NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis

- https://arxiv.org/abs/2012.03927
- [v1] Mon, 7 Dec 2020 18:56:08 UTC (22,848 KB)

1 Introduction

- NeRF: only works for novel view synthesis and does not provide a solution for relighting
  - NeRF only models the amount of outgoing light but **ignoring the interaction between incoming light and the material**
  - the sampling method is too expensive
  - limited to controlled settings which required input images to be illuminated by a single point light
- NeRV: visibility MLP + volume density MLP
  - visibility MLP: estimate **visibility** and **expected termination depth** in any direction

2 Related Work

> NeRV can be thought of as a neural analogue to visibility precomputation techniques, and is specifically designed for use in our neural inverse rendering setting where geometry is dynamically changing during optimization

- NeRF does not enable relighting because it has no mechanism to disentangle the outgoing radiance of a surface into an incoming radiance and an underlying surface material
- strategies for inverse rendering:
  - learning priors on shape, illumination, and reflectance
  - assuming known geometry
  - using multiple input images of the scene under different lighting conditions
    - NeRF, NeRV
- NeRF in the Wild conditions the output appearance on a latent code that encodes a per-image lighting
  - cannot be used to render the same scene under new lighting conditions not observed during training because it does not utilize the physics of light transport
  - just record the result but not record the principle

3 Method

> NeRV represent a scene as a 3D field of oriented particles that absorb and reflect the light emitted by external light sources

3.1 NeRF Overview

- $(\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$
- fixed illumination
- expression of radiance in this paper:
  - $L(\mathbf{c}, \omega_0) = \int_0^\infty V(\mathbf{x}(t), \mathbf{c})\sigma(\mathbf{x}(t))L_e(\mathbf{x}(t), \omega_0)dt$
  - $V(\mathbf{x}(t), \mathbf{c}) = \exp(-\int_0^t \sigma(\mathbf{x}(s))ds)$
  - ω_0: viewing direction
  - $\mathbf{x}(t) = \mathbf{c} - t\omega_0$: a point along the ray
- 直接用MLP得到在图片的固定光照下的某个点的颜色L_e

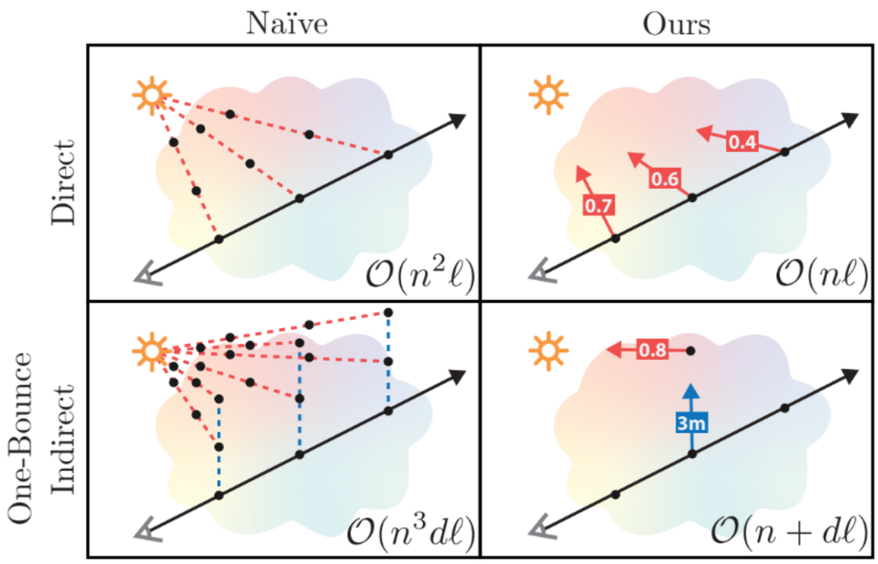
3.2 Neural Reflectance Fields

> Neural Reflectance Fields represents a field of particles that reflect incoming light, a PBR method

