State of the Art on Neural Rendering

- State of the Art on Neural Rendering
- **[v1]** Wed, 8 Apr 2020 04:36:31 UTC (3,192 KB)

1. Introduction

- Occlusion often occurs when two or more objects come too close and seemingly merge or combine with each other.
- Neural rendering brings the promise of addressing both *reconstruction and rendering* by using deep networks to learn complex mappings from captured images to novel images

The definition of *neural rendering* given in this paper:

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

4. Theoretical Fundamentals

4.1. Physical Image Formation

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

4.1.1. Scene Representations

> Representations can be classified into *explicit* and *implicit* representations.

- *Explicit representations* describe scenes as a collection of geometric primitives, such as triangles
    - In practice, most hardware and software renderers are tuned to work best on triangle meshes, and will **convert** other representations into triangles for rendering
- *Implicit representations* include signed distance functions mapping from $\mathbb{R}^3 \rightarrow \mathbb{R}$, such that the surface is defined as the zero-crossing of the function (or any other level-set)
- *Materials* may be represented as bidirectional reflectance distribution functions (BRDFs) or bidirectional subsurface scattering reflectance distribution functions (BSSRDFs)

    - a 5-D/4-D BRDF only models light interactions that happen at a single surface point
        - a spatially varying BRDF
        - Spatially varying behavior across geometry may be represented by binding *discrete materials* to different geometric primitives, or via the use of *texture mapping*
        - mapping is typically applicable only to explicit geometry
    - a 7-D/8-D BSSDRF models how light incident on one surface point is reflected at a different surface point

        > 这个很多地方的名字都不一样，但是缩写一样……大致意思就是光线从一个点入射，经过材料内部散射/吸收后，从另外一个点出射。还可以参考【PathTracing】实时光线追踪和BSSRDF的那些事

- *Sources of light* in a scene can be represented using parametric models
    - Some methods account for continuously varying emission over a surface, defined by a *texture map or function*
    - Often environment maps are used to represent *dense*, *distant scene lighting*

4.1.2. Camera Models

> The most common camera model in computer graphics is the *pinhole camera model*, in which rays of light pass through a pinhole and hit a film plane (image plane). Such a camera can be parameterized by the pinhole's 3D location, the image plane, and a rectangular region in that plane representing the spatial extent of the sensor or film

- The operation of such a camera can be represented compactly using projective geometry, which converts 3D geometric representations using *homogeneous coordinates* into the two dimensional domain of the image plane
    - known as a full perspective projection model, which is non-linear
- Approximations of this model such as the weak perspective projection are often used in computer vision to reduce complexity because of the non-linearity of the full perspective projection

4.1.3. Classical Rendering

> The process of transforming a scene definition including cameras, lights, surface geometry and material into a *simulated camera image* is known as rendering.

- The two most common approaches to rendering are *rasterization* and *raytracing*
    - *Rasterization* is a **feed-forward process** in which geometry is transformed into the image domain, sometimes in back-to-front order known as painter's algorithm
    - *Raytracing* is a process in which rays are cast backwards from the image pixels into a virtual scene, and reflections and refractions are simulated by **recursively casting** new rays from the intersections with the geometry
    - Renderers can also use combinations of rasterization and raycasting to obtain high efficiency and physical realism at the same time
- *rasterization*
    - Hardware-accelerated rendering typically relies on rasterization, because it has good memory coherence
    - requires an explicit geometric representation
    - implicit representations can also be converted to explicit forms for rasterization using the marching cubes algorithm

- *raytracing*
  - many real-world image effects are more easily simulated using raytracing
  - recent GPUs now feature acceleration structures to enable certain uses of raytracing in real-time graphics pipelines
  - can also be applied to implicit representations (suitable for both)
- *inverse rendering*
  - inverse problem in the context of computer vision and computer graphics
  - A drawback of inverse rendering is that the predefined physical model or data structures used in classical rendering don't always accurately reproduce all the features of real-world physical processes, due to either mathematical complexity or computational expense
  - neural rendering introduces learned components into the rendering pipeline in place of predefined physical models

4.1.4. Light Transport

> Light transport considers all the possible paths of light from the emitting light sources, through a scene, and onto a camera.

