

Efficient Implicit Neural Reconstruction Using LiDAR

摘要：文章使用隐式神经表示构建场景几何形状模型。以前的方法在使用彩色或深度图像时取得了令人印象深刻的结果，但它们在处理光照条件较差和大规模场景时仍然面临挑战。以全局点云作为输入的方法需要精确的配准和地面真实坐标标签，这限制了它们的适用范围。本文提出可以在几分钟内使用稀疏 LiDAR 点云和粗略的测程数据有效地重建精细的隐式占据场。我们引入了一个新的损失函数，直接在 3D 空间中监督重建过程，消除了需要进行 2D 渲染并防止信息丢失。此外，我们以端到端的方式优化输入帧的姿态，确保了一致的几何形状，而无需全局点云配准。该方法是第一个仅使用 LiDAR 输入重建隐式场景表示的方法。我们在合成和真实数据集上进行了实验证明了我们方法的有效性、高效性和准确性，获得了与使用密集输入数据的现有方法相媲美的结果。

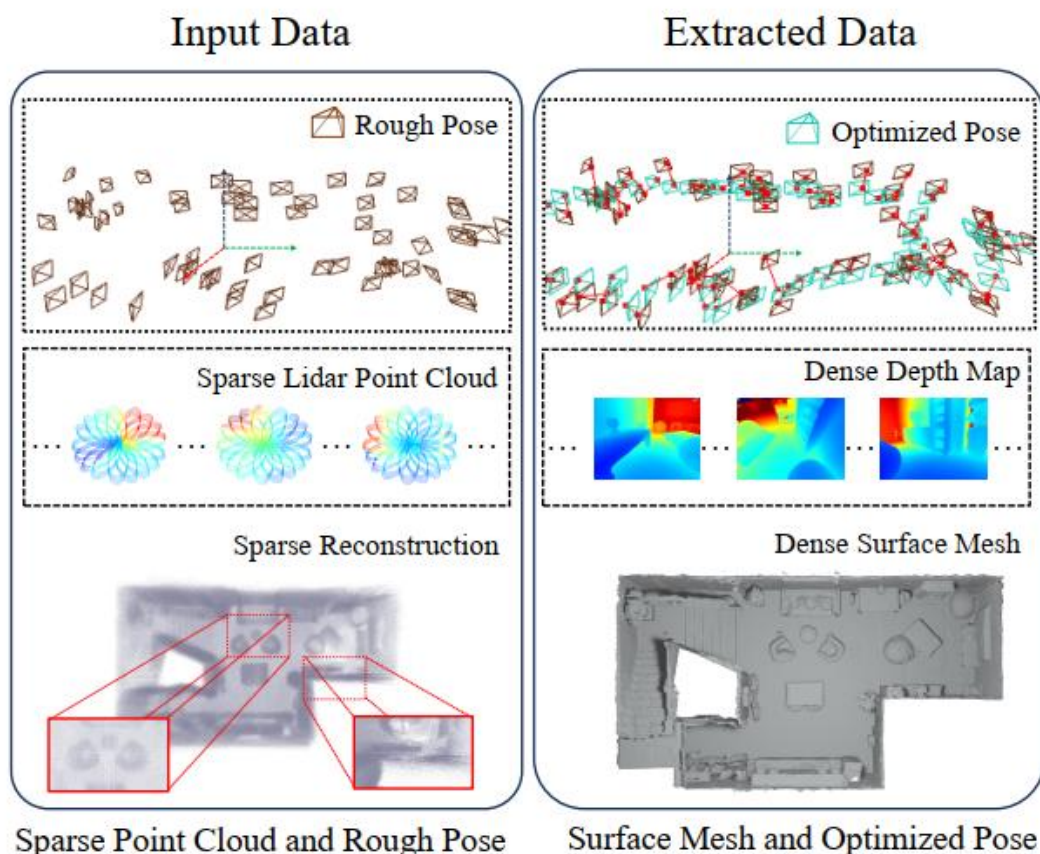


Fig. 1. Our method takes sparse LiDAR point clouds as input and outputs a dense occupancy field using implicit representation. We propose a new direct supervision method to handle the sparsity of the input data and refine the initial rough poses in a joint optimization module.

