series of 2-dimensional convolutions as the final feature of the voxel. Light conditions are also processed by two-layer fully connected layers and tiled to the final feature so that the lighting information is encoded. Finally, a 2D decoder processes the final feature to generate the output image. The NVR + network adds the splatting processing network and the neural rendering network based on the NVR network. The splatting processing network first synthesizes an image by splitting the center of the colored voxels in the target view and then passes this image through a 2D convolution encoder. The output of this network is then concatenated with the features from NVR, and the final result is processed by a U-Net (Neural rendering network) to generate the output image. Fig. 5 shows the neural rendering results of objects with NVR+. Although scene parameters can be modified, there are still some fixed attributes, such as light color, camera focal length, object material, etc.

4.4 Vector-based scene representation rendering methods

Since deep neural networks deal with tensors, it is an intuitive way to represent the scene as a vector that can be directly passed to neural networks. There are also methods dealing with vector-based scene representation in recent years.

Eslami et al.[16] introduced the generative query network (GQN), which is a framework for learning a vector embedding of a scene relying only on a few scene observations. The GQN takes several images taken from different viewpoints, and the corresponding camera poses as input and constructs a latent vector that encodes information about the underlying scene. This latent vector is designed to represent the complete scene (e.g., object geometries, colors, positions, lighting, and scene layout), and it is unaware of the viewpoints. Each time a new observation is added, the latent vector representing the scene will sum up the latent vector of the observation to obtain a more complete and accurate scene representation. The GQN′s generator is responsible for generating an image, given the scene representation and a new camera viewpoint as input. Only when the latent vector represented by the scene is accurate enough can the generation network synthesize the correct novel view image. At the same time, although only a few viewpoints per scene are used to train GQN, it is able to render unseen scenes from arbitrary viewpoints. However, their approach can only handle simple scenes with basic shapes and unreal lighting.

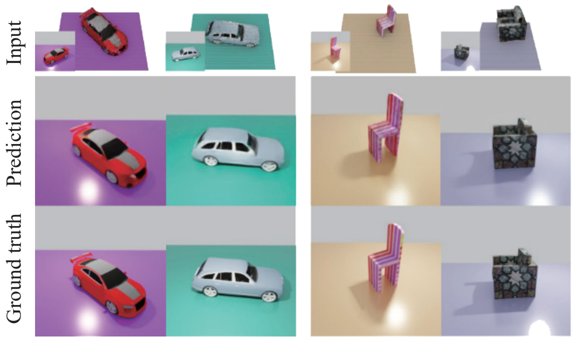


Fig. 5 Neural rendering of cars and textured objects with NVR+. Images taken from Rematas and Ferrari[21].

Liao et al.[23] defined a new task of 3D controllable image synthesis and proposed a method for solving it. They considered generating a vector-based scene representation from a controllable 3D generated and rendering it using a 2D generator. Their method consists of three main parts: a 3D generator, a differentiable projection layer, and a 2D generator. The 3D generator maps a latent code drawn from a Gaussian distribution into a set of abstract 3D primitives. Then, the differentiable projection layer takes each 3D primitive as input and outputs a feature map, an alpha map, and a depth map. Finally, the GAN-based 2D generator refines them and produces the final rendered image. A background vector is also projected and rendered to composite with the final render generating the full rendered result. Although their work can generate controllable scenes (object rotation and translation), the generated geometry and lighting are relatively simple, and there is no control over properties such as materials and lighting.

4.5 Network-based scene representation rendering methods

Whether the scene is represented by voxel, point cloud, mesh, or any discrete form, the precision of the representation is limited, so interpolation is always applied. However, implicit representation of a scene utilizing a neural network provides a continuous 3D scene representation method that infinite precision which is independent of the original scene representation. Therefore, methods for rendering implicit scene representations have emerged in recent years.

Oechsle et al.[34] proposed a novel implicit representation of surface light fields that captures the visual appearance of an object. They condition the surface light field with respect to the location and spectrum of a small light source which allows relighting and novel view synthesis using environment maps or manipulating the light source. Taking the encoding of an input image, the encoding of the corresponding input shape, and a lighting configuration, the conditional implicit surface light field (cSLF) outputs a predicted image which computes a photometric loss with the ground truth image. The cSLF network is a two-step model. First, the 3D location, shape feature vector, and image feature vector are mapped to a D-dimensional appearance feature. This appearance feature is a localized appearance representation independent of the viewpoint and lighting condition. Then, the appearance vector, lighting vector, viewpoint, and shape feature vector are fed into the lighting model to synthesize the RGB image. Their cSLF is capable of inferring light fields of novel unseen objects and preserving the texture, reflection, and shadow effects. However, their work can only handle relatively simple scenes and lighting conditions and cannot restore the specular effect well.

IBRNet[35] proposed a method to synthesize novel view images of complex scenes by interpolating a sparse set of nearby views. Utilizing an MLP and a ray transformer, they estimate radiance and volume density at continuous 5D locations (3D spatial and 2D viewing directions) only taking multi-view images as input. Unlike NeRF-based methods that need to retrain for a novel scene, they learn a generic view interpolation function that generalizes to novel scenes. The framework of their method is very similar to NeRF and is divided into three parts. They first identified a set of neighboring source views and extracted their image feature using a shared U-Net-based convolutional neural network. Then, for each ray in the target view, the IBRNet predicts the colors and densities for each sample along the ray. In practice, they aggregate the image color, features, and view direction from the neighboring source views as the MLP input and output of the color and density feature. The density features are then passed to the ray Transformer, which contains positional encoding and multi-head self-attention to the sequence of density features to predict the final density value for each sample. They use volume rendering to accumulate colors and densities along the ray to render the final image. Although their approach can handle the novel unseen scene, they still need to fine-tune each scene to obtain comparable results (see Fig. 6).

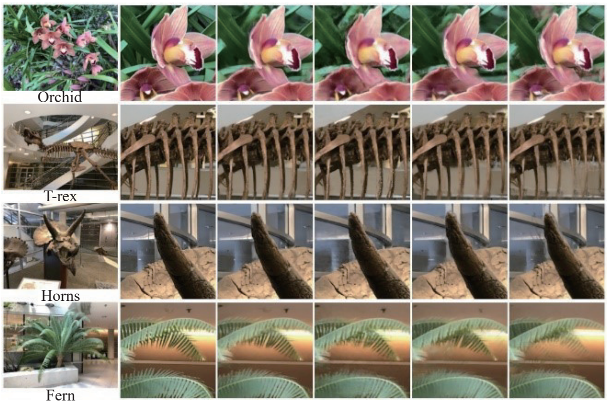


Fig. 6 Qualitative comparison on real-forward-facing data[99]. From left to right: Ground truth, their fine-tuned method, NeRF[70], and their method, LLFF[99]. Images taken from Wang et al.[35]

4.6 Mesh-based scene representation rendering methods

A polygon mesh is the most traditional way to represent scenes for traditional graphics pipelines. However, it is not commonly used in deep learning-based rendering due to the non-derivable nature of rasterization that prevents back-propagation. However, there are still works to solve this problem in a number of ways.

Neural 3D mesh render[26] proposes a method to approximate gradients for rasterization, which enables the back-propagation of neural networks. They are able to perform single image mesh reconstruction supervised with silhouette images utilizing the neural renderer. Their method demonstrates the potential of integrating mesh renderers into neural networks. The focus of this work is on the computation of rasterization gradients. Traditional rasterization is a discrete operation that determines the color of each pixel by judging whether it overlaps with the mesh. They replace the sudden change in pixel color caused by the intersection of the mesh with a gradual change using linear interpolation. Thus, the color change becomes a continuous process associated with mesh vertices.

4.7 Global illumination rendering methods

Although the rendering method mentioned in the above paragraphs can cope with various scene representations and achieve novel view synthesis and relighting, they basically do not handle global illumination caused by light interaction with different objects in the scene.

According to our survey, most deep learning-enhanced global illumination rendering methods do not explicitly use a neural renderer to generate the final rendering image but replace part of the render equation with a neural network while integrating. Therefore, we introduce the following work briefly.

Deep illumination[46] presents a novel deep learning technique for approximating global illumination (GI) in real-time using conditional generative adversarial networks (cGANs)[45]. Their pipeline is intuitive: First, they generate deferred shading buffers (normal map, direct lighting, diffuse map, depth map) and global illumination ground truth via VXGI[100] and GPU path tracing[101]; Then, the generated buffers are passed to a U-Net-based generator network to predict the GI image. Finally, the predicted GI image and buffers or ground truth image are passed to the discriminator network. Their method is a relatively early work and provides a baseline for global illumination neural rendering. Neural control variates (NCV)[47] propose a method for unbiased variance reduction in parametric Monte Carlo integration. Using the neural network to learn a function that is close to the render equation, as well as a neural importance sampler to produce the probability of sampling, and another neural network that infers the solution of the integral equation, they dramatically reduce the noise at the cost of negligible visible bias.

4.8 Direct illumination rendering methods

In general, direct lighting can be easily and efficiently obtained through rasterization or ray tracing. However, there is still work to learn the rendering of direct lighting through neural networks.

Suppan et al.[44] proposed a neural direct-illumination renderer (NDR) to render direct-illumination images of any geometry with opaque materials under distant illumination. The network framework is relatively simple: Given deferred shading buffers (normal map, roughness, depth), they first generate diffuse and specular coarse shading results, then the illumination, which is encoded as a vector of 75 SH coefficients, combined with coarse shading and deferred shading buffers are fed into the NDR to generate the diffuse and specular shading result. Finally, they obtained the final render result by multiplying albedo input with shading results and adding diffuse and specular parts together.

4.9 Post-processing methods

As higher resolutions and refresh rates, as well as more photorealistic effects bring great challenges to real-time rendering, neural networks are used in the post-processing stage to alleviate the burden of rendering pipelines. Superresolution and frame interpolation enable rendering pipeline work at a lower resolution or frame rates and recover target resolution and frame rates by deep learning methods.

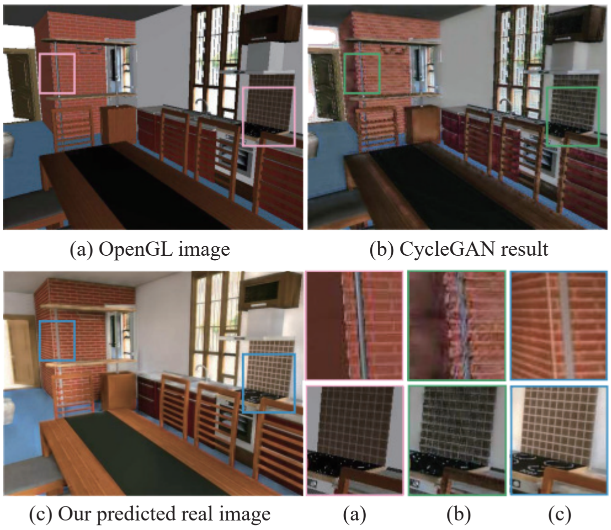
Superresolution is introduced to real-time rendering by NVIDIA. NVIDIA released DLSS2.0[65] in 2019, which is the first deep learning-enhanced superresolution method that can be applied in practice. However, since DLSS2.0 relies on NVIDIA's hardware platform, no technical information is publicly available. Xiao et al.[66] proposed NSRR, using U-Net[102] as the backbone to reconstruct the final result with the input including low resolution color, depth map and motion vector over multiple historical frames. High-resolution results can be achieved with NSRR at a real-time frame rate and most of the high-frequency detail can be recovered.