Classical function:

$$L_o(\mathbf{p}, \omega_o, \lambda, t) = L_e(\mathbf{p}, \omega_o, \lambda, t) + L_r(\mathbf{p}, \omega_o, \lambda, t) \tag{1}$$

- L_o: outgoing radiance, a function of location $\mathbf{p}$, ray direction ω_o, wavelength λ and time t
- L_e: direct surface emission
- L_r: interaction of incident light with surface reflectance

$$L_r(\mathbf{p}, \omega_o, \lambda, t) = \int_\Omega f_r(\mathbf{p}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{p}, \omega_i, \lambda, t)(\omega_i \cdot \mathbf{n}) \mathrm{d}\omega_i \tag{2}$$

- omits consideration of transparent objects and any effects of subsurface or volumetric scattering
- cannot be solved in closed form for non-trivial scenes
- the most accurate approximations employ *Monte Carlo* simulations

4.1.5. Image-based Rendering

> Image-based rendering techniques generate novel images by transforming an existing set of images, typically by warping and compositing them together

- the most common use-case is novel view synthesis of static objects

4.2. Deep Generative Models

> Compared to classical image-based rendering, which historically has used small sets of images (e.g., hundreds), deep generative models can learn image priors from **large-scale image collections**.

- photo-realistic image synthesis has been demonstrated using *Generative Adversarial Networks* (GANs) and its extensions

> inverse transform: https://www.cnblogs.com/mingyangovo/articles/14668731.html

- Deep generative models excel at generating *random* realistic images with statistics resembling the training set
    - user control and interactivity play a key role in image synthesis and manipulation, since a user may want particular scenes rather than random scenes
    - therefore, generative models need to be extended to a conditional setting to gain explicit control of the image synthesis process
- Instead of minimizing the distance between outputs and targets, which may produce unnatural images, *conditional GANs (cGANs)* aim to match the conditional distribution of outputs given inputs
- Conditional GANs have been employed to bridge the gap between coarse computer graphics renderings and the corresponding real-world images, or to produce a realistic image given a user-specified semantic layout

4.2.1. Learning a Generator

> We aim to learn a neural network G that can **map** a *conditional input* $x \in \mathcal{X}$ to an output $y \in \mathcal{Y}$. Here $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output domains. We call this neural network *generator*.

Three commonly-used generator architectures:

- *Fully Convolutional Networks (FCNs)*: a family of models that can take an input image with arbitrary size and predict an output with **the same size**
    - widely used for many image synthesis tasks
- *U-Net*: an FCN-based architecture with improved localization ability
    - skip connections help to produce more detailed outputs, since high-frequency information from the input can be passed directly to the output
- *ResNet-based generators*: use residual blocks to pass the high-frequency information from input to output
    - used in style transfer and image super-resolution

> 可以看到当时还是CNN、GAN和残差网络比较流行……这篇综述发了之后没多久就出现了大获成功的以MLP为主的NeRF（SRN于19年提出）

4.2.2. Learning using Perceptual Distance

> A learning objective *perceptual distance* that better aligns with human's perception of image similarity, which is to be minimized:
>
> $$\mathcal{L}_{prec}(G) = \mathbb{E}_{x,y} \sum_{t=1}^{T} \lambda_t \frac{1}{N}_t \|F^{(t)}(G(x)) - F^{(t)}(y)\|_1 \tag{1}$$
>
> - $F^{(t)}$: the feature extractor in the t-th layer of the pre-trained network F with T layers in total
> - N_t: the total number of features in layer t
> - λ_t: hyper-parameter denoting the weight of each layer
> - $G()$: generator

- such loss compares deep representations that **summarize an entire image holistically** rather than compares ℓ_p-norms that evaluates each pixel independently

- why such statistical representation matches human's perception remains to be further studied
- *per-pixel loss* in ℓ_p-norm: $\mathcal{L}_{recon}(G) = \mathbb{E}_{x,y}\|G(x) - y\|_p$
  - $\mathbb{E}$: expectation of the loss function over training pairs
  - tends to synthesize blurry images or average results over multiple plausible outputs