Contribution

该方法以稀疏 LiDAR 点云和粗略测程数据作为输入，并优化了隐式占据场以进行重建。该方法在效率上表现出色，并比基于渲染的方法更快地实现了良好的重建结果。此外，由于 LiDAR 测量的适应性，该方法适用于各种场景，包括大规模室外场景。我们的主要贡献如下：

- 提出了一种新的损失函数，可以从稀疏的 LiDAR 输入实现密集的隐式重建；
- 在场景优化中添加了姿态优化，以进一步减小初始姿态误差并创建一致的几何形状；
- 我们实施的重建方法高效，并且每个场景只需要几分钟的时间。

System overview

A. 系统概述

提出了一种使用稀疏 LiDAR 点云的新的隐式神经重建方法,以点云帧作为输入,优化占据场。图 2 中展示了方法的流程。给定输入点云序列和来自测程的初始姿态,我们首先根据视点变化选择关键帧,以删除冗余帧并过滤掉低质量的深度测量。然后,我们从关键帧中对点进行采样,并用它们来训练我们的隐式占据场。我们使用多层感知器 (MLP) 和多分辨率哈希编码器作为我们的隐式模型 (第三节 B)。此外,我们提出了一种直接监督方法,以充分利用稀疏点云输入的深度信息 (第三节 C)。最后,我们联合优化我们的隐式网络和传感器姿态 (第三节 D)。最终,我们使用 Marching Cubes 从优化的占据场中提取出密集的表面网格。。

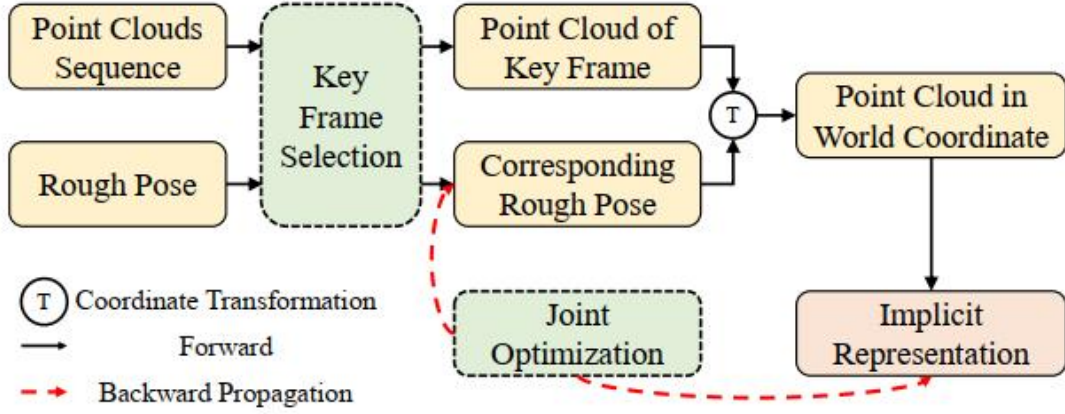


Fig. 2. Pipeline of our method.

B. 使用哈希编码器的隐式表示

为了拟合一个隐式函数,将空间坐标映射到其占用场。由于我们只有稀疏的点云作为输入,隐式函数的监督相对较稀疏。在这种情况下,需要一种低频感知结构来插值没有监督的区域。另一方面,高频信息也需要关注以进行精细的重建。因此,我们遵循 InstantNGP 的方法,利用多分辨率哈希表来编码我们的坐标,以平衡场景的低频和低频部分。

哈希表是建立在场景的多分辨率体素网格上的。我们使用空间哈希函数将可学习特征与体素坐标关联起来。对于三维空间中的空间点,可以通过三线性插值来获得其特征嵌入。然后,我们将来自所有分辨率的特征向量串联起来,形成频率感知的多分辨率潜在向量。哈希表的使用可以大大降低内存成本。然而,高分辨率层中可能会发生哈希冲突,这可能会导致无监督区域出现伪影。为了避免这种情况,我们从输入坐标中提取一个额外的特征向量,并将它们连接成我们的最终特征。

由于哈希表已经存储了大部分场景特征,我们使用具有低容量的浅层 MLP 来解码特征 (x, y, z) 并使用 Sigmoid 函数得到占据决策边界。我们的完整网络结构如图 4 所示。

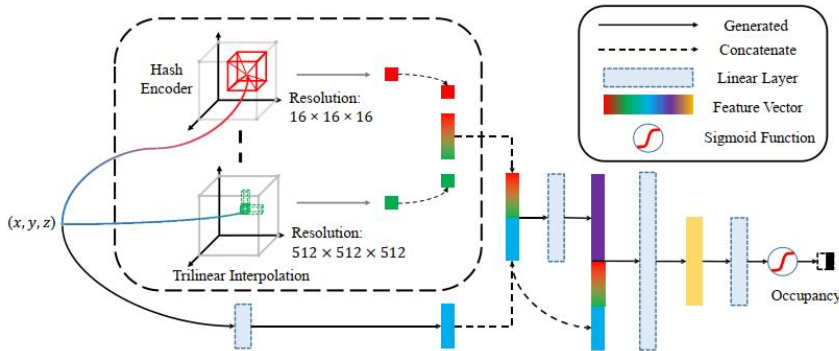


Fig. 4. Implicit network architecture of our method.

