

NeRF相关文章导读

一些基础概念：

隐式神经表达：[（原博客）](#)

传统用于隐式表示的方法有：代数表示，函数表示，水平集表示等。

例如对基础图形通过交并补进行计算，可以达到复杂的图形。同时，用一些解析函数也可以用来表示形状，简单的就是椭圆的函数表达式。例如，符号距离函数也常用来表示三维形状，如图通过提取零值点可以将三维模型的表面提取出来。

而隐式神经表示（也成为基于坐标的表示）是一种对各种信号进行参数化的新方法。传统的信号表示通常是离散的，而隐式神经表示将信号参数化为一个连续函数，将信号的域映射到该坐标上的属性的值（例如对于图像，就是将像素坐标映射到R,G,B颜色）。当然，这些函数通常不是解析性的——即不可能写下将自然图像参数化为数学公式的函数。隐式神经表示通过神经网络来近似信号函数。

优点：

1. **表示不在与空间分辨率相互耦合**：例如，图像与像素数量像耦合的方式。隐式神经表示是连续函数，因此参数化信号所需的内存与分辨率无关，只与底层信号的复杂性相关。同时，因为隐式表达的具有“无限分辨率”——它们可以在任意空间分辨率下进行采样，所以隐式表示解决了在显示表示中显存的显示，使得超分辨率变成可能。
2. **表征能力更强**：神经隐式表示通过学习神经网络权重上的先验，可以用过不同的坐标输入映射到任意属性值。
3. **泛化性好**：学习神经网络权重上的先验，根据提取的先验以及输入的特征，可以得到不同的目标信号。
4. **易于学习**：基于神经网络的隐式神经表示易于与各种网络结构结合，能够快速地学习先验，拟合学习对象。与传统的表示形式相比，神经隐式表征可以灵活地融合到可微分的基于学习的管道中。

常用的网络结构：mlp

The Prelude: Neural Implicit Surfaces

A promising recent direction in computer vision is encoding objects and scenes in the weights of an MLP that directly maps from a 3D spatial location to an implicit representation of the shape.

使用[多层感知机](#)的权重对物体和场景进行编码，直接从3D空间位置映射到形状的隐含表示。

神经体积渲染的直接前驱是使用神经网络来定义隐含的表面表示的方法。许多3D感知的图像生成方法使用体素、网格、点云或其他表示。

1. Occupancy Networks：基于学习函数空间的三维重建表示方法[（原文链接）](#)：

学习模型的方法能够编码3D形状空间中丰富的先验信息，这有助于解决输入的模糊性。生成模型的方法在高分辨率的图像上已经取得了嗯好的效果，但是还没有复制到3D领域。与2D领域相比，暂时还是没有就3D的输出表示达成一致，这种表示既能提高内存效率，又能从数据中有效推断。

现存的表示方法大概有三类：

体素：体素表示是直接像素一般化的情况，随着分辨率的提高，这种方法的内存占用将会呈指数增长，因此限制了分辨率。

点云：点云缺少底层网格的连接结构，从模型中提取3D几何需要额外的过程。

网格：网格首先被考虑用于区分三维分类或分割任务，在网格的顶点和边跨越的图上应用卷积，最近网格也被应用于三维重建的表示方法。不幸的是，大部分方法倾向于产生自交叉的网格，并且只能产生简单的拓扑结构。

占用网络：用神经网络来近似这个3D函数，该神经网络分配给每一个p一个在0到1之间的占用概率，这个玩过相当于一个二分类的神经网络，而我们关注的是对象表面的决策边界。

当使用占用网络对物体进行三维重建时，必须以输入作为条件。

represent surfaces as 3D occupancy fields and use a numerical method to find the surface intersection for each ray, then calculate an exact derivative using implicit differentiation.