4.2.3. Learning with Conditional GANs

- Minimizing distances between output and ground truth does not guarantee realistic looking output, and small distance doesn't mean photorealism either
- deep generative models focus on matching the distribution of generated results to the distribution of training data
- *Generative Adversarial Networks* (GANs) have shown promising results for many computer graphics tasks
  - a GAN generator $G : z \rightarrow y$ learns a mapping from a *low-dimensional random vector* z to an *output image* y.
  - the input vector is typically sampled from a multivariate *Gaussian* or *Uniform distribution*
  - the generator G is trained to **produce** outputs that cannot be distinguished from "real" images by an adversarially trained discriminator D, which is trained to **detect** synthetic images generated by the generator
  - the quality of objects is usually good but the quality of background is usually poor
- *conditional GANs* (cGANs) learn a mapping $G : \{x, z\} \rightarrow y$ from an observed input x and a randomly sampled vector z to an output image y
  - objective function:
    $\min_G \max_D \mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$

    > 判别器：希望真实图像的判别结果接近1，希望生成图像的判别结果接近0
    >
    > 生成器：希望生成图像的判别结果接近1
    >
    > 目标函数的左半期望即是真实图像的判别结果（能认出真的东西是真的），右半期望即是生成图像的判别结果（能认出假的东西是假的）
    >
    > x是标签，z即噪声输入。判别器不仅仅是判断图片的真假，还要判断图片和标签是否一致。这意味着生成器还得根据标签生成图片，这就实现了conditioning

  - to stabilize training, cGANs-based methods also adopt per-pixel loss and perceptual distance loss
- In a high-level abstraction, an accurate computer vision model (F or D) for assessing the quality of synthesized results $G(x)$ can
  significantly help tackle neural rendering problems
  - perceptual distance aims to measure the *discrepancy* between an output instance and its ground truth, while conditional GANs measure the closeness of the conditional *distributions* of real and fake images
  - for perceptual distance, the feature extractor is **pre-trained and fixed**, while conditional GANs **adapt** its discriminator on the fly
  - two methods are complementary and are usually combined to use in practice

4.2.4. Learning without Paired Data

> Paired/Labelled data is expensive!

- The model is given a source set $\{x_i\}_{i=1}^N (x_i \in \mathcal{X})$ and a target set $\{y_j\}_{j=1} (y_j \in \mathcal{Y})$
- All we know is which target *domain* the output $G(x)$ should come from, like an image from domain $\mathcal{Y}$
- given a particular input, we do not know which target *image* the output should be, which means there can be infinitely many mappings — **additional constraints are necessary**
    - cycle-consistency loss for enforcing a bijective mapping
    - distance preserving loss for encouraging that the output is close to the input image either in pixel space or in feature embedding space

        > 保持两个输入的距离在映射之后不变，即对应的两个输出的距离应该接近两个输入之间的距离

    - weight sharing strategy for learning shared representation across domains

        > 在多个域的学习中，如联合分布，使用共享一部分权重的策略来使得多个GAN可以在高级抽象中保持一致/共享latent code/...

5. Neural Rendering

> Deep generative networks are now starting to produce visually compelling images and videos either from random noise, or conditioned on certain user specifications like scene segmentation and layout. However, they **do not yet** allow for fine-grained control over scene appearance and cannot always handle the complex, non-local, 3D interactions between scene properties. In contrast, neural rendering methods **hold the promise** of combining these approaches to enable controllable, high-quality synthesis of novel images from input images/videos.
>
> A typical neural rendering approach takes as input **images** corresponding to certain scene conditions (for example, viewpoint, lighting, layout, etc.), builds a "*neural*" scene representation from them, and "*renders*" this representation under novel scene properties to **synthesize novel images**. The learned scene representation is not restricted by simple scene modeling approximations and can be optimized for high quality novel images.