C. Direct Supervision

为了充分利用每个关键帧的视图信息，将点云中的点视为光线。具体来说，给定一个包含 n 个点的点云帧，其中每个点都是一个坐标为 (x_i, y_i, z_i) 的向量，第 i 条光线上的点可以表示 $\tilde{\mathbf{p}}_i(z) = \mathbf{o}_i + z\mathbf{d}_i$ ，其中 \mathbf{o}_i 是传感器的起点， \mathbf{d}_i 是光线的方向。

根据 LiDAR 测量的机制，激光在击中 \mathbf{p}_i 之前可以沿着光线传播，这意味着 \mathbf{p}_i 之前沿着光线的占用值应该为 0，而 \mathbf{p}_i 之后的值应该为 1。这样，可以直接将简单的二元交叉熵 (binary cross-entropy, BCE) 损失应用于占用函数。

$$\mathcal{L}_d(\tilde{\mathbf{p}}_i(z)) = \begin{cases} -\log(1 - f_\theta(\tilde{\mathbf{p}}_i(z))), & \text{for } z < z_i \\ -\log(f_\theta(\tilde{\mathbf{p}}_i(z))), & \text{for } z \geq z_i \end{cases} \quad (1)$$

然而，由于遮挡，我们无法获得物体的厚度，因此无法确定要监督的占用区域的范围。简单地将被遮挡的空间视为已占用会导致当其他视图观察到同一区域为非占用时出现歧义。

为了解决遮挡问题并对物体后面的点进行监督，我们提出了一个简单的物体厚度假设。我们提出了一个称为广义厚度 (generalized thickness) 的定义。广义厚度 b 指的是射线穿过的单个连续占用区域的距离：

$$b = \min(z_s - z_i \mid \mathcal{O}(\mathbf{o} + z_s\mathbf{d}) = 0.5), \quad (2)$$

其中， $\mathcal{O} = 0.5$ 表示 the level-set of the surface， z_s 表示射线相交的下一个表面的深度。基于这个定义，我们将环境中物体的广义厚度视为随机变量 B 。我们假设它服从对数正态分布，依赖于场景几何形状的先验信息，可以推导出射线上采样的点在 z 位于 (z_i, z_s) 范围内被占用的概率：

$$P_{occ}(z) = P(B > z - z_a) = 1 - F_B(z - z_a), \quad (3)$$

这里， F_B 表示变量 B 的累积分布函数。一旦获得了被遮挡采样点的占用概率，就可以对物体后面的点应用监督。同时，非占用点会被赋予概率 1。通过用观测概率加权简单的 BCE 损失函数，可以将对点的遮挡感知直接损失修改为如下形式：

$$\mathcal{L}_d(\tilde{\mathbf{p}}_i(z)) = \begin{cases} -\log(1 - f_\theta(\tilde{\mathbf{p}}_i(z))), & \text{for } z < z_i \\ -P_{occ}(z) \log(f_\theta(\tilde{\mathbf{p}}_i(z))), & \text{for } z \geq z_i \end{cases} \quad (4)$$

D. Joint Optimization

由于我们已经在每条射线上有了表面位置，为了更关注近表面区域，我们使用正态分布在每条射线上采样 m 个点，并假定与场景大小相关的超参数。我们另外以分层方式采样其他 k 个点。在优化过程中，首先从 N_f 个点云中采样一个批次的 N_b 个点，并形成 $N_f \times N_b$ 条射线。然后，在每条射线上采样 $m + k$ 个点，并使用这些总共的 $N_f \times N_b \times (m + k)$ 个点进行监督。

除了 III-C 节中提到的损失函数 $\mathcal{L}_d(\mathbf{p}_i(z))$ ，我们还对表面点添加了另一个表面正则化损失

$$\mathcal{L}_n(\mathbf{p}_i) = |1 - \mathbf{n}(\mathbf{p}_i) \cdot \mathbf{n}(\mathbf{p}_i + \epsilon)|, \quad (5)$$

最终损失函数

$$\mathcal{L} = \frac{1}{N_f N_b} \sum_{i=0}^{N_f N_b} \left(\frac{1}{m+k} \sum_{j=0}^{m+k} \lambda_d \mathcal{L}_d(\tilde{\mathbf{p}}_i(z_j)) + \lambda_n \mathcal{L}_n(\mathbf{p}_i) \right) \quad (7)$$