Occupancy Networks 将整个曲面表示为三维占位场，用数值方法找到每条射线的曲面交点，然后用隐式微分法计算出使用隐式微分法计算精确导数。

在cvpr2019上，[IM-NET](#)结合了五个resnet模块，获得了特征向量，一个3D点和一个预测的二值化占用率。

2. signed distance function

通过全连接层从一个三维的向量获得标记距离，或者是将三维点重新投射到一个像素对齐的特征表示中。

Building on Implicit Functions

有许多方法是在隐藏函数的最顶端来构建的，并普遍去训练2D图片。许多的结构化的隐层函数被提出，例如场景表达网络，微分体积渲染，隐层可微渲染以及NASA。

SRN基于DEEPSDF(signed distance function),增加了一个可微调的光线行进算法，以找到学习到的隐性表面的最接近的交点，并增加了一个MLP来回归颜色，使其能够从多个姿势的图像中学习。

可微几何渲染网络展示了隐式场景表示可以与可微分的渲染器结合起来，使其可以从图像中训练。他们还是提出了一个巧妙地技巧，使隐含表面的深度计算可以微分，而不是使用体积上的积分。

Neural Volume Rendering

神经集合渲染与occupancy和SDF式网络的共同点是，MLP被用作从三维坐标到标量或多变量场的函数，因此这些方法有时也被称为基于坐标的场景表示网络。

Neural Volumes

*[we]propose using a volumetric representation consisting of opacity and color at each position in 3D space, where rendering is realized through integral projection. During optimization, this semi-transparent representation of geometry disperses gradient information along the ray of integration, effectively widening the basin of convergence, enabling the discovery of good solutions**.*

三维空间的每个位置使用由不透明和颜色组成，这可以通过整体投影实现渲染。在优化过程中，这种半透明的几何体表示方法将梯度的形成沿着整合的射线分散开来，有效展开了收敛的方法。

Nerf

A NeRF stores a volumetric scene representation as the weights of an MLP, trained on many images with known pose.

nerf的方法忽略了signed distance function，而是基于**色彩和密度**来进行渲染。然后他们使用一种（容易微分的）数值积分方法来近似于真正的体积渲染步骤。

nerf能够实现优秀的渲染效果是因为它编码了一个3D点，并且在一条射线上的相关视图方向使用了使用周期性**激活函数**，即傅里叶特征。

其实，nerf的逻辑很简单，就是用过一个多层感知机输入一个五维向量并且输出它的颜色和密度，但是它也有很多改进的空间：

- 它的速度很慢，无论是训练还是渲染。
- 它只能表现静态场景
- It “bakes in” lighting（会曝光？）
- 经过训练的NeRF表示法不能推广到其他场景/物体上。

2021 ICCV基于nerf的渲染策略：

Fundamentals

[Mip-NeRF](#) 改编了图形学中的mip-map思想，通过沿观察射线的锥形部分进行积分来取代对光场的采样，从而解决了vanilla NeRF的严重混叠问题。

[MVSNerF](#) 训练了一个模型，其能够跨越很多的场景并且渲染新的视角，并且基于很少的几个姿势的试图来渲染新的试图。使用中间的体素化特征来编码要渲染的体积。

[DietNeRF](#) 通过语义损失监督NeRF的训练过程，通过使用CLIP评估任意的视图来创建，所以它可以从任意类别的单一视图学习中学习NeRF。

[UNISURF](#)提出通过占用网络来代替nerf中的密度，并且通过root-finding层次化的采样。这样就能够进行体积和表层渲染，并且得到了一个有所提升的几何形状。