Approaches may be classified by:

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

5.1. Control

A main axis in which the approaches differ is in how the control signal is **provided** to the network:

- directly pass the scene parameters as input to the first or an intermediate network layer

- tile/concatenate scene parameter
- rely on the spatial structure of images and employ an image-to-image translation network to map from a "*guide image*" or "*conditioning image*" to the output

5.2. Computer Graphics Modules

- One emerging trend in neural rendering is the integration of computer graphics knowledge into the network design. Therefore, approaches might differ in the level of "*classical*" graphics knowledge that is embedded in the system
- Classical graphics components add a **physically inspired inductive bias** to the network, while still allowing for end-to-end training via backpropagation. This can be used to analytically **enforce a truth about the world** in the network structure, **frees up network capacity**, and leads to **better generalization**, especially if only limited training data is available

5.3. Explicit vs. Implicit Control

- explicit control: a user can edit the scene parameters manually in a **semantically meaningful manner**
    - usually requires training datasets with images/videos and their corresponding scene parameters
- implicit control: by way of a representative sample. While approaches can copy the scene parameters from a reference image/video, one cannot manipulate these parameters explicitly
    - usually requires less supervision

5.4. Multi-modal Synthesis

- To achieve various outputs of various modals which are significantly different from each other the network or control signals must have some stochasticity or structured variance
    - variational auto-encoders (VAEs)
        - https://arxiv.org/abs/1312.6114
        - https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73
        - https://www.jianshu.com/p/0dac27698013
        - https://blog.csdn.net/kittyzc/article/details/124682802
        - https://blog.csdn.net/u012856866/article/details/121497294

> In this paper we will therefore also refer to the recognition model $q_\phi(\mathbf{z}|\mathbf{x})$ as a *probabilistic encoder*, since given a data point $\mathbf{x}$ it produces a distribution (e.g. a Gaussian) over the possible values of the code $\mathbf{z}$ from which the data point $\mathbf{x}$ could have been generated. In a similar vein we will refer to $p_\theta(\mathbf{x}|\mathbf{z})$ as a *probabilistic decoder*, since given a code $\mathbf{z}$ it produces a distribution over the possible corresponding values of $\mathbf{x}$.

5.5. Generality

- Some methods aim to train a *general purpose model* once, and apply it to all instances of the task at hand
- other methods are *instance-specific*

6. Applications of Neural Rendering

Better to refer to the table in the original paper

6.1. Semantic Photo Synthesis and Manipulation

> *Semantic photo synthesis and manipulation* enable interactive image editing tools for controlling and modifying the appearance of a photograph in a semantically meaningful way

- Semantic Photo Synthesis
- Semantic Image Manipulation
- Improving the Realism of Synthetic Renderings

6.2. Novel View Synthesis for Objects and Scenes

> *Novel view synthesis* is the problem of generating novel camera perspectives of a scene given a fixed set of images of the same scene. Novel view synthesis methods thus deal with image and video synthesis conditioned on camera pose.

Key challenges underlying novel view synthesis are **inferring the scene's 3D structure** given sparse observations, as well as **inpainting of occluded and unseen parts** of the scene

- *Image-based rendering* (IBR) methods typically rely on optimization-based multi-view stereo methods to reconstruct scene geometry and warp observations into the coordinate frame of the novel view
    - if only few observations are available, the scene contains view-dependent effects, or a large part of the novel perspective is not covered by the observations, IBR may fail, leading to results with ghosting-like artifacts and holes

    > 个人理解：只是组合，没有重建
    >
    > https://www.docin.com/p-1061204404.html

- In *Neural Image-based Rendering*, previously hand-crafted parts of the IBR pipeline are replaced or augmented by learning-based methods
- Other approaches reconstruct a learned representation of the scene from the observations, learning it end-to-end with a differentiable renderer
- Neural rendering methods are restricted to a specific use-case and are limited by the training data. Especially, view-dependent effects such as reflections are still challenging.

6.2.1. Neural Image-based Rendering

> *Neural Image-based Rendering* (N-IBR) is a hybrid between classical image-based rendering and deep neural networks that replaces hand-crafted heuristics with learned components.

- A classical IBR method uses a set of captured images and a proxy geometry to create new images
    - The proxy geometry is used to **reproject** image content from the captured images to the new target image domain.
    - In the target image domain, the projections from the source images are blended to **composite** the final image.