Frame interpolation in rendering is another way to reduce rendering task. Guo et al.[67] proposed ExtraNet to predict an extrapolated frame according to previous frames and current Gbuffers. Briedis et al.[68] presented a frame interpolation method for offline rendering applications.

Deep CG2Real[69] presented a method to improve the quality of OpenGL rendered images as a two-stage post-processing process. Their two-stage pipeline first generates an accurate shading with the supervision of physically-based renderings (PBR). Furthermore, they increase the realism of texture and shading utilizing a CycleGAN[11] network. They demonstrate that their method yields more realistic results compared to other approaches via evaluations on the SUNCG[103] dataset. They first leverage the generative neural networks that take deferred shading buffers (albedo map, normal map, and OpenGL shading map) as input and predict the PBR shading map. This shading map is then product with the albedo map generating the PBR image. Note that this training process is supervised with PBR rendering results. Later, another generative neural network predicted the real albedo and shading image which are responsible for generating the real result. This stage of training is supervised with unpaired data. Fig. 7 shows their predicted real image compared to the OpenGL image and CycleGAN result.

4.10 Human-related rendering methods

Human-related rendering has drawn great attention in computer graphics including skin rendering, hair rendering, face rendering, body animation rendering, etc. Traditional computer graphics usually model the human appearance as the physical process of light interacting with the human body and rendering with real-time rasterization or offline ray tracing. With the development of neural networks in recent years, deep learning-based human-related rendering methods have started to proliferate and are gradually replacing traditional methods. Some of the methods described below are similar to the previous paragraphs, but the focus of this paragraph is on human-re-


(a) OpenGL image (b) CycleGAN result

(c) Our predicted real image (a) (b) (c)

Fig. 7 Compared to OpenGL rendering (a) and single-stage prediction with CycleGAN (b), their result restores more realistic lighting and textures. Images taken from Bi et al.[69]

lated rendering; thus, it's all covered here.

Wei et al.[57] present an adversarial network for rendering photorealistic hair that takes a strand-based 3D hair model as input and provides user control for color and lighting through reference images. Benefiting from the simple forward pass of their network, they achieve a real-time rendering rate. Given a natural image, they generate four processed images by three sequential image processing operators: the segment hair image, the gray image, the orientation map, and the edge activation map. The segment hair, gray image, and orientation map are all encoded into their own latent space with a feature vector. Given a 3D hair model at inference time, they first extract the edge activation map from a randomized rendered image of the desired viewpoint. Then generators are applied sequentially in the inverse order of the image processing flow (see Fig. 8).

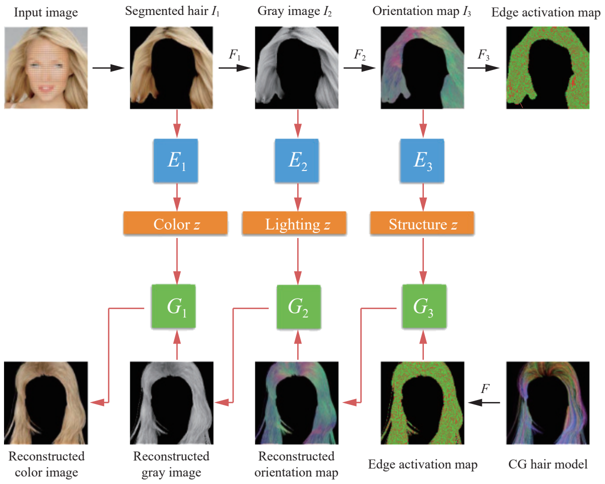


Fig. 8 Pipeline of [57]. The top row shows the image processing flow of an input natural image, and the bottom row shows the inference flow from right to left. Image taken from Wei et al.[57]

LookinGood[58] proposed a novel method to augment a real-time performance capture system with a deep neural network that takes a coarse rendered textured 3D reconstruction from a novel viewpoint and outputs high-quality rendering results that perform super-resolution, denoising, and completion of the original images. They test their method in two situations: One involving an upper-body reconstruction of an actor from a single RGB-D camera, and the second consisting of full-body capture. They use extra cameras except for the reconstruction camera to provide ground truth, which achieves self-supervised training. The backbone of LookinGood is a U-Net-like architecture. The system is specifically designed for VR and AR headsets and accounts for consistency between two stereo views. Fig. 9 shows the re-rendering results w.r.t. to viewpoint changes. Although their method can be generalized to a different actor, the quality of the unseen actor is reduced.

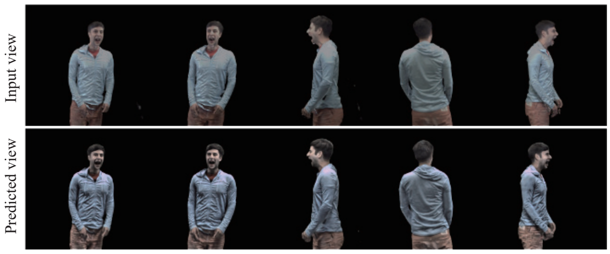


Fig. 9 Neural re-rendering results of different viewpoints. Image taken from Martin-Brualla et al.[58]

Meka et al.[59] introduced a method that combines traditional graphics pipelines with neural rendering to generate photorealistic renderings of dynamic performances under novel viewpoints and lighting assuming the availability of an approximate geometry of the subject for every frame of the performance. Their method is capable of rendering unseen subject poses and novel subject identities and significantly outperforms the existing state-of-the-art solutions. A U-Net architecture is first exploited to extract features from the two spherical gradient illumination images of each viewpoint that concatenated each pixel with view direction. After acquiring the feature, they warp the features of every viewpoint using warp fields and pool all of them together into a single tensor to remove the dependency on the order of the input images according to the feature weights computed by the dot product between the camera viewing direction and the surface normals. The texture coordinate feature is then sampled by a warp corresponding to the target camera view to generate the resampled features. Then, they generate a reflection map and a light visibility map and multiply the light visibility map elementwise with the concatenation of resampled features, reflection map, and view direction, generating the neural diffuse rendering image. Additionally, the resampled features are fed into an Alpha Matting network, predicting the alpha mask. Finally, the neural diffuse rendering image and alpha mask are passed through a U-Net that generates the actual rendered images, see Fig. 10 for a comprehensive understanding.

Rendering with style[60] proposes combining the traditional rendering pipeline and neural rendering of faces, automatically and seamlessly generating full-head photorealistic portrait renders from only facial skin render without any artist intervention. Their method is also capable of rendering and preserving identity over animated performance sequences. They first synthesize a high-quality skin render via a traditional rendering pipeline with an alpha mask from a 3D face geometry and appearance maps. This rendered image is then projected into a pretrained Style-GAN2 network[104] to realistically inpaint the missing pixels of the portrait (eyes, hair, the interior of the mouth). The final compositing step overlays the raytraced skin appearance on top of the projection results.

5 Specific methods

Usually, only scenes or objects specified during network training can be rendered by specific methods. A new network needs to be retrained for every new scene or object. For example, if the method operates on a single car scene (with a specific lighting condition, in a specific location), then changing the instance of the car, increasing the number of cars, changing the lighting conditions, etc., will disable the network. In general, specific methods produce higher quality than general methods at the expense of training time. The applications described below are specific methods by default, and no additional explanations will be given.

5.1 Voxel-based scene representation rendering methods

DeepVoxels[22] is a learned viewpoint-invariant, persistent, and uniform 3D voxel grid of feature representation that encodes the view-dependent appearance of a 3D scene without explicitly modelling its geometry. The final rendered image is formed based on a 2D network that receives the perspective resampled version of the 3D volume. The scene-specific DeepVoxels feature representation is formed from a set of multi-view images without explicit 3D supervision. They first extract 2D feature maps using 2D U-Net and explicitly lift the features to 3D based on a differentiable lifting layer. The lifted 3D feature volume is fused using a gated recurrent network architecture. After feature fusion, the feature volume is processed by a 3D U-Net and then mapped to the camera coordinate system of the two target views via a differentiable reprojection layer. An occlusion network then computes the soft visibility of each voxel. Finally, a learned 2D U-Net rendering network generates the two final output images. Their network is trained end-to-end by a 2D re-rendering loss that forces the predictions to match the target views. They show several novel view synthesis results on challenging scenes and outperform baseline methods (see Fig. 11).

5.2 Point-based scene representation rendering methods

In addition to voxel representation, another common

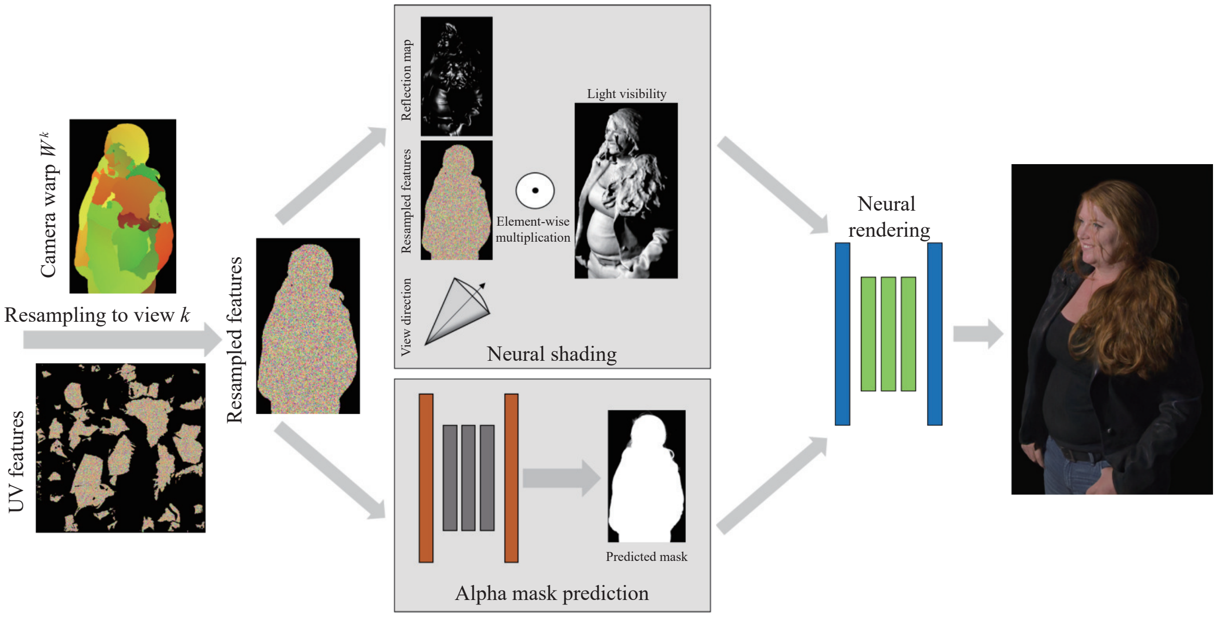


Fig. 10 Neural rendering pipeline. Given a target camera, the UV features are resampled. A neural shading model adds view-related information to resampled features. An alpha mask is predicted by the Alpha Matting network. A U-Net finally renders the target images. Image taken from Meka et al.[59]

