GeoNeRF

> GeoNeRF: Generalizing NeRF with Geometry Priors

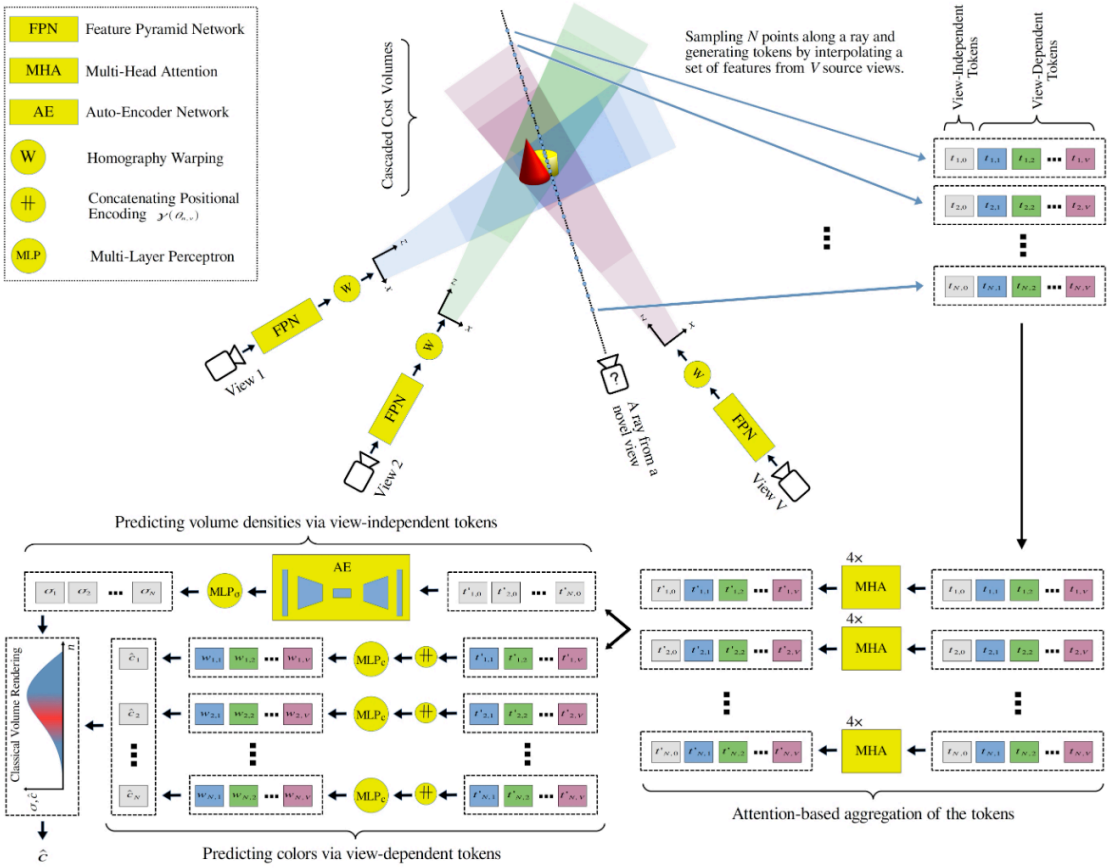

Figure 2. The overview of GeoNeRF. 2D feature pyramids are first generated via Feature Pyramid Network (FPN) [31] for each source view v. We then construct cascaded cost volumes at three levels for each view by homography warping of its nearby views (see Section 3.1). Guided by the distribution of the cascaded cost volumes in the 3D space, $N = N_c + N_f$ points $\{x_n\}_{n=1}^{N}$ are sampled along a ray for a novel pose (see Section 3.2). By interpolating both 2D and 3D features ($\boldsymbol{f}_{n,v}^{(0)}$, $\{\boldsymbol{\Phi}_{n,v}^{(l)}\}_{l=0}^{2}$) from FPN and cascaded cost volumes for each sample point x_n, one view independent token $\boldsymbol{t}_{n,0}$ and V view-dependent tokens $\{\boldsymbol{t}_{n,v}\}_{v=1}^{V}$ are generated. These $V+1$ tokens go through four stacked Multi-Head Attention (MHA) layers and yield more refined tokens $\{\boldsymbol{t}'_{n,v}\}_{v=0}^{V}$. The MHA layers are shared among all sample points on a ray. Thereafter, the view-independent tokens $\{\boldsymbol{t}'_{n,0}\}_{n=1}^{N}$ are regularized and aggregated along the ray samples through the AE network, and volume densities $\{\boldsymbol{\sigma}_n\}_{n=1}^{N}$ of the sampled points are estimated. Other tokens $\{\boldsymbol{t}'_{n,v}\}_{v=1}^{V}$, supplemented with the positional encodings $\{\gamma(\theta_{n,v})\}_{v=1}^{V}$, predict the color weights $\{\boldsymbol{w}_{n,v}\}_{v=1}^{V}$ with respect to source views, and the color $\hat{\boldsymbol{c}}_n$ of each point is estimated in a weighted sum fashion (see Section 3.3). Finally, the color of the ray $\hat{\boldsymbol{c}}$ is rendered using classical volume rendering.