    - This simplified process gives accurate results only for **diffuse** objects with precise geometry reconstructed with a **sufficient** number of captured views
    - N-IBR often learns blending functions or corrections that take into account view-dependent effects

6.2.2. Neural Rerendering

> *Neural Rerendering* combines classical 3D representation and **renderer with deep neural networks** that rerender the classical render into a more complete and realistic views

- Neural rerendering does not use input views at runtime, and instead relies on the deep neural network to **recover the missing details**
- *Neural Rerendering in the Wild* takes as input a rendered deep buffer, containing depth and color channels, together with an appearance code, and outputs realistic views of the scene
- Semantic labels can be used to deal with transient objects
- Pittaluga et al. use a neural rerendering technique to invert Structure-from-Motion reconstructions and highlight the privacy risks of Structure-from-Motion 3D reconstructions, that typically contain color and SIFT features. Sparse point clouds can be inverted and generate realistic novel views from them

> Q: 这段话根本看不懂……有空看看原文

6.2.3. Novel View Synthesis with Multiplane Images

- Given a sparse set of input views of an object, Xu et al. also address the problem of rendering the object from novel view points. Unlike previous view interpolation methods that work with images captured under natural illumination and at a small baseline, they aim to capture the light transport of the scene, including view-dependent effects like specularities. Moreover they attempt to do this from a sparse set of images captured at large baselines, in order to make the capture process more light-weight
    - To handle the occlusions caused by the large baseline, they propose **predicting attention maps** that capture the visibility of the input viewpoints at different pixels. These attention maps are used to modulate the appearance plane sweep volume and remove inconsistent content

> Q: baseline可能指两个图像的摄像机之间的距离……没搜到解释……这篇论文拍摄的角度是招财猫正面的那个半球

- *DeepView* is a technique to visualize light fields under novel views

6.2.4. Neural Scene Representation and Rendering

> While neural rendering methods based on multi-plane images and image-based rendering have enabled some impressive results, they prescribe the model's internal representation of the scene as a point cloud, a multi-plane image, or a mesh, and do not allow the model to learn an **optimal representation** of the scene's geometry and appearance.

- A scene is represented by a collection of observations, where each observation is a tuple of an image and its respective camera pose
- The *Generative Query Network* is a framework for learning a low-dimensional feature embedding of a scene, explicitly modeling the stochastic nature of such a neural scene representation due to incomplete observations

6.2.5. Voxel-based Novel View Synthesis Methods

> A line of neural rendering approaches has emerged that instead proposes to represent the scene as a voxel grid, thus enforcing 3D structure

- Learned, unstructured neural scene representations disregard the natural 3D structure of scenes and thus fail to discover multi-view and perspective geometry in regimes of limited training data
- *RenderNet* proposes a convolutional neural network architecture that implements differentiable rendering from a scene explicitly represented as a 3D voxel grid.
- *DeepVoxels* enables joint reconstruction of geometry and appearance of a scene and subsequent novel view synthesis.
- *Visual Object Networks (VONs)* is a 3D-aware generative model for synthesizing the appearance of objects with a disentangled 3D representation
- *HoloGAN* builds on top of the learned projection unit of *RenderNet* to build an unconditional generative model that allows explicit viewpoint changes

6.2.6. Implicit-function based Approaches

> Implicit-function based approaches model geometry as the *level set* of a neural network

- 3D voxel grids' memory requirement scales cubically with spatial resolution, and they do not parameterize surfaces smoothly, requiring a neural network to learn priors on shape as joint probabilities of neighboring voxels
    - cannot parameterize large scenes at a sufficient spatial resolution, and have so far failed to generalize shape and appearance across scenes
- *Pixel-Aligned Implicit Functions* represent object color via an implicit function.
    - An image is first encoded into a pixel-wise feature map via a convolutional neural network.
    - A fully connected neural network then takes as input the feature at a specific pixel location as well as a depth value *z*, and classifies the depth as inside/outside the object.
    - The same architecture is used to encode color
- *Scene Representation Networks* (SRNs) encodes both scene geometry and appearance in a single fully connected neural network (MLP) that maps world coordinates to a feature representation of local scene properties
    - The SRN takes as input (x, y, z) world coordinates and computes a feature embedding.
    - To render an image, camera rays are traced to their intersections with scene geometry (if any) via a differentiable, learned raymarcher, which computes the length of the next step based on the feature returned by the SRN at the current intersection estimate.
    - The SRN is then **sampled** at ray intersections, yielding a feature for every pixel.
    - SRNs generalize across scenes in the same class by representing each scene by a code vector $\mathbf{z}$. The code vectors $\mathbf{z}$ are mapped to the parameters of a SRN via a fully connected neural network, a so-called **hypernetwork**.