- account for both scene geometry and reflectance

- a single point light
- expression of radiance in this paper:
    - $L(\mathbf{c}, \omega_0) = \int_0^\infty V(\mathbf{x}(t))\sigma(\mathbf{x}(t))L_r(\mathbf{x}(t), \omega_0)dt$
    - $V(\mathbf{x}(t), \mathbf{c}) = \exp(-\int_0^n \sigma(\mathbf{x}(s))ds)$
    - $L_r(\mathbf{x}, \omega_o) = \int_{\mathcal{S}} L_i(\mathbf{x}, \omega_i)R(\mathbf{x}, \omega_i, \omega_o)d\omega_i$
    - since there is only one single point light, L_r can be simplified as $L_r(\mathbf{x}, \omega_o) = f_r(\mathbf{x}, \omega_o, \omega_i, \mathbf{n}(\mathbf{x}), \mathbf{R}(\mathbf{x}))L_i(\mathbf{x}, \omega_i) = f_r(\mathbf{x}, \omega_o, \omega_i, \mathbf{n}(\mathbf{x}), \mathbf{R}(\mathbf{x}))\tau_1(\mathbf{x})L_1(\mathbf{x})$
    - τ_1: the transmittance from the light to the shading point
    - $\mathbf{l}$: the point light source
    - f_r: a differentiable reflectance model with parameter $\mathbf{R}$
    - $\mathbf{n}$: the local surface shading normal, which can be inferred from the shape MLP itself $\mathbf{n} = \nabla_{\mathbf{x}} MLP_\theta(\mathbf{x}(t))$
- L_e in the NeRF equation is replaced with an integral over the sphere of incoming directions, which means it is calculated rather than approximated
- 通过记录反射的信息来得到不同位置的单一点光源下的某个点的颜色L_r
    - 可以指定一个新的点光源

3.3 & 3.4 Light Transport via Neural Visibility Fields & Rendering



> 3.1，3.2和3.6都是讨论建模/reconstruction的事情，这里主要是渲染/rendering的问题

- if conventional brute-force method is used for **indirect illumination** or **multiple-light-source illumination**, it is way too expensive
- method: **use learned approximations rather than brute-force volume density integrals**, which is implemented as a visibility MLP
- visibility MLP $MLP_\phi: (\mathbf{x}, \omega) \to (V_\phi, D_\phi)$
    - $V(\mathbf{x}, \omega) = \exp\left(-\int_0^\infty \sigma(\mathbf{x} + s\omega)ds\right)$, environment lighting visibility
    - $D(\mathbf{x}, \omega) = \int_0^\infty \exp\left(-\int_0^\infty \sigma(\mathbf{x} + s\omega)ds\right)t\sigma(\mathbf{x} + t\omega)dt$, expected termination depth
- shading function of a single point
    - decompose reflected light L_r into **direct light** (reflecting light from light source) L_e and **indirect light** (reflecting light from other location/particles) L
    - $$L_r(\mathbf{x}, \omega_o) = \int_{\mathcal{S}} \big(L_e(\mathbf{x}, \omega_i) + L(\mathbf{x}, -\omega_i)\big)R(\mathbf{x}, \omega_i, \omega_o)d\omega_i = \int_{\mathcal{S}} L_e(\mathbf{x}, \omega_i)R(\mathbf{x}, \omega_i, \omega_o)d\omega_i + \int_{\mathcal{S}} L(\mathbf{x}, -\omega_i)R(\mathbf{x}, \omega_i, \omega_o)d\omega_i$$
- direct
    - with environment map E, L_e can be written as $L_e(\mathbf{x}, \omega_i) = V(\mathbf{x}, \omega_i)E(\mathbf{x}, -\omega_i)$, in which V **should calculated as an integral originally**
    - with visibility MLP's approximation $\tilde{V}_\phi$, direct lighting of a point can be written as $\int_{\mathcal{S}} \tilde{V}_\phi(\mathbf{x}, \omega_i)E(\mathbf{x}, -\omega_i)R(\mathbf{x}, \omega_i, \omega_o)d\omega_i$
    - full calculation of a camera ray: $\int_0^\infty V(\mathbf{x}(t), \mathbf{c})\sigma(\mathbf{x}(t)) \int_{\mathcal{S}} \tilde{V}_\phi(\mathbf{x}(t), \omega_i)E(\mathbf{x}(t), -\omega_i)R(\mathbf{x}(t), \omega_i, \omega_o)d\omega_i dt$
- indirect
    - first approximation: replace the single point evaluation by **treating the volume as a hard surface located at the expected termination depth** t', and we have indirect shading function at the point $\int_{\mathcal{S}} L(\mathbf{x}(t'), -\omega_i)R(\mathbf{x}(t'), \omega_i, \omega_o)d\omega_i$