概述

本文提出了GeoNeRF，一个结合几何先验的通用化NeRF模型。它能够生成高质量的视图，并且不需要逐场景优化。GeoNeRF包含两个主要部分：几何推理器和渲染器。

1. 几何推理器 (Geometry Reasoner)

GeoNeRF的几何推理器通过构建级联代价体积（Cost Volume）来提取场景的几何信息，进而为渲染提供条件。给定一组相邻视图：

$$\{I_v\}_{v=1}^{V} \tag{48}$$

其中：

- I_v：第 v 个源视图的输入图像。
- V：源视图的总数。

1.1 特征金字塔生成

首先，使用特征金字塔网络 (Feature Pyramid Network, FPN) 从每个源视图提取不同尺度的特征图。对于每个视图 I_v，FPN生成三个尺度的特征图：

$$f_v^{(l)} = \mathrm{FPN}(I_v) \in \mathbb{R}^{\frac{H}{2^l} \times \frac{W}{2^l} \times 2^l C}, \quad \forall l \in \{0, 1, 2\} \tag{49}$$

其中：

- H, W：图像的高度和宽度。
- C：通道数。
- l：尺度级别（0、1、2），分别代表不同分辨率的特征图。

1.2 构建级联代价体积

接着，参考CasMVSNet的方法，通过单应性变换 (Homography Warping) 对源视图进行平面扫描，生成多层代价体积 $P_v^{(l)}$。构建代价体积时，使用不同的尺度来逐步细化深度估计：

$$D_v^{(l)}, \Phi_v^{(l)} = R_{3D}^{(l)}(P_v^{(l)}), \quad \forall l \in \{0, 1, 2\} \tag{50}$$

其中：

- $D_v^{(l)}$：在第 l 层生成的深度图。
- $\Phi_v^{(l)}$：在第 l 层生成的三维特征图，用于增强场景的几何信息。
- $R_{3D}^{(l)}$：3D正则化网络，将代价体积进一步处理，输出深度图和三维特征图。

2. 新视图采样 (Sampling Points on a Novel Ray)

为了生成新的视图，沿每条相机光线均匀采样 N_c 个点以覆盖整个深度范围。同时，为提高精度，从代价体积的概率密度分布中再采样 N_f 个点。因此，最终沿每条光线的采样点数为 $N = N_c + N_f$。

- x_n：光线上的第 n 个采样点。

3. 渲染器 (Renderer)

渲染阶段，结合所有视图的特征，通过注意力机制将信息聚合，生成视角独立和视角相关的特征。

3.1 特征插值与表示

在渲染过程中，首先在每个采样点 x_n 处插值所有视图的2D和3D特征。然后，生成全局（视角独立）和视角相关的特征：

$$t_{n,v} = \mathrm{LT}\left([f_{n,v}^{(0)}; \{\Phi_{n,v}^{(l)}\}_{l=0}^{2}]\right), \quad t_{n,0} = \mathrm{LT}\left([\mathrm{mean}\{f_{n,v}^{(0)}\}_{v=1}^{V}; \mathrm{var}\{f_{n,v}^{(0)}\}_{v=}\right.$$