6.3. Free Viewpoint Videos

> *Free Viewpoint Videos*, also known as *Volumetric Performance Capture*, rely on multi-camera setups to acquire the 3D shape and texture of performers

LookinGood with Neural Rerendering

- The *LookinGood* system by Martin-Brualla et al. introduced the concept of neural rerendering for performance capture of human actors. \
- The framework relies on a volumetric performance capture system, which reconstructs the performer in real-time.
- methods:
    - rerender
    - image-to-image translation

Neural Volumes

- *Neural Volumes* addresses the problem of automatically creating, rendering, and animating high-quality object models from multi-view video data
    - warp field not only helps to model the motion of the scene but also reduces blocky voxel grid artifacts by **deforming** voxels to better match the geometry of the scene and allows the system to shift voxels to make better use of the voxel resolution available

Free Viewpoint Videos from a Single Sensor

- Cost of obtaining multi-view images is still expensive. Parallel efforts try to make the capture technology accessible through consumer hardware by dropping the infrastructure requirements through deep learning
- Reconstructing performers from a single image is very related to the topic of synthesizing humans in unseen poses

6.4. Learning to Relight

> Photo-realistically rendering of a scene under novel illumination—a procedure known as "relighting"—is a fundamental component of a number of graphics applications including compositing, augmented reality and visual effects

- An effective way to accomplish this task is to use image-based relighting methods that take as input **images of the scene captured under different lighting conditions** (also known as a "reflectance field"), and **combine** them to render the scene's appearance under novel illumination
    - require slow data acquisition with expensive, custom hardware, precluding the applicability of such methods to settings like dynamic performance and "in-the-wild" capture

Deep Image-based Relighting from Sparse Samples

- An image-based relighting method that can relight a scene from a sparse set of five images captured under learned, **optimal** light directions was proposed
    - uses a deep convolutional neural network to regress a relit image under an arbitrary directional light from these five images
    - by training a *non-linear* neural relighting network, this method is able to accomplish relighting from sparse images
    - The relighting quality depends on the input light directions, and the authors propose combining a *custom-designed sampling network* with the relighting network, in an end-to-end fashion, to jointly learn both the optimal input light directions and the relighting function

Multi-view Scene Relighting

- Given multiple views of a large-scale outdoor scene captured under uncontrolled natural illumination, a method can render the scene under novel outdoor lighting (parameterized by the sun position and cloudiness level)
    - The input views are used to reconstruct the 3D geometry of the scene. This geometry to construct intermediate buffers—normals, reflection features, and RGB shadow maps—as auxiliary inputs to guide a neural network-based relighting method
    - also uses a shadow refinement network to improve the removal and addition of shadows
    - While the entire method is trained on a synthetically rendered dataset, it generalizes to real scenes

Deep Reflectance Fields

- *Deep Reflectance Fields* presents a novel technique to relight images of human faces by learning a model of facial reflectance from a database of 4D reflectance field data of several subjects in a variety of expressions and viewpoints
    - The high-quality results of the method indicate that the *color gradient images* contain the information needed to estimate the full 4D reflectance field, including specular reflections and high frequency details

Single Image Portrait Relighting

- The relighting model in these methods consists of a deep neural network that has been trained to take a single RGB image as input and produce as output a relit version of the portrait image under an arbitrary user-specified environment map.
    - Additionally, the model also predicts an estimation of the current lighting conditions
    - can run on mobile devices

6.5. Facial Reenactment

> *Facial reenactment* aims to modify scene properties beyond those of viewpoint and lighting

Deep Video Portraits

- *Deep Video Portraits* is a system for full head reenactment of portrait videos
    - The head pose, facial expressions and eye motions of the person in a video are transferred from another reference video.
    - A facial performance capture method is used to compute 3D face reconstructions for both reference and target videos.
    - This reconstruction is represented using a **low-dimensional** semantic representation which includes identity, expression, pose, eye motion, and illumination parameters
    - A rendering-to-video translation network, based on U-Nets, is trained to convert classical computer graphics renderings of the 3D models to photo-realistic images
    - The training data consists of pairs of training frames, and their corresponding 3D reconstructions

Editing Video by Editing Text

- *Text-based Editing of Talking-head Video* takes as input a one-hour long video of a person speaking, and the **transcript** of that video