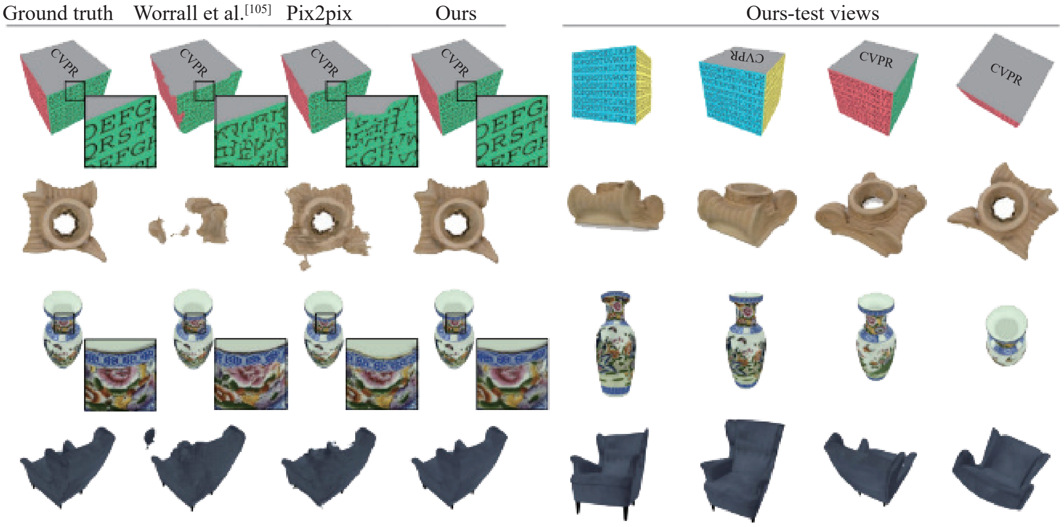


Fig. 11 Comparison of the best three performing models to ground truth and other samples of novel views generated by their model. From left to right: ground truth, Worral et al.[105], Isola et al.[15] and DeepVoxels. Images taken from Sitzmann et al.[22]

method of scene representation is using a point cloud. The advantage of this representation is that there have been works in computer vision to reconstruct point clouds from images, and the reconstructed point clouds can be rendered directly using these methods.

Meshry et al.[30] applied traditional 3D reconstruction from internet photos of a tourist landmark to generate a point cloud corresponding to the landmark. They train a neural rendering network that takes deferred shading buffers (depth, color, semantic labelling) as input and generates realistic renderings of the landmark with relighting and novel view synthesis. Given a large internet photo collection of a scene, they first reconstruct a dense colored point cloud using structure-from-motion[106] and multi-view stereo[107] and then render the point cloud from the viewpoint of each image to generate the aligned dataset. Per-pixel albedo and depth are generated by using point splatting with a z-buffer. To model the different appearances with relighting under a viewpoint, they pre-train an appearance encoder that takes deferred buffers and real images as inputs using a triplet loss. Then a neural renderer is trained using reconstruction loss and GAN loss and finally fine-tuned with an appearance encoder. To account for the transient object in the scene (pedestrians, cars, etc.), they also concatenate the semantic label to the deferred buffer. The ground truth semantic segmentations are computed using DeepLab[108] on the input image, while they train a separate semantic labelling network that takes deferred shading buffers as input for inference. However, their work produces poor results for landmark details, such as text, and there are not enough input images for a scene.

Aliev et al.[31] presented a novel point-based method that uses a raw point cloud representation of the scene and generates novel view synthesis render results with a learnable neural descriptor of each point and a deep rendering network. They first attach an 8-dimensional descriptor to each point and rasterize the points with a z-buffer at several resolutions corresponding to the given camera parameters. Each rasterization is fed to different downsampling layers of U-Net and synthesis of the final render result. They optimized the parameters of the rendering network and the neural descriptors by back-propagating the perceptual loss function. They also show that their approach is able to model and render scenes captured by hand-held RGBD cameras as well as simple RGB streams.

Dai et al.[32] presented a novel neural point cloud rendering pipeline through multi-plane projections. The neural network takes the raw point cloud of a scene as input and outputs image or image sequences from novel camera views. They propose a method to project 3D points into a layered volume of camera frustum so that the network automatically learns the visibility of 3D points. The whole framework of the network consists of two modules: multi-plane-based voxelization and multi-plane rendering. The first module divides the 3D space of the camera view frustum uniformly into small frustum voxels according to image size and a predefined number of planes. Aggregation operations are also adopted for each small frustum to generate a multi-plane 3D representation that concatenates with normalized view direction and sends it to the render network. The render network predicts a 4 channels output (RGB + blend weight) for each plane. The final output is produced by blending all planes according to blend weights. Finally, the whole framework is supervised by perceptual loss. They demonstrate that their method produces more stable renderings compared to previous methods (see Fig. 12).

Sanzenbacher et al.[33] trained a deep neural network to generate photorealistic rendering results of a specific scene in real time by learning light transport in a static or dynamic scene. Their approach operates in both 3D and 2D space, thus enabling global illumination effects

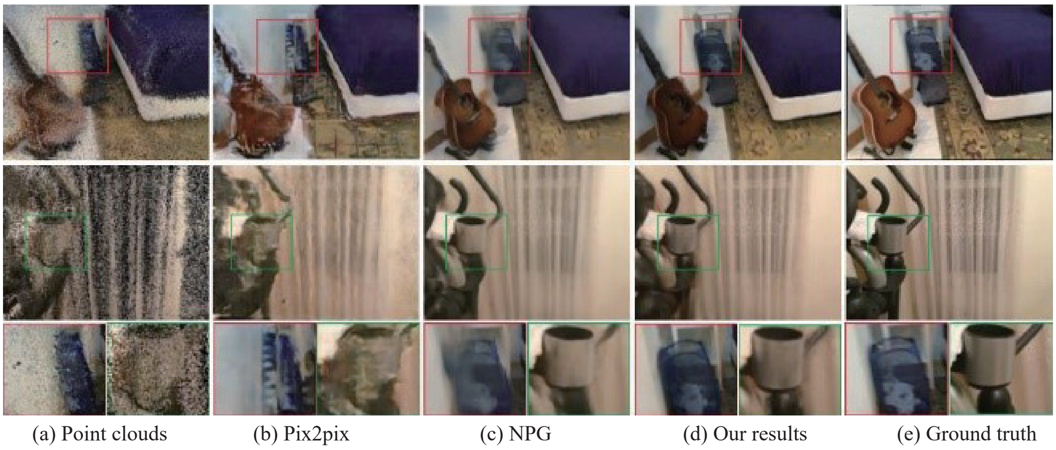


Fig. 12 Two comparisons on cases with noisy depth. NPG[31] and Pix2Pix[15] either completely miss the correct objects or produce a mixture of foreground and background. Image taken from Dai et al.[32]

and 3D scene geometry manipulation. They first represent the scene in the form of an unstructured point cloud sampled from the scene's surface and attach additional properties (albedo, light spectrum) to each point. The point cloud is then processed with a light transport layer which is a PointNet-based architecture[109] with ResNet-blocks[110] of depth two to learn light transport in the scene. The network output is projected into a 2D image. Then combined with additional image space information (depth, normal, albedo, view ray), the projection features are sent to the image synthesis layer to synthesize the final image. By minimizing the MSE error of the generated image and noisy rendering obtained from a physically-based renderer, they jointly optimize the whole model. They also prove that using noisy images as ground truth, the gradient estimates are unbiased.

5.3 Vector-based scene representation rendering methods

Chen et al.[24] proposed a novel relightable neural renderer (RNR) for novel view synthesis and relighting utilizing multi-view images as input. RNR models the physical rendering process of image generation, specifically, in terms of environment lighting, object intrinsic attributes, and light transport function (LTF). RNR conducts regression on these three individual components rather than translating deep features to appearance. Benefiting from the physically-based rendering process, their method improves the quality of novel view synthesis and relighting. They decomposed the render equation into albedo, LTFs and lighting, and use spherical harmonics (SH) to fit the lighting. They first follow the step of [111] and apply a K-nearest neighbor (K-NN) method to search the neighborhood of each 3D mesh vertex and then apply multiple graph convolutional networks (GCNs) to extract the global features as a vector of the 3D geometry. After that, they repeat and concatenate the feature vector with the U-Net feature map after the first downsampling layer. This U-Net network is the light transport net (LTN) and takes neural texture, normal map, and light direction map as input and outputs a light transport map that contains per-pixel light transport at each sampled light direction. Finally, they retrieved the radiance on each sampled light direction and integrated it with albedo and SHs to render the final image. They use the L1 loss for the difference between rendered images and ground truth images. Fig. 13 shows the relighting and novel view synthesis results of RNR.

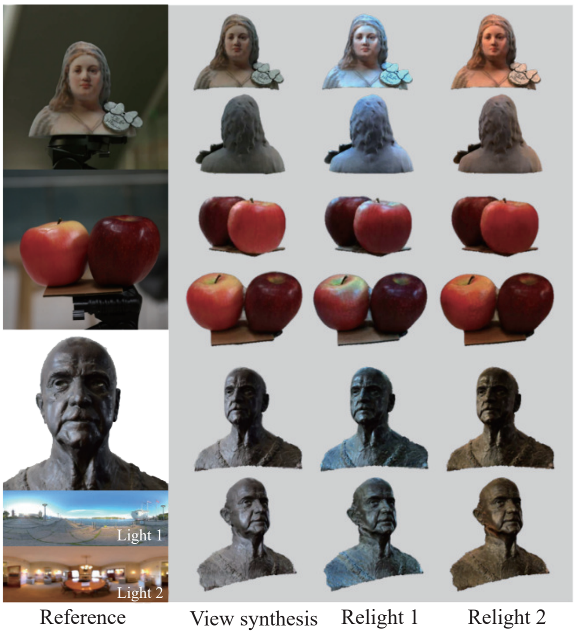


Fig. 13 Relighting and novel view synthesis results of RNR on real data. Image taken from Chen et al.[24]

Granskog et al.[25] present a technique to adaptively disentangle lighting, material, and geometric information, generating a vector-based scene representation that preserves the orthogonality of these components. The scene encoding network takes several high-quality observations of the scene attached with deferred shading buffers (position, normal, depth) and camera parameters as input and produces a view-independent neural scene representation vector. This vector is the average of all generated observation feature vectors. For a novel view, the representation vector, camera parameters, and corresponding deferred shading buffer are passed into a neural renderer to obtain an image of the novel viewpoint. Their method is similar to [16] but focuses on adaptively partitioning the neural scene representation and in-depth analysis of existing image generators with respect to the partitioned representation. Since their work disentangles elements in a scene, it is possible to use the lighting of one scene to relight another scene by replacing the lighting part of the scene representation vector (see Fig. 14).