其中：

- $f_{n,v}^{(0)}$：第 v 个视图在 x_n 处的2D特征。
- $\Phi_{n,v}^{(l)}$：第 v 个视图在 x_n 处的3D特征。
- $t_{n,v}$：表示采样点 x_n 的第 v 个视角的特征（视角相关）。
- $t_{n,0}$：表示采样点 x_n 的全局特征（视角独立）。

接下来，使用多头注意力机制 (Multi-Head Attention, MHA) 进一步聚合特征。

3.2 体密度估计

全局特征经过自编码器网络 (Auto-Encoder, AE) 的正则化，生成采样点的体密度 σ_n：

$$\sigma_n = \mathrm{MLP}_{\sigma}(\mathrm{AE}(\{t'_{n,0}\}_{n=1}^{N})) \tag{52}$$

其中：

- σ_n：采样点 x_n 的体密度，表示该点的不透明度。
- AE：自编码器网络，用于沿光线方向聚合几何信息。
- MLP_{σ}：用于体密度估计的多层感知器 (MLP) 网络。

3.3 颜色估计

结合不同视角的信息，为每个采样点计算颜色。首先计算加权颜色：

$$w_{n,v} = \mathrm{Softmax}(\{\mathrm{MLP}_c([t'_{n,v}; \gamma(\theta_{n,v})]), M_{n,v}\}_{v=1}^{V}) \tag{53}$$

$$c_n = \sum_{v=1}^{V} w_{n,v} c_{n,v} \tag{54}$$

其中：

- c_n：采样点 x_n 的颜色。
- $w_{n,v}$：采样点 x_n 在第 v 个视图下的颜色权重。
- $\mathrm{Softmax}(\cdot)$：Softmax函数，用于标准化各视角的颜色权重。
- $c_{n,v}$：第 v 个视图中采样点 x_n 的颜色。
- $M_{n,v}$：遮挡掩码，指示 x_n 是否被第 v 个视图遮挡。
- $\gamma(\theta_{n,v})$：位置编码函数，用于编码新视图与源视图的角度关系。

3.4 体渲染

使用体渲染公式合成光线的最终颜色：

$$\hat{c} = \sum_{n=1}^{N} \exp\left(-\sum_{k=1}^{n-1} \sigma_k\right)(1 - \exp(-\sigma_n))c_n \tag{55}$$

其中：

- $\hat{c}$：最终渲染的光线颜色。
- $\exp(\cdot)$：指数函数，用于计算累积透明度。

4. 损失函数 (Loss Functions)

GeoNeRF在训练过程中使用颜色损失、深度损失和自监督深度估计损失。

4.1 颜色损失

颜色损失衡量渲染颜色与真实颜色之间的误差：

$$L_c = \frac{1}{|R|} \sum_{r \in R} \|\hat{c}(r) - c_{\mathrm{gt}}(r)\|^2 \tag{56}$$

其中：

- L_c：颜色损失。
- R：训练集中光线的集合。
- $\hat{c}(r)$：渲染得到的光线颜色。
- $c_{\mathrm{gt}}(r)$：真实光线颜色。

4.2 深度损失

对于带有真实深度的样本，深度损失用于监督深度估计：

$$L_d = \frac{1}{|R_d|} \sum_{r \in R_d} \| \hat{d}(r) - d_{\text{gt}}(r) \|_{s1} \tag{57}$$

其中：

- L_d：深度损失。
- R_d：具有真实深度的光线集合。
- $\hat{d}(r)$：渲染得到的深度。
- $d_{\text{gt}}(r)$：真实深度。
- $\| \cdot \|_{s1}$：光滑 $L1$ 损失函数。

5. 总结

GeoNeRF结合几何推理和注意力机制，实现了高效的视图合成。通过级联代价体积和自监督的深度估计，GeoNeRF能够生成更准确的场景