    - The editor changes the transcript in a text editor, and the system synthesizes a new video in which the speaker appears to be speaking the revised transcript
    - Spoken snippets in the training data will be used
    - Each video frame is converted to a low-dimensional representation
    - Neural rendering is used to convert the low-fidelity render into a photo-realistic frame. A GAN-based encoder-decoder, is trained to add high frequency details and hole-fill to produce the final result

Image Synthesis using Neural Textures

- *Deferred Neural Rendering* enables novel-view point synthesis as well as scene-editing in 3D
    - scene-specific/object-specific
    - Besides ground truth color images, it requires a **coarse** reconstructed and tracked 3D mesh including a **texture parametrization**

Neural Talking Head Models

- a *generalized* face reenactment approach
    - train a common network to control faces of any identity using sparse 2D keypoints, which consists of an embedder network to extract pose-independent identity information
    - the output of the network is fed to a generator network

Deep Appearance Models

- *Deep Appearance Models* model facial geometry and appearance with a conditional variational autoencoder
    - VAE is conditioned on the viewpoint of the camera, which enables the decoder network to correct geometric tracking errors by decoding a texture map that reprojects to the correct place in the rendered image
    - this method is a system for animating the learned face model from cameras mounted on a virtual reality headset
    - challenge
        - Full head reenactment, including control over the head pose, is very challenging in dynamic environments.
        - Many of the methods discussed do not preserve high-frequency details in a temporally coherent manner.
        - Photorealistic synthesis and editing of hair motion and mouth interior including tongue motion is challenging.
        - Generalization across different identities without any degradation in quality is still an open problem

6.6. Body Reenactment

> Neural pose-guided image and video generation enables the control of the **position**, **rotation**, and **body pose** of a person in a target image/video

- challenging, due to the large non-linear motion space of humans
- Full body performance cloning approaches
    - transfer the motion in a source video to a target video
    - person-specific

- A method predicts two consecutive frames of the output video and employs a space-time discriminator for more temporal coherence
- A method employs a network with two separate branches that is trained in a hybrid manner based on a mixed training corpus of paired and unpaired data.
    - The paired branch employs paired training data extracted from a reference video to directly supervise image generation based on a reconstruction loss.
    - The unpaired branch is trained with unpaired data based on an adversarial identity loss and a temporal coherence loss
- *Textured Neural Avatars* predict dense texture coordinates based on rendered skeletons to sample a learnable, but static, RGB texture.
    - no need for geometry

Summary

一些方法：

- 跳过物理方法，直接从输入映射到输出，相当于数据驱动吧
- 首先从学习经验中生成一个不那么好的结果，再利用一个GAN或者神经网络添加细节
- 用多个数据结构保存多种信息
- 组合方法总是依赖大数据，而学习方法又总是有很多artifacts

7. Open Challenges

7.1. Generalization

> In a way, approaches learn to interpolate between the training examples

- One solution to achieve better generalization is to explicitly **add the failure cases to the training corpus**
    - comes at the expense of network capacity and all failures cases might not be known a priori
    - if many of the scene parameters have to be controlled, the curse of dimensionality makes capturing all potential scenarios infeasible
    - if a solution should work for arbitrary people, we cannot realistically gather training data for all potential users and such success cannot be guaranteed
- One possibility to improve generalization is to explicitly **build a physically inspired inductive bias into the network**
- To explore how **additional information at test time** can be employed to improve generalization

7.2. Scalability

> Scalability is additionally needed to successfully process complex, cluttered, and large scenes, for example to enable dynamic crowds, city- or global-scale scenes to be efficiently processed

- Software engineering and improved use of available computational resources

- Let the network reason about compositionality
    - it has to be able to segment a scene into objects, understand local coordinate systems, and robustly process observations with partial occlusions or missing parts
    - steps towards unsupervised learning strategies have to be developed

7.3. Editability

> Neural rendering approaches today do not always offer editability

- it is important to achieve an intuitive way to edit abstract feature based representations
- it is important to understand and reason about the network output

7.4. Multimodal Neural Scene Representations

- A network that uses both visual and audio as input may learn useful ways to process the additional input modalities
- Many applications demand multimodal outputs