5.4 Network-based scene representation rendering methods

Scene representation networks (SRNs)[37] is a classic method that proposes a continuous 3D-structure-aware scene representation that encodes both geometry and appearance. They map the world coordinates to a feature representation of local scene properties. Taking only 2D images and their camera poses as input, SRNs can be trained end-to-end with a differentiable ray-marching algorithm. In practice, the scene representation function is represented by a multi-layer perceptron (MLP) that learns to map a spatial location to a feature representation of scene properties of that spatial location. A two-step differentiable ray-marching algorithm is used to generate the final rendered image by first finding the world coordinates of the intersections of the camera ray with scene geometry and then mapping the feature vector to a color. They introduced a ray marching long short-term memory (RM-LSTM) to handle the first problem and a per-pixel MLP to map a single feature vector to a single RGB vector. After training, the view-independent MLP can be queried by a novel camera view with the ray-marching algorithm and then rendered by the per-pixel MLP.

Differentiable volumetric rendering (DVR)[38] presents a differentiable rendering formulation for implicit shape and texture representations. Similar to [37], they also represent the shape and appearance with a neural network. However, they design an occupancy network assigning a probability of occupancy to every point in 3D space and extract the object surface using isosurface extraction techniques[112] instead of the RM-LSTM network. They also generate the final rendered image directly from the

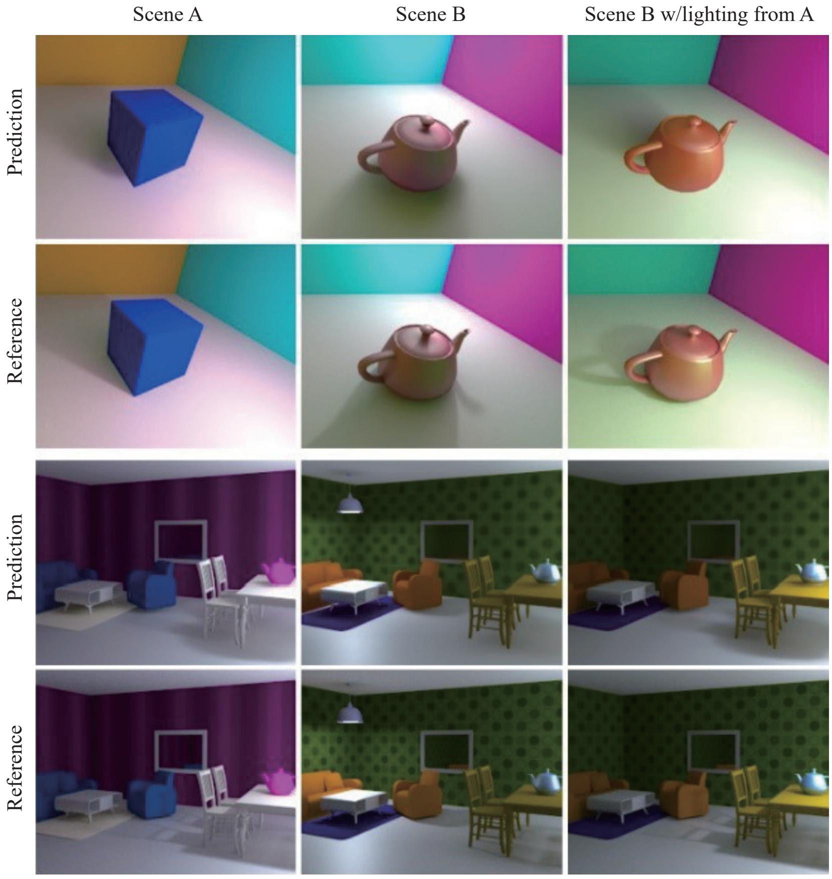


Fig. 14 The lighting partition of scene A replaces the lighting partition of scene B. The relighting result is shown in the right column. Image taken from Granskog et al.[25]

texture field[113] instead of the per-pixel MLP. For the single-view reconstruction task, they process the input image with an encoder and use the output to condition the occupancy network and texture field. They show their multi-view 3D reconstruction results and single-view reconstructions (see Fig. 15). Similar to [37], their method can also achieve novel view synthesis.

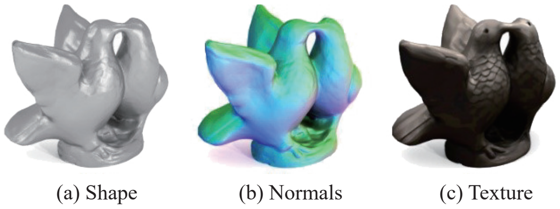


Fig. 15 They show the shape, normals, and the textured shape for their method trained with 2D images and sparse depth maps for scan 106 of the DTU dataset[1]. Image taken from Niemeyer et al.[38]

Yariv et al.[36] modelled the volume density as a function of the geometry, different from previous work modelling the geometry as a function of the volume density. They defined the volume density function as Laplace's cumulative distribution function (CDF) applied to a signed distance function (SDF) representation. This new density representation produces high-quality geometry reconstruction and enables the disentanglement of shape and appearance. Their framework consists of two MLPs, the first approximating the SDF of the learned geometry and the global geometry feature of dimension 256. The second MLP presents the scene's radiance field. Fig. 16 shows qualitative results sampled from the BlendedMVS[115] dataset.

Neural lumigraph rendering (NLR)[39] implicitly represents a scene surface and radiance field using a neural network that accelerates state-of-the-art neural rendering by approximately two orders of magnitude and is compatible with traditional graphics pipelines which enable real-time rendering rates. They present both the shape and appearance of 3D objects similar to IDR[116]. However, their backbone network is sinusoidal representation networks (SIREN)[117] instead of MLP. They model the shapes of the scene as SIREN-based SDF representation. The appearance is modelled as a radiance field for directions. They take multi-view 2D images and object masks as input to supervise the 3D representation. The loss function is relatively complex and contains a L1 image reconstruction loss for true foreground pixels, an eikonal constraint to regularize the scene representation network, a soft mask loss proposed in [116] defined for the non-foreground pixels, also a smoothness term to linearize the angular behavior of SIREN. Compared to the NeRF-based method, they only need to use sphere tracking to find the first intersection of the ray and the model, and then query the value of the radiance field, without accumulating samples along the ray, which leads to a faster rendering process. They also embedded their method into the traditional rasterizing pipeline to achieve a real-time rendering rate by extracting the mesh from SDF using marching cubes and then rasterizing the mesh using OpenGL to compute the vertex position buffer and angles between the ray towards the current rendering camera and the rays towards each of the projective texture map viewpoints. Finally, they apply the unstructured lumigraph rendering technique[118], generating the rendered image.

5.5 Mesh-based scene representation methods

Deep surface light fields (DSLF)[27] present a neural network called the DSLF to model light transport on vertices of object mesh using only moderate sampling of multi-view images. Their DSLF can achieve a high data compression ratio while performing real-time rendering on the GPU. They first obtain the 3D model of the object as a mesh by structure-from-motion (SFM) and then register the multi-view images with the mesh using feature extraction and matching and perspective-n-point (PnP)[119] techniques. They also conduct texture-aware remeshing to avoid blurring of the line features. The deep network finally takes the vertex position (represented by texture coordinates) and ray direction as input and outputs the final light transport of that vertex. During rendering,

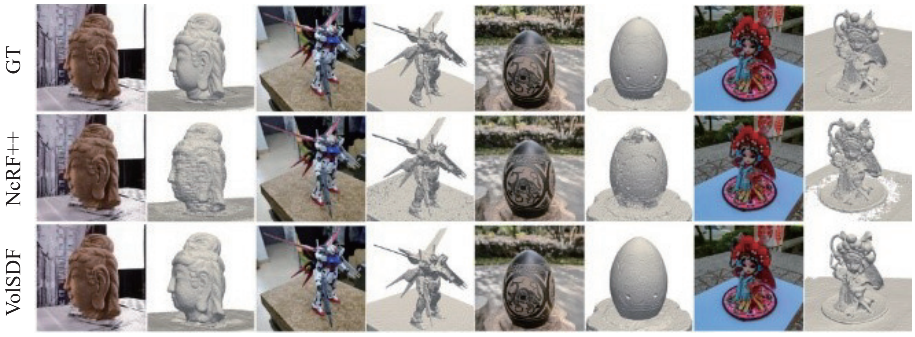


Fig. 16 Qualitative results sampled from the BlendedMVS dataset. From top to bottom, the ground truth, the NeRF++[114] results, and their results. Image taken from Yariv et al.[36]

their method integrates well with traditional rasterization pipelines by replacing the vertex shader with DSLF to predict the vertex color. Fig. 17 shows the novel view synthesis results of synthesis and real scenes.

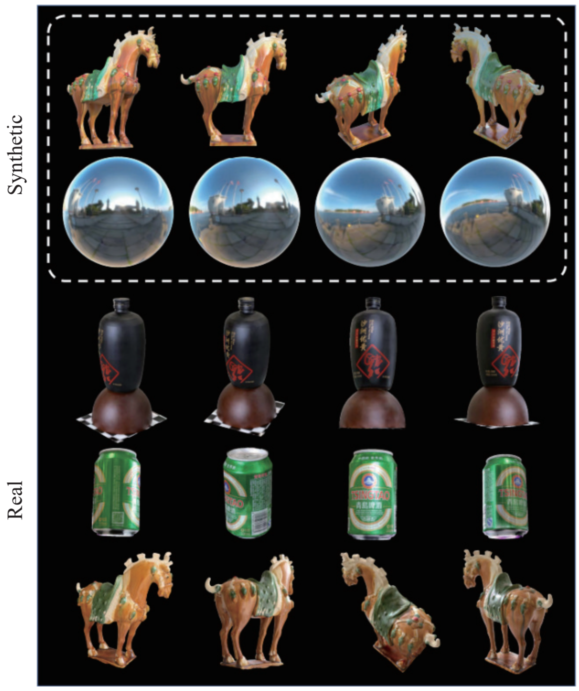


Fig. 17 DSLF rendering results from different viewpoints. Their method produces high fidelity results in both real and synthesis scenes with different materials. Images taken from Chen et al.[27]

Deferred neural rendering[28] introduces the neural textures, a learned feature map that is stored as maps on top of 3D mesh proxies that contain significantly more information than traditional textures. Different from the original 2D generative neural networks, their method achieves explicit control over the generated output. They show the effectiveness of their method on novel view synthesis, scene editing, and facial reenactment. They first obtain the coarse geometric proxy with UV-map parameterization and camera parameters using the COLMAP[107] structure-from-motion technique. Taking the geometry mesh and a neural texture as input, the standard graphics pipeline is used to render a view-dependent screen space feature map. This feature map is then converted to a photorealistic image via a U-Net-based deferred neural renderer.