[NerfingMVS](#)使用来自SfM管道的稀疏深度图来训练一个特定场景的深度网络，随后指导NeRF的自适应采样策略。

Performance

因为nerf的训练和渲染速度都非常缓慢，因此很多的文章希望能够去加速nerf的速度，其中的大部分其中在渲染的过程中。

[FastNeRF](#) 将nerf的体积渲染等式分解成了两个分支，他们结合去得到一个和nerf相同的结果，但是允许更有效的缓存，产生3000倍的速度。

[PlenOctrees](#) 使用球面谐波来模拟与视图有关的颜色，然后将其压缩成一个类似八角形的数据结构。

[SNeRG](#) 预先计算并将NeRF “烘烤”成一个新的稀疏神经辐射网格（SNeRG）表示，从而实现实时渲染。

[RtS](#) 专注于使用快速的 “surface NeRF”或sNeRF渲染器，对包括NeRF在内的各种表面表示法进行高效、正确的衍生渲染。

Pose-free

另一个加速nerf的方法就是去移除姿势监督的需求

[BARF](#) 通过捆绑场景的方式由粗到细地对同时对场景和相机姿势进行优化

[SCNeRF](#) 和BARF很相似，但还对内在因素进行优化，包括径向失真和每像素非线性失真

[GNeRF](#) 提出了一个粗略的初始姿势网络。它自己使用了一个GAN风格的网络来训练GARF，并且解决了初始化困难的问题。

Contional

还有一种优化nerf的方法是在各种latent代码上调节类似NeRF的模型。

[GRF](#) 更接近于基于图像的渲染，在测试时只使用少数图像。GRF在一个典型的空间而不是在视图空间中运行。

[GSN](#) 是一个场景生成器，它接受一个全局代码，并将其转化为一个局部代码的网格，每个代码都与一个局部辐射度模型相关。一个小的convnet有助于提高最终输出的比例。

[GANcraft](#) 将一个语义信息模块时间转化为一个像素绑定的NeRF-models的集合，其使得逼真图像的渲染对应一“Minecraft时间，这是给予一个风格隐藏函数。（我的世界渲染到真实情境的那种）

[CodeNeRF](#) 训练一个GRAF式的条件NeRF（一个形状和外观潜伏代码），然后在推理时对潜伏代码和物体姿势进行优化。

2022CVPR

[LOLNeRF](#) 使用了一个姿势估计和语义分割技术在单一视角上训练一个有条件的Nerf网络，这个网络可以在推理时生成不同的用相同的姿势生成不同的面部，或者用相同的面部生成不同的姿势。

[Pix2NeRF](#) 在 π -GAN中扩展了一个编码器，与GAN共同训练，允许将图像映射回潜流形，允许使用单一输入图像进行以物体为中心的新视图合成。

[StylizedNeRF](#) 预训练了一个基于3D场景的标准的Nerf网络来风格化，使用一个风格网络替代它的颜色预测模块，以此来获得一个风格化的Nerf

Composition

[EditNeRF](#) 在构成新的nerf场景方面有一些非常有趣的事情。学习了一个特殊类别的conditional NeRF模型。它是由GRAF启发的，但是多了一个实例无关的分支，并且展示各种策略，以互动方式编辑颜色和形状。

[ObjectNeRF](#) 训练了一个 像素编辑网络反馈两个路径：场景和物体。通过调整像素编码还会被移动、克隆或删除。

2022CVPR

[AutoRF](#) 对于给定的类和物体来学习了外观和先验外形，来获得对于新视角的合成的为给定的一类物体学习外观和形状先验，以便为新的视图合成进行单镜头重建。

[PNE](#) 适应了一个分离的Nerf来个性化物体实力，并创建一个泛光辐射场，能够通过结合多实例Nerf和单个“stuff”-NeRF来渲染动态场景

Dynamic

[NeRFlow](#) 是一个并行的工作，它学习了一个单一的一致连续的时空辐射场，它被约束以产生跨空间和时间的一致4D视图合成

[NR-NeRF](#) 也使用变形MLP对非刚性场景进行建模。它不依赖预先计算的场景。

[AD-NeRF](#) 从带有音频的短视频中训练出一个条件性的nerf，并且将DeepSpeech的特征和头部姿势串联到输入中，实现新的音频驱动的合成以及输入片段的编辑。