8. Social Implications

- Neural rendering approaches have the potential to lower the barrier for entry, making manipulation technology accessible to non-experts with limited resources
- It is critical that synthesized images and videos **clearly present themselves as synthetic**
- It is essential to obtain permission from the content owner and/or performers for any alteration before sharing a resulting video
- It is important that we as a community continue to develop forensics, fingerprinting and verification techniques (digital and non-digital) to identify manipulated video
- Researchers must also employ responsible disclosure when appropriate, carefully considering how and to whom a new system is released

Forgery Detection

- The verification of the integrity of an image can be done using a pro-active protection method, like digital signatures and water marking, or a passive forensic analysis
- *Secure Digital Camera*: not only introduces a watermark but also stores a biometric identifier of the person who took the photograph
    - failed to implement in hardware
- *Manipulation-specific detection methods* learn to detect the artifacts produced by a specific manipulation method
- *Manipulation-independent methods* concentrate on image plausibility

Q

1. 为什么GAN要以噪声为输入?

往往是假设数据服从一个高斯分布，便于进行两个分布之间的映射，实现生成数据的目的

2. 为什么VAE要使用随机变量?

假设隐藏变量的先验分布，可以更好地计算或逼近棘手的后验概率（intractable posterior）

3. 为什么要在生成模型中引入随机变量?

生成模型是生成数据的模型，维基百科说

> a **generative model** is a model of the conditional probability of the observable *X*, given a target *y*, symbolically $P(X|Y=y)$

引入随机变量的目的很多：

- 引入噪声，增强鲁棒性，如Denoising Auto-Encoders
- 假设隐藏变量的先验分布，可以更好地计算或逼近棘手的后验概率（intractable posterior），如Variational Auto-Encoders
- 假设一个连续分布，方便进行分布之间的映射，以生成数据，如GAN

4. VAE里面的先验到底是什么用？和编码器的后验概率是什么关系?

- 为什么提出VAE：AE的隐藏空间不连续，无法实现随机生成数据，也不对隐藏空间作任何要求。为了实现生成数据的功能，提出一个具有连续隐藏空间的VAE。encoder相当于实现了数据的降维和泛化
- 如何生成数据：训练VAE（encoder+decoder），然后给decoder一个服从连续分布的输入即可生成数据
- 损失函数：$||x-x'|| + KL(N(\mu,\sigma),N(0,1))$，左半部分是期望生成的数据要与观察（输入）的数据接近，右半部分是希望学习到的后验概率接近于（人类假设的）正态分布。注意这里是后验概率接近于先验，这是经过公式变换的结果（公式的初始表达是两个后验概率逼近）
- prior $p(z)$：一般取标准正态分布，即人类假设隐藏空间为一个正态分布，优化的时候需要用到隐藏空间的分布（即用到先验）。设置高斯分布可以使得隐藏空间的分布更加规范，可以方便地生成一个输入以生成数据
- encoder $q_\phi(\mathbf{z}|\mathbf{x})$：代表着如何将输入空间映射到隐藏空间，encoder的学习目标是使得encoder学习到的后验概率逼近真正的后验概率
- decoder $p_\theta(\mathbf{x}|\mathbf{z})$：代表着如何将隐藏空间映射到输出空间，decoder的学习目标也就是1)encoder学习的后验概率逼近真正的后验概率；2)隐藏空间的分布逼近先验；当decoder是一个高斯分布时，其方差是不变的
- 学习时会为每一个输入数据都生成一个正态分布，而这个正态分布又会随着学习逼近标准正态分布

5. Level Set

- 表面定义为 $S=\{\vec{x}|\phi(\vec{x},t)=k\}$
  - t为时间
  - $\vec{x}$为坐标
  - $\phi()$即为水平集函数
- 将N维的表面映射到N+1维的函数上去，通过令这个函数取特定值来表示一个表面
- 适合于tracking interface/curve（随时间演变的）
- 在reconstruction中：表示出geometry/shape的SDF，取SDF为水平集函数，SDF为0（0-level set）时得到的geometry/shape即为需要重建的geometry/shape
  - NeuS

    - 可以理解为，一个三维的面被铺平到了一个二维的平面上，形成了SDF
- https://www.youtube.com/watch?v=RDUH2412ZU0