Deferred neural lighting[29] proposes a novel method for novel viewpoint relighting of a specific scene. Different from traditional methods, which require dense samples of the view direction and lighting condition combination, their method utilizes unstructured photographs taken from a handheld acquisition scheme that only requires two cellphones. They demonstrate the effectiveness of their method in a variety of real-world and synthetic scenes. Similar to deferred neural rendering[28], they also reconstructed the geometric mesh with UV-mapping via COLMAP and generate a neural texture to represent the feature of the object. Instead of directly passing the projected neural texture to a neural renderer, they combined (via per-pixel multiplication) radiance cues, which are synthesized by rendering scene-independent bias materials under the target light onto the rough geometry, with projected neural texture and passed to a neural renderer to produce the final relit image of the scene. Moreover, they also predict a binary mask from the projected neural texture for compositing the relit appearance. Fig. 18 shows their simultaneously novel view synthesis and relighting results on captured scenes.

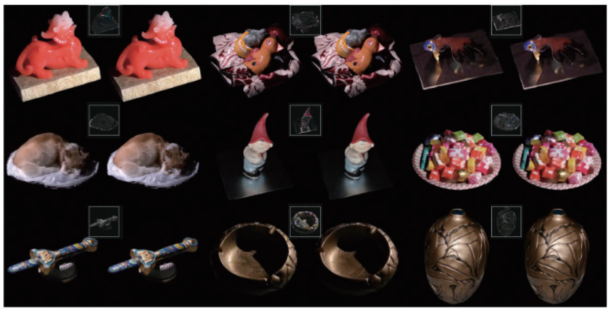


Fig. 18 Qualitative comparison between captured scenes ground truth (right) and simultaneously novel view synthesis and relighting results of their method (left). Difference images are shown in the insets. Images taken from Gao et al.[29]

5.6 Global illumination rendering methods

Ren et al.[48] proposed the first method to model global illumination with a neural network. They introduce a radiance regression function (RRF), presenting a non-linear mapping from local attributes to indirect illuminations. They first define the closed-form of indirect illumination and then train an MLP that takes position, view direction, point light position, and normal and reflectance parameters as input and predicts the RGB components of indirect illuminations. Combined with direct illumination, they finally obtain the global illumination result. Neural radiosity[49] directly uses a neural network to predict the solution of the rendering equation by minimizing the norm of its residual for each point in a 3D scene. They derive the MC estimate of the residual norm and the MC approximation of the residual norm gradient with respect to network parameters. Different from traditional neural network optimization, their model optimizes network parameters in the traditional ray tracing framework and computes gradients using the formula they derived. After training, images from arbitrary viewpoints can be computed efficiently (see Fig. 19). Diolatzis et al.[50] introduced the latest active exploration (AE) method using Markov chain Monte Carlo (MCMC) to render novel scene instances given an explicit parameterization of the scene configuration. The scene configura-

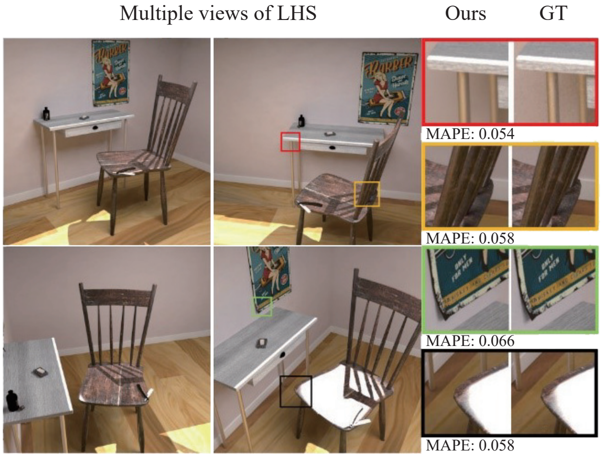


Fig. 19 Multiple views of a solution of their network. Images taken from Hadadan et al.[49]

tion controls the variables of the scene, such as changing objects, materials, lights, and viewpoint. The MCMC method generates scene configuration samples that best help training hard light transport paths (e.g., caustics and transmittance). During training, they explicitly model the scene and the set of variable parameters as a vector. They generate difficult instances of the variable scene to guide the PixelGenerator network using AE. In addition to the scene representation vector, the PixelGenerator also takes auxiliary deferred shading buffers as input and predicts the global illumination image path. At inference time, the explicit scene representation vector which contains requested configuration information is fed into the PixelGenerator with deferred shading buffers of the corresponding scene configuration to predict the final image. Note that their method is relatively efficient (4–6 FPS) and is capable of interactively altering the scene illumination by moving objects, the viewpoint, and modifying materials. Fig. 20 shows the interactive rendering results of their methods and controllable variables depicted in red.

5.7 Human-related rendering methods

Lombardi et al.[61] introduced a data-driven deep appearance model for rendering the human face that learns both facial geometry and appearance from a multi-view capture system. Their method generates realistic novel view images with no need for an accurate geometry model, which is a significant departure from the traditional graphics pipeline. They also integrated their model with an unsupervised technique for mapping images to facial states into virtual reality applications. Begining with multi-view input photos of an identity and a reconstructed mesh, they first unwrap the photos to generate the view-specific texture maps. They then computed the average texture of the texture maps. The average texture and the mesh are sent to a variational autoencoder (VAE)[12] which is conditioned by an output viewpoint and predicts a mesh and view-specific texture corresponding to the output viewpoint. With texture and geometry, it can easily render images from a novel point of view. The whole VAE is supervised by the generated mesh and view-specific texture reconstruction loss.

Liu et al.[62] proposed a method to generate video-realistic animations of real humans under user control. Compared to traditional human character rendering, they do not require a high-quality photorealistic 3D model, but a video sequence and a 3D template model of the person. They first reconstruct a 3D character model of the target person from static posture images and then obtain the training motion data from the monocular training video based on the method of [120]. These motion data are then fed to the character-to-image translation network with the color and depth of body part images to produce video-realistic output frames. At inference time, the Character-to-Image translation network takes the source motion data with the conditioning input (color and depth body part images with a 3D character model) as input to reenact the target human character. The source motion

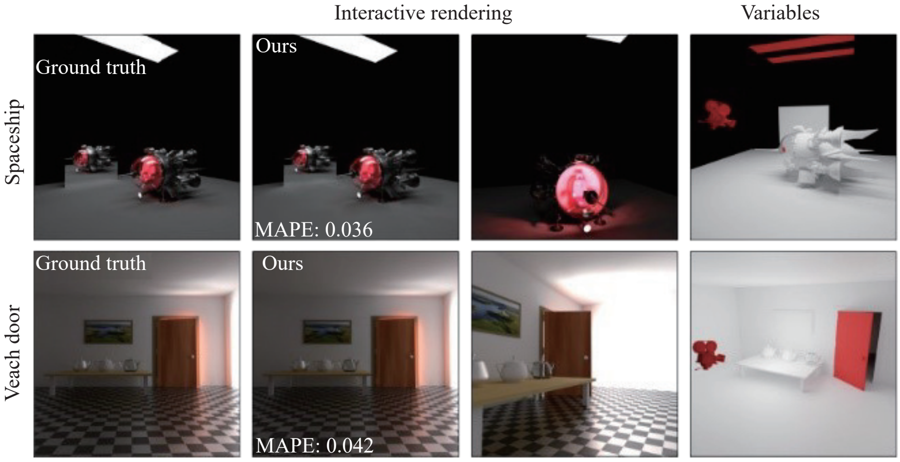


Fig. 20 Interactive rendering results. From left to right: Ground truth path traced images; their interactive neural renderer with 4 to 6 FPS; the rendering result of a varied scene; the variable parts of the scene depicted in red. Images taken from Diolatzis et al.[50]

data can not only be provided by monocular video but also from user-defined motion or motion capture data. Their results outperformed the state-of-the-art methods in learning-based human video synthesis.

Wu et al.[63] present a neural human renderer (NHR) for rendering photorealistic free-view video (FVV) from dynamic human captures under a multi-view setting. Experiments show that NHR outperforms the state-of-the-art neural and image-based rendering techniques, especially on hands, hair, nose, foot, etc. As the input to the NHR, the multi-view stereo (MVS), which consists of a synchronized, multi-view video sequence, is exploited to construct a point cloud at each frame. Each point in the point cloud has color, computed through reprojection on the input view images. Next, feature extraction (FE) based on PointNet++[121] was used to process the spatiotemporal point cloud sequence generating 3D-point descriptors. The descriptor with camera parameters is projected and rasterized to produce a feature map and depth map corresponding to the viewpoint. Finally, a U-Net-based renderer maps the feature map and depth map to the output RGB image and mask image. The point cloud reconstructed from MVS produces patches of holes on textureless or occluded regions. Thus, they refine their geometry by rendering a dense set of new views and using the resulting masks as silhouettes and conduct visual hull reconstruction based on space-carving or shape-from-silhouettes (SfS). Fig. 21 shows the FVV results on a dance scene.

Zhang et al.[64] proposed a method for learning a neural representation of light transport (LT) of the human body with a rough 3D geometry and multi-view one light at a time (OLAT) images. They model non-diffuse and global LT in texture space as residuals added to physically based diffuse rendering and enable high-quality (with complex material effects and global illumination) novel view synthesis and relighting simultaneously. Their framework consists of two main paths: the observation path and query path. The observation path first takes several nearby texture-space residual maps (observed minus diffuse base) sampled around the target light and viewing direction. The physically-based diffuse base textures are generated by multiplying the albedo texture, the light cosines texture, and the view visibility texture corresponding to the sampled view and lighting direction. The residual maps are then fed into an encoder generating multiscale features that are pooled to remove the dependence on their order and number. The pooled feature is concatenated to the feature activations of the query path network, which takes the query light cosines map, query view cosines map, and diffuse base map of the target view and lighting direction. The query path network then synthesizes the non-diffuse residuals, which represent global illumination and complex material effects. Finally, the non-diffuse residuals and diffuse base textures are wrapped to image space utilizing UV wrapping predefined by coarse geometry. Fig. 22 shows the simultaneous relighting and views synthesis results of their method.