[DynamicVS](#) 则是挑战了自由视点视频合成的问题，使用了场景流和许多的规则化的结果来产生令人印象深刻的结果。

[NARF](#) 使用姿势监督来训练每个铰接部分的小型局部占用网络，然后用来调制一个有条件训练的NeRF模型。

[AnimatableNeRF](#) 使用来自mocap数据和多视角视频的跟踪骨架来训练基于骨架的混合场，然后转换辐射场，实现骨架驱动的人的头像合成。

[SRT](#) 处理了一个来自于新的领域的有姿势或者是无姿势的RGB图像，推断出一个“集合-静态”，并合成一个新的视图。所有这些的操作都是在一个前馈通道中完成。

Priors (2022 CVPR)

这是一个改善新视角的合成的重要方法，其采用了各种形式的通用或深度驱动的先验

[Dense Depth Priors for NeRF](#) 使在Sfm点云上的深度完成网络来估计深度，在输入图像稀疏的场景中产生更高的图像质量。

[Depth-supervised NeRF](#) 也使用了一个深度完成网络在结构-运动点云上面来提升一个深度监督损失，这是为了使用更少的场景来更快的训练

[InfoNeRF](#) 通过射线熵正则化，惩罚NeRF在有限输入视角的场景中过度拟合射线密度的行为，从而在渲染新视角时产生更高质量的深度图。

[RapNeRF](#) 集中试图一致上来实现试图扩展。它使用了两种技术，随机射线投射和射线图集。

[RegNeRF](#) 通过渲染在不可见场景中的patches和最小化外观和深度先验平滑，在试图图像中实现了良好的重建。

Multi-View

该方法是在推断时使用附近的参考视图

[GeoNeRF](#) 使用了特征金字塔网络 and 同位素翘曲在输入图像上来重构级联体积（），推断出局部的几何和外观在新的是叫上，基于了transformer的方法。

[Light Field Neural Rendering](#) 使用了一个对于目标像素及其外延段使用光场参数化。使用了新颖的transformer结构生成一个高质量的渲染结果。

[NAN](#) 基于IBRNet and NeRF实现了突发denoising（增加噪声？），现在是应对低光成像条件的标准方式。

[NeRFusion](#) 首先使用了临近的视图重构了每一个视角的局部体积特征，并且使用递归处理来构建全局神经体积。

2022-performance

一个新的趋势是体现出了基于体素的，非常迅速的NeRF网络，许多的放弃了原始nerf核心的巨大MLP

[DVGO](#) 用一个体素网格代替了一些大型的MLP，直接存储了不透明度和局部的颜色特征，插值后放进一个小的MLP里面产生与视图相关的颜色。

[EfficientNeRF](#) 从图片的特征中学习评估固体的几何，有效地在物体表面采样，减少需要渲染的时间并改进辐射场的构建。

[Fourier PlenOctrees](#) 对于动态场景有效的神经建模和实时的渲染，使用了“Fourier PlenOctrees”，比起nerf实现了3000倍的加速

[Plenoxels](#) 抛弃了全部的多层感知机，并且在3d体素上优化了不透明度和视线相关的颜色（使用了球面谐波）

[Point-NeRF](#) 使用了MVS技术来获得密集点云，它对于每一个点的特征使用，然后他们被送入一个小的MLP进行体积渲染。

Large-scale

[Block-NeRF](#) 缩放NeRF来渲染城市规模的场景。这个工作把场景分解到独立训练的NeRFs里面，然后把他们结合在一起去渲染出整个场景。

[Mega-NeRF](#) 将一个大场景分解成单元，每个单元都是一个独立的nerf，这样使得大场景的重构能够显著的比先前的方法用时要少。

[Urban Radiance Fields](#) 通过补偿光度影响和用基于激光雷达的深度监督模型训练，允许使用全景图和激光雷达信息对城市环境进行精确的三维重建。