> 也就是，计算t'处会进入摄像头的光（光从光源经过t''再到t'再到摄像头）

  - to avoid infinite recursive evaluation, limit the indirect light to be one-bounce
  - second approximation: use the hard surface approximation again to replace the integral along the ray $L(\mathbf{x}(t'), -\omega_i) \approx \int_S L_e(\mathbf{x}'(t''), \omega_i') R(\mathbf{x}'(t''), \omega_i', -\omega_i) d\omega_i'$

    > 也就是，计算间接光的根源，t''处折射的光

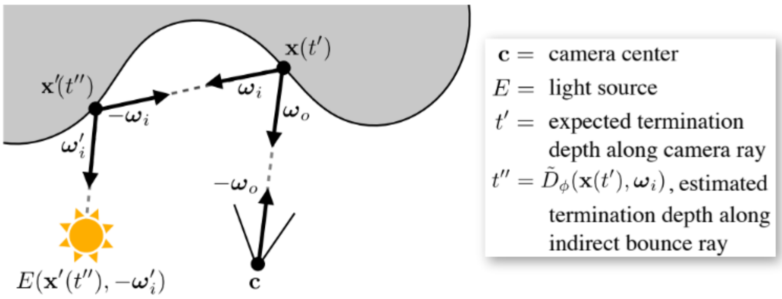


Figure 4: The geometry of an indirect illumination path from camera to light source, and a visualization of our notation.

- combine together:

$$L(\mathbf{c}, \boldsymbol{\omega}_o) = \int_0^\infty V(\mathbf{x}(t), \mathbf{c}) \sigma(\mathbf{x}(t)) \int_S \tilde{V}_\phi(\mathbf{x}(t), \boldsymbol{\omega}_i) E(\mathbf{x}(t), -\boldsymbol{\omega}_i) R(\mathbf{x}(t), \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) d\boldsymbol{\omega}_i dt$$
$$+ \iint_S \tilde{V}_\phi(\mathbf{x}'(t''), \boldsymbol{\omega}_i') E(\mathbf{x}'(t''), -\boldsymbol{\omega}_i') R(\mathbf{x}'(t''), \boldsymbol{\omega}_i', -\boldsymbol{\omega}_i) d\boldsymbol{\omega}_i' R(\mathbf{x}(t'), \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) d\boldsymbol{\omega}_i \quad (14)$$

- in real rendering, use stratified sampling rather than calculating real integrals

> 这个hard surface到底有哪些性质？把线性方向上的积分变成球面？

3.5 Training and Implementation Details

- positional encoding
  - 2^7 for 3D coordinates and 2^4 for 3D direction vectors
- shape MLP: 8 fully connected ReLU layers with 256 channels
- reflectance MLP: 8 fully connected ReLU layers with 256 channels
- visibility MLP:
  - 8 fully connected ReLU layers with 256 channels to map encoded $\mathbf{x}$ to an 8D feature vector
  - concatenate the 8D feature vector with the encoded ω
  - 4 fully-connected ReLU layers with 128 channels