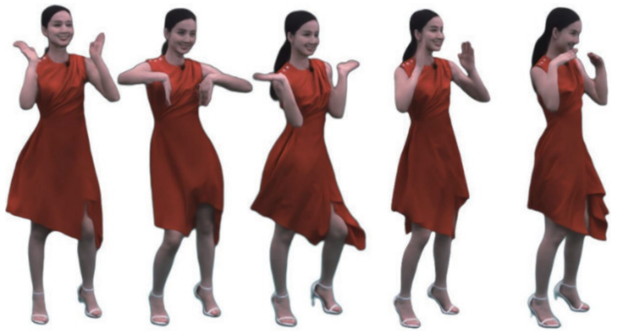


Fig. 21 Free view video results on a challenging dance scene using NHR. Red blouses impose significant challenges in 3D reconstruction. Image taken from Wu et al.[63]

6 Conclusions

Deep learning-enhanced rendering has drawn great attention in both computer graphics and computer vision research fields in recent years. This state-of-the-art report spans a variety of use cases that range from general and specific methods of ambient occlusion generation, volume and subsurface rendering, multiple scene repres-

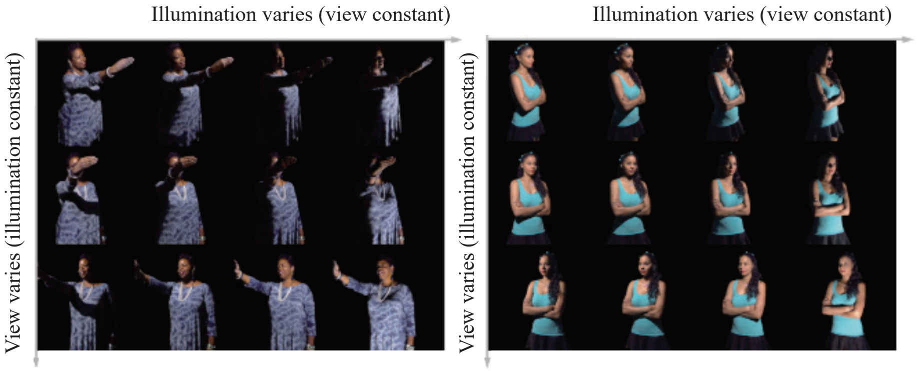

Illumination varies (view constant)
View varies (illumination constant)
Illumination varies (view constant)
View varies (illumination constant)

Fig. 22 Their model is able to perform simultaneous relighting and view synthesis, and it produces accurate renderings for unobserved viewpoints and light directions. Image taken from Zhang et al.[64]

entation rendering, global illumination rendering, direct illumination rendering, and human-related rendering, which embodies the rapid growth of deep learning-enhanced rendering methods. Deep learning-enhanced rendering has already shown impressive ability at real-time global illumination rendering, novel view synthesis, and relighting with only several images as input. We believe that the traditional graphics rendering pipeline can be partially or completely replaced by deep learning-enhanced rendering in the future. We hope that our report can provide researchers with a deep understanding of deep learning-enhanced rendering, and help them develop the next generation of deep learning-enhanced rendering and graphics applications.

Declarations of conflict of interest

The authors declared that they have no conflicts of interest to this work.

References

[1] C. Donner, H. W. Jensen. A spectral BSSRDF for shading human skin. In *Proceedings of the 17th Eurographics conference on Rendering Techniques*, Eurographics Association, Nicosia, Cyprus, pp. 409–417, 2006.

[2] L. Q. Yan, M. Hašan, B. Walter, S. Marschner, R. Ramamoorthi. Rendering specular microgeometry with wave optics. *ACM Transactions on Graphics*, vol. 37, no. 4, Article number 75, 2018. DOI: 10.1145/3197517.3201351.

[3] L. Q. Yan, W. L. Sun, H. W. Jensen, R. Ramamoorthi. A BSSRDF model for efficient rendering of fur with global illumination. *ACM Transactions on Graphics*, vol. 36, no. 6, Article number 208, 2017. DOI: 10.1145/3130800.3130802.

[4] E. Veach, L. J. Guibas. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, Los Angeles, USA, pp. 65–76, 1997. DOI: 10.1145/258734.258775.

[5] M. Pauly, T. Kollig, A. Keller. Metropolis light transport for participating media. In *Proceedings of Eurographics Workshop on Rendering Techniques,* Springer, Brno, Czech Republic, pp. 11–22, 2000. DOI: 10.1007/978-3-7091-6303-0_2.

[6] Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, J. Pantaleoni. ReSTIR GI: Path resampling for real-time path tracing. *Computer Graphics Forum*, vol. 40, no. 8, pp. 17–29, 2021. DOI: 10.1111/cgf.14378.

[7] M. McGuire, M. Mara, D. Nowrouzezahrai, D. Luebke. Real-time global illumination using precomputed light field probes. In *Proceedings of the 21st ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, San Francisco, USA, pp. 2, 2017. DOI: 10.1145/3023368.3023378.

[8] D. P. Fan, Z. L. Huang, P. Zheng, H. Liu, X. B. Qin, L. Van Gool. Facial-sketch synthesis: A new challenge. *Machine Intelligence Research*, vol. 19, no. 4, pp. 257–287, 2022. DOI: 10.1007/s11633-022-1349-9.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 2672–2680, 2014.

[10] M. Arjovsky, S. Chintala, L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, pp. 214–223, 2017.

[11] J. Y. Zhu, T. Park, P. Isola, A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of IEEE International Conference on Computer Vision*, Venice, Italy, pp. 2242–2251, 2017. DOI: 10.1109/ICCV.2017.244.

[12] D. P. Kingma, M. Welling. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Canada, 2013. DOI: 10.48550/arXiv.1312.6114.

[13] Y. C. Pu, Z. Gan, R. Henao, X. Yuan, C. Y. Li, A. Stevens, L. Carin. Variational autoencoder for deep learning of images, labels and captions. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Barcelona, Spain, pp. 2360–2368, 2016.

[14] A. Vahdat, J. Kautz. NVAE: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, vol. 33, pp. 19667–19679, 2020.

[15] P. Isola, J. Y. Zhu, T. H. Zhou, A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 5967–5976, 2017. DOI: 10.1109/CVPR.2017.632.

[16] S. M. A. Eslami, D. Jimenez Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, D. P. Reichert, L. Buesing, T. Weber, O. Vinyals, D. Rosenbaum, N. Rabinowitz, H. King, C. Hillier, M. Botvinick, D. Wierstra, K. Kavukcuoglu, D. Hassabis. Neural scene representation and rendering. *Science*, vol. 360, no. 6394, pp. 1204–1210, 2018. DOI: 10.1126/science.aar6170.

[17] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J. Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B. Goldman, M. Zollhöfer. State of the art on neural rendering. *Computer Graphics Forum*, vol. 39, no. 2, pp. 701–727, 2020. DOI: 10.1111/cgf.14022.

[18] C. Zhang, T. Chen. A survey on image-based rendering – representation, sampling and compression. *Signal Processing: Image Communication*, vol. 19, no. 1, pp. 1–28, 2004. DOI: 10.1016/j.image.2003.07.001.

[19] J. Y. Zhu, Z. T. Zhang, C. K. Zhang, J. J. Wu, A. Torralba, J. B. Tenenbaum, W. T. Freeman. Visual object networks: Image generation with disentangled 3D representation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Canada, pp. 118–129, 2018.

[20] T. H. Nguyen-Phuoc, C. Li, S. Balaban, Y. L. Yang. RenderNet: A deep convolutional network for differentiable rendering from 3D shapes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 7902–7912, 2018.

[21] K. Rematas, V. Ferrari. Neural voxel renderer: Learning an accurate and controllable rendering tool. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 5416–

5426, 2020. DOI: 10.1109/CVPR42600.2020.00546.

[22] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, M. Zollhöfer. Deepvoxels: Learning persistent 3D feature embeddings. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 2432–2441, 2019. DOI: 10.1109/CVPR.2019.00254.

[23] Y. Liao, K. Schwarz, L. Mescheder, A. Geiger. Towards unsupervised learning of generative models for 3D controllable image synthesis. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 5870–5879, 2020. DOI: 10.1109/CVPR42600.2020.00591.

[24] Z. Chen, A. P. Chen, G. L. Zhang, C. Y. Wang, Y. Ji, K. N. Kutulakos, J. Y. Yu. A neural rendering framework for free-viewpoint relighting. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 5598–5609, 2020. DOI: 10.1109/CVPR42600.2020.00564.

[25] J. Granskog, F. Rousselle, M. Papas, J. Novák. Compositional neural scene representations for shading inference. *ACM Transactions on Graphics*, vol. 39, no. 4, Article number 135, 2020. DOI: 10.1145/3386569.3392475.

[26] H. Kato, Y. Ushiku, T. Harada. Neural 3D mesh renderer. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 3907–3916, 2018. DOI: 10.1109/CVPR.2018.00411.

[27] A. P. Chen, M. Y. Wu, Y. L. Zhang, N. Y. Li, J. Lu, S. H. Gao, J. Y. Yu. Deep surface light fields. *Proceedings of ACM on Computer Graphics and Interactive Techniques*, vol. 1, no. 1, Article number 14, 2018. DOI: 10.1145/3203192.

[28] J. Thies, M. Zollhöfer, M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics*, vol. 38, no. 4, Article number 66, 2019. DOI: 10.1145/3306346.3323035.

[29] D. Gao, G. J. Chen, Y. Dong, P. Peers, K. Xu, X. Tong. Deferred neural lighting: Free-viewpoint relighting from unstructured photographs. *ACM Transactions on Graphics*, vol. 39, no. 6, Article number 258, 2020. DOI: 10.1145/3414685.3417767.

[30] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, R. Martin-Brualla. Neural rerendering in the wild. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 6871–6880, 2019. DOI: 10.1109/CVPR.2019.00704.

[31] K. A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, V. Lempitsky. Neural point-based graphics. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 696–712, 2020. DOI: 10.1007/978-3-030-58542-6_42.

[32] P. Dai, Y. D. Zhang, Z. W. Li, S. C. Liu, B. Zeng. Neural point cloud rendering via multi-plane projection. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 7827–7836, 2020. DOI: 10.1109/CVPR42600.2020.00785.

[33] P. Sanzenbacher, L. Mescheder, A. Geiger. Learning neural light transport, [Online], Available: https://arxiv.org/abs/2006.03427, 2020.

[34] M. Oechsle, M. Niemeyer, C. Reiser, L. Mescheder, T. Strauss, A. Geiger. Learning implicit surface light fields. In *Proceedings of International Conference on 3D Vision*, IEEE, Fukuoka, Japan, pp. 452–462, 2020. DOI: 10.1109/3DV50981.2020.00055.

[35] Q. Q. Wang, Z. C. Wang, K. Genova, P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, T. Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 4688–4697, 2021. DOI: 10.1109/CVPR46437.2021.00466.

