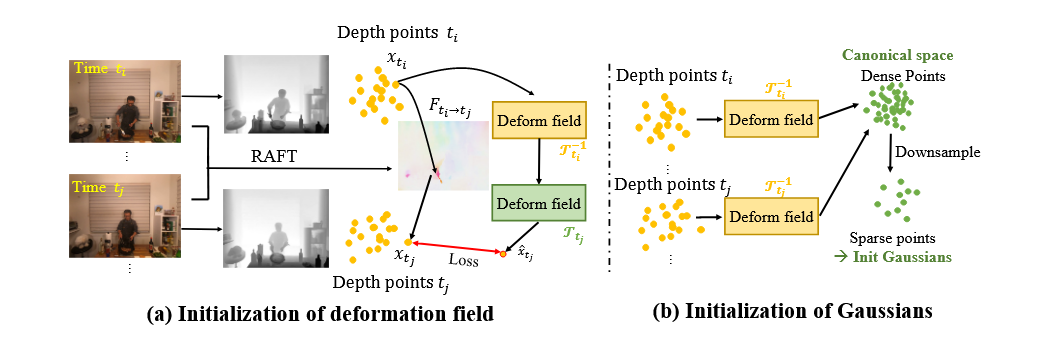
对于输入的单目视频，用深度估计器对每一帧的深度进行计算，用光流估计方法RAFT对任意两帧之间的光流进行计算。然后经过如下的方式进行变形场和高斯场景的初始化。



原始的3dgs依赖sfm的点云结果和相机位姿，而这个工作是单目视觉，所以比较正常的想法是通过深度估计来对第一帧进行深度估计，然后在对应深度铺上高斯单元进行初始化，但作者证明这是次优的结果。

而作者的初始化是基于3D感知的初始化。

1. **深度尺寸的初始化：**通过SAM2或者2D光流留来判断出静态区域，因为不同的帧计算出来的深度图可能是不同数值为基础的，作者通过将不同帧的静态区域的深度估计值投影到第一帧，然后计算静态区域的差值，通过减小静态区域深度估计的差值来重新更正所有深度图。（这个比较好理解）
2. **变形网络的初始化：**因为我们有了深度图，可以把每个像素看作是一个空间中的3D点，根据深度图先弄出这个，然后根据光流，可以得到一个3D流，那么实际上我们就知道了3D点随着时间的变化，这个可以简单地训练变形网络，进行初始化。用下面的损失函数进行训练。

QQ20241112-213839

这里解释一下这个-1是说可以从第i帧计算出最原始的场景，这个损失函数的意思就是把第i帧的场景放到变形网络里算出最开始的，然后再算出第j帧的，和第j帧通过一开始那个3D点算出来的进行对比，进行训练

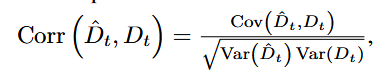
1. **高斯场景的初始化：**根据2中的步骤，可以将每帧的3D点都弄出来，然后利用变形网络把他们都映射回正则空间（即原始空间），然后组合在一起，但是因为点太多了，作者根据原先预先设定好的体素大小对点云进行下采样，从而得到了类似sfm得到的点云文件。

损失函数：

作者管这个叫Ordinal depth loss，阶梯状深度损失：

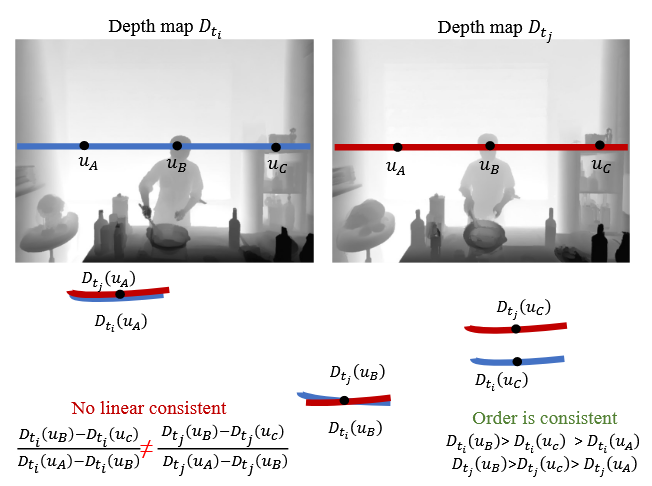
下面一步步介绍这个阶梯状深度损失的由来

1. Pearson correlation loss皮尔森相关损失:对渲染出来的图像和真实图像的深度估计损失进行计算。（因为单目深度估计在尺寸范围上是不一样的，用皮尔森相关性直接避免这种不一致的尺寸，免得再来一遍**深度尺寸初始化**中的步骤）



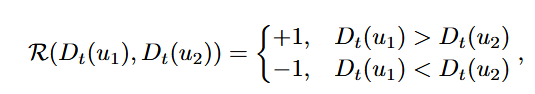
1. Pearson correlation loss的局限：然而，由于深度估计器不一定是完全精准的，在两个估计出来的深度图中，我们虽然常常只是说他们尺度范围不同，但实际上把他们归一化到一个尺寸上后，像素间的深度值也不能满足某种同样的线性关系（比如说利用静态区域的线性关系将两张图归一化到同一个尺寸后，会发现别的没有前后移动只进行了左右移动的区域的深度仍然有可能不一样）。同样的，皮尔森相关性也同样有这样的问题，即使不归一化，像素间也不会满足某种线性关系。

实际上深度估计器虽然不能准确的估计出两张具有某种线性关系的深度图，但是他 估计出的前后顺序一般是比较准确的。

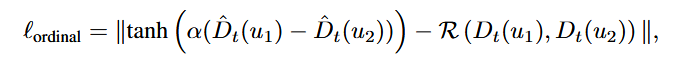


1. 最终提出的损失函数：

作者先定义一个顺序图对深度图每个像素的顺序进行计算



然后用下面这个进行计算，通过随机采样10万个像素对u1和u2来进行计算



最终的损失函数还要加上mse损失

总结：本文创新点在于提出一种靠光流和深度估计的高斯和变形网络初始化，以及一个比较容易理解的损失函数。还有一个要点是这里使用的MLP是可逆的MLP

作者论文中与SCGS、Deformable GS、HexPlane、RoDynRF进行了对比

个人感觉他这个利用像素的深度顺序作为损失函数，可能在精度上不一定高。