Articulated

第二个高潮就是神经辐射场对于一些铰链式模型的应用，类似于人或者是猫猫

[BANMo](#) 结合可变形形状模型、典型嵌入和NeRF风格的体积渲染，从许多随意的RGB视频中训练出高保真的、有关节的3D模型。

[DoubleField](#) 使用了共享的特征编嵌入，训练了一个表层领和辐射场，能通过有限的输入视图进行高保真的人体重建和渲染。

[HumanNeRF](#) 优化了对于人在可变姿势的几何表达，并且通过非刚性和骨骼成分对一个运动场进行估计。

[HumanNeRF \(2\)](#) 通过动态的NeRF方法和来自相邻视图的神经外观混合模型来估计人体的几何形状和外观，以创建动态的自由视点视频，使用的输入视图只有六个。

[NeuralHOFusion](#) 学习从RGBD图像提取出来的稀疏的输入掩码分离出人类和物体的模型，尽管有遮挡物和具有挑战性的姿势，仍能得到真实的自由视角视频。

[Structured Local Radiance Fields](#) 使用姿势估计来建立一套特定于SMPL模型上的节点的局部辐射场，当与外观嵌入相结合时，可以产生真实的三维动画。

[Surface-Aligned NeRF](#) 将一个查询坐标映射到其在预先定义的人类网格上的分散投影点上，利用网格本身和视图方向输入到NeRF，进行高质量的动态渲染。

[VEOs](#) 使用非刚性NeRF的多视图变体来进行物体重建和跟踪plushy物体，然后可以以新的变形状态进行渲染。

Portrait

有些文章聚焦在可控制的面部图像或者3D头部模型生成。

[EG3D](#) 是一个几何感知的GAN，它使用一个新颖的三面体表示（介于隐含和体素之间），允许实时渲染到低分辨率的图像，通过超级分辨率进行放大。

[FENeRF](#) 学会了一种带有两个潜伏代码的3D感知人脸表示法，它可以生成可编辑的、与视图一致的2D人脸图像。

[GRAM](#)使用一个单独的流形预测网络来约束表面附近的体积渲染样本，产生高质量的结果和精细的细节。

[HeadNeRF](#) 将2D渲染整合到NeRF的渲染过程中，以40帧/秒的速度渲染可控的头像。

[RigNeRF](#) 通过使用一个由三维可变形脸部模型引导的变形场，能够完全控制从单个人像视频中学习到的头部姿势和面部表情。

[StyleSDF](#) 结合了一个有条件的SDFNet, 一个Nerf-style的几何渲染和一个2D风格转换网络来生成高质量的面部模型和图像。

Editable

可控制或可编辑的NerFs是密切相关的。

[CLIP-NeRF](#) 支持通过他们的CLIP嵌入，使用文本或图像指导编辑一个条件模型。

[CoNeRF](#) 基于一个单一的视频，和一些属性注释，并允许在独立控制属性和视角的情况下进行重新渲染。

[NeRF-Editing](#) 根据其更新的几何形状弯曲射线的方向，在被编辑的组件周围创建一个连续的变形场，这样够编辑从Nerf输出的重构网格。

other

[Dream Fields](#) 从文字说明中合成了一个Nerf，最小化基于CLIP的损失并且正则化了透射率来减少伪影

[DyNeRF](#) 使用了紧凑的隐编码来表现3D视频中的帧，它能够渲染自由视角的场景，并且有impressive的体积渲染效果。

[Kubric](#) 不能算是一个Nerf文章，但是提出了一个开源的python框架，使用pybullet和blender对接来生成逼真的场景，者能够直接给Nerf pipelines 提供训练数据。

[NICE SLAM](#) 使用层次化的体素网格NeRF变体来渲染RGBD，一个实时和可扩展的并行跟踪和映射密集SLAM系统 用于RGBD输入。

[STEM-NeRF](#) 为扫描 透射电子显微镜 提供了一个可微的图像形成模型