[36] L. Yariv, J. T. Gu, Y. Kasten, Y. Lipman. Volume rendering of neural implicit surfaces. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 4805–4815, 2021.

[37] V. Sitzmann, M. Zollhöfer, G. Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, NeurIPS, Vancouver, Canada, pp. 101, 2019.

[38] M. Niemeyer, L. Mescheder, M. Oechsle, A. Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 3501–3512, 2020. DOI: 10.1109/CVPR42600.2020.00356.

[39] P. Kellnhofer, L. C. Jebe, A. Jones, R. Spicer, K. Pulli, G. Wetzstein. Neural lumigraph rendering. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 4285–4295, 2021. DOI: 10.1109/CVPR46437.2021.00427.

[40] O. Nalbach, E. Arabadzhiyska, D. Mehta, H. P. Seidel, T. Ritschel. Deep shading: Convolutional neural networks for screen space shading. *Computer Graphics Forum*, vol. 36, no. 4, pp. 65–78, 2017. DOI: 10.1111/cgf.13225.

[41] U. Erra, N. Capece, R. Agatiello. Ambient occlusion baking via a feed-forward neural network. In *Proceedings of European Association for Computer Graphics: Short Papers*, Lyon, France, pp. 13–16, 2017. DOI: 10.2312/egsh.20171003.

[42] D. J. Zhang, C. H. Xian, G. L. Luo, Y. H. Xiong, C. Han. DeepAO: Efficient screen space ambient occlusion generation via deep network. *IEEE Access*, vol. 8, pp. 64434–64441, 2020. DOI: 10.1109/ACCESS.2020.2984771.

[43] L. Ren, Y. Song. AOGAN: A generative adversarial network for screen space ambient occlusion. *Computational Visual Media*, vol. 8, no. 8, pp. 483–494, 2022. DOI: 10.1007/s41095-021-0248-2.

[44] C. Suppan, A. Chalmers, J. Zhao, A. Doronin, T. Rhee. Neural screen space rendering of direct illumination. In *Proceedings of the 29th Pacific Conference on Computer Graphics and Applications*, Pacific Graphics, Wellington, New Zealand, pp. 37–42, 2021.

[45] M. Mirza, S. Osindero. Conditional generative adversarial nets, [Online], Available: https://arxiv.org/abs/1411.1784, 2014.

[46] M. M. Thomas, A. G. Forbes. Deep illumination: Approximating dynamic global illumination with generative adversarial network, [Online], Available: https://arxiv.org/abs/1710.09834, 2017.

[47] T. Müller, F. Rousselle, A. Keller, J. Novák. Neural control variates. *ACM Transactions on Graphics*, vol. 39, no. 6, Article number 243, 2020. DOI: 10.1145/3414685.3417804.

[48] P. R. Ren, J. P. Wang, M. M. Gong, S. Lin, X. Tong, B. N. Guo. Global illumination with radiance regression functions. *ACM Transactions on Graphics*, vol. 32, no. 4, Article number 130, 2013. DOI: 10.1145/2461912.2462009.

[49] S. Hadadan, S. H. Chen, M. Zwicker. Neural radiosity. *ACM Transactions on Graphics*, vol. 40, no. 6, Article number 236, 2021. DOI: 10.1145/3478513.3480569.

[50] S. Diolatzis, J. Philip, G. Drettakis. Active exploration for neural global illumination of variable scenes. *ACM Transactions on Graphics*, vol. 41, no. 5, Article number 171, 2022. DOI: 10.1145/3522735.

[51] S. Kallweit, T. Müller, B. Mcwilliams, M. Gross, J. Novák. Deep scattering: Rendering atmospheric clouds with radiance-predicting neural networks. *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 1–11, 2017. DOI: 10.1145/3130800.3130880.

[52] M. Panin, S. Nikolenko. Faster RPNN: Rendering clouds with latent space light probes. In *Proceedings of SIGGRAPH Asia Technical Briefs*, ACM, Brisbane, Australia, pp. 21–24, 2019. DOI: 10.1145/3355088.3365150.

[53] F. Abbas, M. C. Babahenini. Babahenini. Forest fog rendering using generative adversarial networks. *The Visual Computer*, vol. 39, no. 3, pp. 943–952, 2023.

[54] Q. Zheng, G. Singh, H. P. Seidel. Neural relightable participating media rendering. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 15203–15215, 2021.

[55] P. Hermosilla, S. Maisch, T. Ritschel, T. Ropinski. Deep-learning the latent space of light transport. *Computer Graphics Forum*, vol. 38, no. 4, pp. 207–217, 2019. DOI: 10.1111/cgf.13783.

[56] D. Vicini, V. Koltun, W. Jakob. A learned shape-adaptive subsurface scattering model. *ACM Transactions on Graphics*, vol. 38, no. 4, Article number 127, 2019. DOI: 10.1145/3306346.3322974.

[57] L. Y. Wei, L. W. Hu, V. Kim, E. Yumer, H. Li. Real-time hair rendering using sequential adversarial networks. In *Proceedings of the 15th European Conference on Computer Vision*, Springer, Munich, Germany, pp. 105–122, 2018. DOI: 10.1007/978-3-030-01225-0_7.

[58] R. Martin-Brualla, R. Pandey, S. R. Yang, P. Pidlypenskyi, J. Taylor, J. Valentin, S. Khamis, P. Davidson, A. Tkach, P. Lincoln, A. Kowdle, C. Rhemann, D. B. Goldman, C. Keskin, S. Seitz, S. Izadi, S. Fanello. *LookinGood*: Enhancing performance capture with real-time neural re-rendering. *ACM Transactions on Graphics*, vol. 37, no. 6, Article number 255, 2018. DOI: 10.1145/3272127.3275099.

[59] A. Meka, R. Pandey, C. Häne, S. Orts-Escolano, P. Barnum, P. David-Son, D. Erickson, Y. D. Zhang, J. Taylor, S. Bouaziz, C. Legendre, W. C. Ma, R. Overbeck, T. Beeler, P. Debevec, S. Izadi, C. Theobalt, C. Rhemann, S. Fanello. Deep relightable textures: Volumetric performance capture with neural rendering. *ACM Transactions on Graphics*, vol. 39, no. 6, Article number 259, 2020. DOI: 10.1145/3414685.3417814.

[60] P. Chandran, S. Winberg, G. Zoss, J. Riviere, M. Gross, P. Gotardo, D. Bradley. Rendering with style: Combining traditional and neural approaches for high-quality face rendering. *ACM Transactions on Graphics*, vol. 40, no. 6, Article number 223, 2021. DOI: 10.1145/3478513.3480509.

[61] S. Lombardi, J. Saragih, T. Simon, Y. Sheikh. Deep appearance models for face rendering. *ACM Transactions on Graphics*, vol. 37, no. 4, Article number 68, 2018. DOI: 10.1145/3197517.3201401.

[62] L. J. Liu, W. P. Xu, M. Zollhöfer, H. Kim, F. Bernard, M. Habermann, W. P. Wang, C. Theobalt. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics*, vol. 38, no. 5, Article number 139, 2019. DOI: 10.1145/3333002.

[63] M. Y. Wu, Y. H. Wang, Q. Hu, J. Y. Yu. Multi-view neural human rendering. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 1679–1688, 2020. DOI: 10.1109/CVPR42600.2020.00175.

[64] X. M. Zhang, S. Fanello, Y. T. Tsai, T. C. Sun, T. F. Xue, R. Pandey, S. Orts-Escolano, P. Davidson, C. Rhemann, P. Debevec, J. T. Barron, R. Ramamoorthi, W. T. Freeman. Neural light transport for relighting and view synthesis. *ACM Transactions on Graphics*, vol. 40, no. 1, Article number 9, 2021. DOI: 10.1145/3446328.

[65] A. Edelsten, P. Jukarainen, A. Patney. Truly Next-GEN: Adding Deep Learning to Games & Graphics. In *Proceedings of Game Developers Conference Recording (GDC Vault)*, USA, 2019.

[66] L. Xiao, S. Nouri, M. Chapman, A. Fix, D. Lanman, A. Kaplanyan. Neural supersampling for real-time rendering. *ACM Transactions on Graphics*, vol. 39, no. 4, Article number 142, 2020. DOI: 10.1145/3386569.3392376.

[67] J. Guo, X. H. Fu, L. Q. Lin, H. J. Ma, Y. W. Guo, S. Q. Liu, L. Q. Yan. ExtraNet: Real-time extrapolated rendering for low-latency temporal supersampling. *ACM Transactions on Graphics*, vol. 40, no. 6, Article number 278, 2021. DOI: 10.1145/3478513.3480531.

[68] K. M. Briedis, A. Djelouah, M. Meyer, I. McGonigal, M. Gross, C. Schroers. Neural frame interpolation for rendered content. *ACM Transactions on Graphics*, vol. 40, no. 6, Article number 239, 2021. DOI: 10.1145/3478513.3480553.

[69] S. Bi, K. Sunkavalli, F. Perazzi, E. Shechtman, V. Kim, R. Ramamoorthi. Deep CG2Real: Synthetic-to-real translation via image disentanglement. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 2730–2739, 2019. DOI: 10.1109/ICCV.2019.00282.

[70] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the 16th European Conference on Computer Vision*, Springer, Glasgow, UK, pp. 405–421, 2020. DOI: 10.1007/978-3-030-58452-8_24.

[71] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P. P. Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 5835–5844, 2021. DOI: 10.1109/ICCV48922.2021.00580.

[72] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, D. Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 7206–7215, 2021. DOI: 10.1109/CVPR46437.2021.00713.

[73] K. Park, U. Sinha, P. Hedman, J. T. Barron, S. Bouaziz, D. B. Goldman, R. Martin-Brualla, S. M. Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Transactions on Graphics*, vol. 40, no. 6, Article number 238, 2021. DOI: 10.1145/3478513.3480487.

[74] A. Pumarola, E. Corona, G. Pons-Moll, F. Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 10313–10322, 2021. DOI: 10.1109/CVPR46437.2021.01018.

[75] S. Y. Su, F. Yu, M. Zollhöfer, H. Rhodin. A-NeRF: Articulated neural radiance fields for learning human shape, appearance, and pose. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 12278–12291, 2021.

[76] G. Gafni, J. Thies, M. Zollhöfer, M. Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Nashville, USA, pp. 8645–8654, 2021. DOI: 10.1109/CVPR46437.2021.00854.

[77] C. Gao, Y. Shih, W. S. Lai, C. K. Liang, J. B. Huang. Portrait neural radiance fields from a single image, [Online], Available: https://arxiv.org/abs/2012.05903, 2020.

[78] B. B. Yang, Y. D. Zhang, Y. H. Xu, Y. J. Li, H. Zhou, H. J. Bao, G. F. Zhang, Z. P. Cui. Learning object-compositional neural radiance field for editable scene rendering. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Montreal, Canada, pp. 13759–13768, 2021. DOI: 10.1109/ICCV48922.2021.01352.