3.6 NeRV

> NeRF本质上是将场景建模为往外发射光线的实体，而NeRV则是将场景当做可以反射光线的实体。由于物体颜色的本质就是光照的结果，因此NeRV就单独对密度和反射光照建模（NeRF是对密度和颜色联合建模），密度和NeRF是没啥区别的，而反射光照则是基于BRDF的反射方程，即要求对应的MLP预测出物体表面的粗糙程度和3D反射率。另外，值得一提的是，物体表面的法线是可以由密度信息推理出来的。

- shape MLP $MLP_\theta: \mathbf{x} \to \sigma$
  - σ: volume density
- reflectance MLP $MLP_\psi: \mathbf{x} \to (\mathbf{a}, \gamma)$
  - $\mathbf{a}$: 3D diffuse albedo, which is adopted from a paper
  - γ: a scalar specular roughness, which is adopted from a paper
- visibility MLP $MLP_\phi: (\mathbf{x}, \omega) \to (V_\phi, D_\phi)$
  - $V(\mathbf{x}, \omega) = \exp\left(-\int_0^\infty \sigma(\mathbf{x} + s\omega) ds\right)$, environment lighting visibility
  - $D(\mathbf{x}, \omega) = \int_0^\infty \exp\left(-\int_0^\infty \sigma(\mathbf{x} + s\omega) ds\right) t\sigma(\mathbf{x} + t\omega) dt$, expected termination depth
- loss
  - at each training iteration
    - 512 pixel rays for pixel radiance $\sum_{\mathbf{r} \in \mathcal{R}} \left\| \tau(\tilde{L}(\mathbf{r})) - \tau(L(\mathbf{r})) \right\|_2^2 +$

- 256 rays for the expected visibility and expected depth

  $$\lambda \sum_{\mathbf{r}' \in \mathcal{R}' \cup \mathcal{R}, t} \left(\left\| \tilde{V}_\phi(\mathbf{r}'(t)) - V_\theta(\mathbf{r}'(t)) \right\|_2^2 + \left\| \tilde{D}_\phi(\mathbf{r}'(t)) - D_\theta(\mathbf{r}'(t)) \right\|_2^2 \right)$$

  - $$\mathcal{L} = \sum_{\mathbf{r} \in \mathcal{R}} \left\| \tau(\tilde{L}(\mathbf{r})) - \tau(L(\mathbf{r})) \right\|_2^2 + \qquad (16)$$
    $$\lambda \sum_{\mathbf{r}' \in \mathcal{R}' \cup \mathcal{R}, t} \left(\left\| \tilde{V}_\phi(\mathbf{r}'(t)) - V_\theta(\mathbf{r}'(t)) \right\|_2^2 + \left\| \tilde{D}_\phi(\mathbf{r}'(t)) - D_\theta(\mathbf{r}'(t)) \right\|_2^2 \right),$$

    where $\tau(x) = x/1+x$ is a tone-mapping operator [13], $L(\mathbf{r})$ and $\tilde{L}(\mathbf{r})$ are the ground truth and predicted camera ray radiance values (ground-truth values are simply the colors of input image pixels), $\tilde{V}_\phi(\mathbf{r})$ and $\tilde{D}_\phi(\mathbf{r})$ are the predicted visibility and expected termination depth from our visibility MLP given its current weights ϕ, $V_\theta(\mathbf{r})$ and $D_\theta(\mathbf{r})$ are the estimates of visibility and termination depth implied by the shape MLP given its current weights θ, and $\lambda = 20$ is the weight of the loss terms encouraging the visibility MLP to be consistent with the shape MLP. Note that the visibility MLP is not supervised using any "ground truth" visibility or termination depth — it is only optimized to be consistent with the NeRV's current estimate of scene geometry, by evaluating Equations 5 and 6 using the densities σ emitted by the shape MLP_θ. We apply a "stop gradient" to V_θ and

4 Results

> 总得来说就是看得出效果好了很多，但是地板渲染得没有真实图片那么平整（是把低频信号也当成高频信号了吗），以及还是有一些虚影

Other Resources

1. NeRF及其发展
2.