[79] Y. C. Huo, S. E. Yoon. A survey on deep learning-based Monte Carlo denoising. *Computational Visual Media*, vol. 7, no. 2, pp. 169–185, 2021. DOI: 10.1007/s41095-021-0209-9.

[80] K. Gao, Y. N. Gao, H. J. He, D. N. Lu, L. L. Xu, J. Li. NeRF: Neural radiance field in 3D vision, a comprehensive review, [Online], Available: https://arxiv.org/abs/2210.00379, 2022.

[81] P. Wang, L. J. Liu, Y. Liu, C. Theobalt, T. Komura, W. P. Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 27171–27183, 2021.

[82] Z. H. Yu, S. Y. Peng, M. Niemeyer, T. Sattler, A. Geiger. MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction, [Online], Available: https://arxiv.org/abs/2206.00665, 2022.

[83] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, Dallas, USA, pp. 143–150, 1986. DOI: 10.1145/15922.15902.

[84] E. Veach. Robust Monte Carlo methods for light transport simulation, Ph. D. dissertation, Stanford University, Stanford, USA, 1998.

[85] M. Pharr, W. Jakob, G. Humphreys. *Physically Based Rendering: From Theory to Implementation*, 3rd ed., Cambridge, USA: Morgan Kaufmann, 2016.

[86] A. Dosovitskiy, J. T. Springenberg, T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern recognition*, Boston, USA, pp. 1538–1546, 2015. DOI: 10.1109/CVPR.2015.7298761.

[87] Y. Blau, T. Michaeli. The perception-distortion tradeoff. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 6228–6237, 2018. DOI: 10.1109/CVPR.2018.00652.

[88] X. E. Zhang, R. Ng, Q. F. Chen. Single image reflection separation with perceptual losses. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Salt Lake City, USA, pp. 4786–4794, 2018. DOI: 10.1109/CVPR.2018.00503.

[89] C. Atapattu, B. Rekabdar. Improving the realism of synthetic images through a combination of adversarial and perceptual losses. In *Proceedings of International Joint Conference on Neural Networks*, IEEE, Budapest, Hungary, pp. 1–7, 2019. DOI: 10.1109/IJCNN.2019.8852449.

[90] J. Johnson, A. Alahi, L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 694–711, 2016. DOI: 10.1007/978-3-319-46475-6_43.

[91] A. Rehman, Z. Wang. SSIM-based non-local means image denoising. In *Proceedings of the 18th IEEE International Conference on Image Processing*, Brussels, Belgium, pp. 217–220, 2011. DOI: 10.1109/ICIP.2011.6116065.

[92] J. Hwang, C. S. Yu, Y. Shin. SAR-to-optical image translation using SSIM and perceptual loss based cycle-consistent GAN. In *Proceedings of International Conference on Information and Communication Technology Convergence*, IEEE, Jeju, Republic of Korea, pp. 191–194, 2020. DOI: 10.1109/ICTC49870.2020.9289381.

[93] T. Karras, S. Laine, T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4396–4405, 2019. DOI: 10.1109/CVPR.2019.00453.

[94] L. Bavoil. Horizon-based Ambient Occlusion Using Compute Shaders, NVIDIA, USA, 2011.

[95] J. Wilhelms, A. Van Gelder. A coherent projection approach for direct volume rendering. *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 275–284, 1991. DOI: 10.1145/127719.122758.

[96] P. Kutz, R. Habel, Y. K. Li, J. Novák. Spectral and decomposition tracking for rendering heterogeneous volumes. *ACM Transactions on Graphics*, vol. 36, no. 4, Article number 111, 2017. DOI: 10.1145/3072959.3073665.

[97] B. Miller, I. Georgiev, W. Jarosz. A null-scattering path integral formulation of light transport. *ACM Transactions on Graphics*, vol. 38, no. 4, Article number 44, 2019. DOI: 10.1145/3306346.3323025.

[98] C. Donner, J. Lawrence, R. Ramamoorthi, T. Hachisuka, H. W. Jensen, S. Nayar. An empirical BSSRDF model. New Orleans Louisiana, USA, Article number 30, 2009. DOI: 10.1145/1576246.1531336.

[99] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, A. Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics*, vol. 38, no. 4, Article number 29, 2019. DOI: 10.1145/3306346.3322980.

[100] C. Crassin, F. Neyret, M. Sainz, S. Green, E. Eisemann. Interactive indirect illumination using voxel cone tracing.

Computer Graphics Forum, vol. 30, no. 7, pp. 1921–1930, 2011. DOI: 10.1111/j.1467-8659.2011.02063.x.

[101] Y. Tokuyoshi, S. Ogaki. Real-time bidirectional path tracing via rasterization. In *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, Costa Mesa, USA, pp. 183–190, 2012. DOI: 10.1145/2159616.2159647.

[102] O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, Munich, Germany, pp. 234–241, 2015. DOI: 10.1007/978-3-319-24574-4_28.

[103] S. R. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 190–198, 2017. DOI: 10.1109/CVPR.2017.28.

[104] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 8107–8116, 2020. DOI: 10.1109/CVPR42600.2020.00813.

[105] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, G. J. Brostow. Interpretable transformations with encoder-decoder networks. In *Proceedings of IEEE International Conference on Computer Vision*, Venice, Italy, pp. 5737–5746, 2017. DOI: 10.1109/ICCV.2017.611.

[106] J. L. Schönberger, J. M. Frahm. Structure-from-motion revisited. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp. 4104–4113, 2016. DOI: 10.1109/CVPR.2016.445.

[107] J. L. Schönberger, E. L. Zheng, J. M. Frahm, M. Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the 14th European Conference on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 501–518, 2016. DOI: 10.1007/978-3-319-46487-9_31.

[108] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018. DOI: 10.1109/TPAMI.2017.2699184.

[109] R. Q. Charles, S. Hao, K. C. Mo, L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, USA, pp. 77–85, 2017. DOI: 10.1109/CVPR.2017.16.

[110] K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, USA, pp. 770–778, 2016. DOI: 10.1109/CVPR.2016.90.

[111] G. H. Li, M. Müller, A. Thabet, B. Ghanem. DeepGCNs: Can GCNs go as deep as CNNs? In *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 9266–9275, 2019. DOI: 10.1109/ICCV.2019.00936.

[112] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Long Beach, USA, pp. 4455–4465, 2019. DOI: 10.1109/CVPR.2019.00459.

[113] M. Oechsle, L. Mescheder, M. Niemeyer, T. Strauss, A. Geiger. Texture fields: Learning texture representations in function space. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, IEEE, Seoul, Republic of Korea, pp. 4530–4539, 2019. DOI: 10.1109/ICCV.2019.00463.

[114] K. Zhang, G. Riegler, N. Snavely, V. Koltun. NeRF++: Analyzing and improving neural radiance fields, [Online], Available: https://arxiv.org/abs/2010.07492, 2020.

[115] Y. Yao, Z. X. Luo, S. W. Li, J. Y. Zhang, Y. F. Ren, L. Zhou, T. Fang, L. Quan. BlendedMVS: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Seattle, USA, pp. 1787–1796, 2020. DOI: 10.1109/CVPR42600.2020.00186.

[116] L. Yariv, Y. Kasten, D. Moran, M. Galun, M. Atzmon, R. Basri, Y. Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Vancouver, Canada, Article number 210, 2020.

[117] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, G. Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, vol. 33, no. 7, pp. 7462–7473, 2020.

[118] C. Buehler, M. Bosse, L. McMillan, S. Gortler, M. Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th Annual Conference on Computer graphics and Interactive Techniques*, ACM, Los Angeles, USA, pp. 425–432, 2001. DOI: 10.1145/383259.383309.

[119] V. Lepetit, F. Moreno-Noguer, P. Fua. EPnP: An accurate $O(n)$ solution to the PnP problem. *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009. DOI: 10.1007/s11263-008-0152-6.

[120] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H. P. Seidel, W. P. Xu, D. Casas, C. Theobalt. VNect: Real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics*, vol. 36, no. 4, Article number 44, 2017. DOI: 10.1145/3072959.3073596.

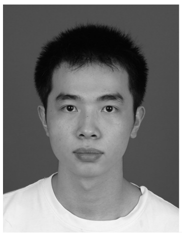
[121] C. R. Qi, L. Yi, H. Su, L. J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, USA, pp. 5105–5114, 2017.

Qi Wang received the B. Eng. and M. Eng. degrees in computer science and technology from Beijing Institute of Technology, China in 2017 and 2019, respectively. He is currently a Ph. D. degree candidate in computer graphics at State Key Laboratory of CAD&CG, Zhejiang University, China.

His research direction is human-related rendering.

E-mail: wqnina1995@gmail.com

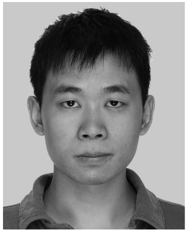
ORCID iD: 0000-0002-6326-3209



Zhihua Zhong received the B. Eng. degree in computer network from Jinan University, China in 2017. He is currently a master student in computer graphics at State Key Laboratory of CAD&CG, Zhejiang University, China.

His research direction is superresolution in real-time rendering.

E-mail: zhi564133873k@126.com



Yuchi Huo received the Ph. D. degree from State Key Laboratory of CAD&CG, Zhejiang University, China. He is a "Hundred Talent Program" researcher in State Key Laboratory of CAD&CG, Zhejiang University, China.

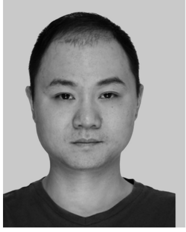
His research interests include rendering, deep learning, image processing, and computational optics, which are aiming for the realization of next-generation neural rendering pipeline and physical-neural computation.

E-mail: huoyc@zhejianglab.com

Hujun Bao received the B. Eng. and Ph. D. degrees in applied mathematics from Zhejiang University, China in 1987 and 1993, respectively. He is currently the deputy director of Zhejiang Laboratory and the deputy director of Informatics Department of the Science and Technology Committee of the Ministry of Education, China.

His research interests include rendering, modelling and virtual reality.

E-mail: bao@cad.zju.edu.cn



Rui Wang received the B. Sc. degree in computer science and the Ph. D. degree in mathematics from Zhejiang University, China in 2001 and 2007, respectively. He is currently a professor at State Key Laboratory of CAD&CG, Zhejiang University, China.

His research interests include real-time rendering, realistic rendering, GPU-based computation and 3D display techniques.

E-mail: wang.rui@acm.org (Corresponding author)

ORCID iD: 0000-0003-4267-0347

MIR | Machine Intelligence Research

Citation: Q. Wang, Z. Zhong, Y. Huo, H. Bao, R. Wang. State of the art on deep learning–enhanced rendering methods. *Machine Inteligence Research*, vol.20, no.6, pp.799–821, 2023. https://doi.org/10.1007/s11633–022–1400–x

Articles may interest you