Photo-SLAM:实时同步定位和真实感地图构建
适用于单目、立体和 RGB-D 相机

Huajian Huang1 Longwei Li2 Hui Cheng2 Sai-Kit Yeung1
1香港科技大学2中山大学
hhuangbg@connect.ust.hk, lilw23@mail2.sysu.edu.cn, Chengh9@mail.sysu.edu.cn, saikit@ust.hk

抽象的

神经渲染和 SLAM 系统的集成最近在联合定位方面显示出了良好的效果
和照片级真实感视图重建。然而,现有的
完全依赖于隐式表示的方法是如此
占用大量资源,无法在便携设备上运行,
这背离了 SLAM 的初衷。在这个
论文中,我们提出了 Photo-SLAM,一种新颖的 SLAM 框架
用超原始图。具体来说,我们同时利用显式几何特征进行定位和

学习隐式光度特征来表示所观察环境的纹理信息。此外

基于几何的主动加密超原始
特征,我们进一步引入基于高斯金字塔的
训练方法逐步学习多层次特征,
增强照片级真实感映射性能。对单目、立体和 RGB-D 进行了广泛的实验

数据集证明我们提出的系统 Photo-SLAM 显著优于当前最先进的在线
真实感地图 SLAM 系统,例如,PSNR 为 30%

渲染速度提高了数百倍
Replica 数据集。此外,Photo-SLAM 可以在
使用 Jet-son AGX Orin 等嵌入式平台实时速度,展示机器人应用的潜
力。项目页面和代码： https://huajianup.

github.io/research/Photo-SLAM/。

1. 简介

使用同步定位和地图构建 (SLAM)
相机是计算机视觉和机器人技术中的一个基本问题,旨在实现自主系统

导航和理解周围环境。传统的 SLAM 系统[7–9, 24]主要侧重于几何映
射,提供准确但视觉上简单的环境表示。然而,神经渲染领域[35, 40]的最
新发展已经证明了

集成真实感视图重建的潜力

图 1. 渲染和轨迹结果。Photo-SLAM 可以使用单目、立体、

和 RGB-D 相机,渲染速度高达 1000 FPS。

进入 SLAM 管道,增强机器人系统的感知能力。

尽管通过神经渲染和 SLAM 的集成取得了令人鼓舞的成果,但现有
方法仅仅严重依赖于隐式表示,

这使得它们计算量大,不适合
在资源受限的设备上部署。例如,Nice-SLAM [46]利用分层网格[42]来

存储代表环境的可学习特征,同时
ESLAM [16]利用多尺度紧凑张量组件[3]。然后它们联合估计相机姿态
和
通过最小化重构损失来优化特征
一批射线采样[21]。这样的优化过程非常耗时。因此,

让它们结合相应的深度信息
从 RGB-D 相机等各种来源获得,
密集光流估计器[33],或单目深度估计器

timators [12]来确保高效收敛。此外，
由于隐式特征是通过多层解码的
感知器（MLP），通常需要仔细
定义一个边界区域来规范射线采样以获得最佳性能，如[14]中所述。它本质上限制了
系统的可扩展性。这些限制意味着
它们无法使用便携式平台在未知环境中提供实时探索和映射功能，而这是
SLAM 的主要目标之一。

在本文中，我们提出了一种创新的 Photo-SLAM
解决可扩展性和计算问题的框架
现有方法的资源限制，同时实现
精确定位和在线真实感地图绘制。我们
维护一个由以下部分组成的超原始映射
点云存储 ORB 特征[26]、旋转、缩放、
密度和球谐函数（SH）系数[10,38]。
超原始映射允许系统有效地
使用因子图求解器优化跟踪并学习
通过在原始图像和渲染图像之间反向传播损失来实现相应的映射。图像是通过 3D 高斯铺展[18]而不是

射线采样。虽然引入了 3D 高斯
splatting 渲染器可以减少视图重建成本，它
无法生成高保真渲染
在线增量建图，特别是在单目场景中。为了实现高质量的建图，而无需依赖

在密集深度信息上，我们进一步提出了一种基于几何的致密化策略和一种基于高斯金字塔的
（GP）学习方法。重要的是，GP 学习有助于
多层次特征的逐步获取有效地提高了系统的映射性能。

为了评估我们提出的方法的有效性，我们
使用不同的数据集进行广泛的实验
由单目、立体和 RGB-D 相机拍摄。这些
实验结果明确表明，Photo-SLAM 在定位效率、真实感地图质量和

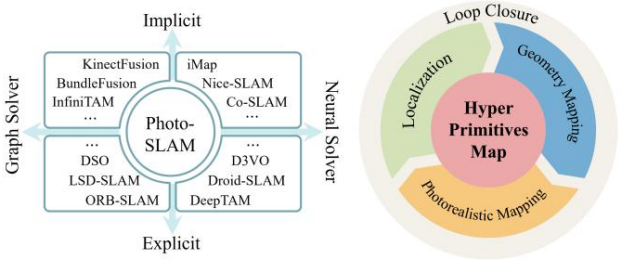
渲染速度。此外，实时执行
嵌入式设备上的 Photo-SLAM 系统展示了其在实际机器人应用方面的潜力。

Photo-SLAM 的示意图如下
图1和图2b。

总而言之，这项工作的主要贡献包括：

- 我们开发了第一个基于超基元的同步定位和照片级真实感地图绘制系统

  地图。新框架支持单目、立体、
  以及室内和室外环境中的 RGB-D 相机。
- 我们提出了基于高斯金字塔的学习方法，允许
  模型能够高效、有效地学习多层
  实现高保真映射的功能。
- 该系统完全采用 C++ 和 CUDA 实现，
  达到了最先进的性能，即使在嵌入式平台上也可以实时运行。



(a)分类　　(b)Photo-SLAM 概述

图 2. Photo-SLAM 包含四个主要组件，包括定位、显式几何映射、隐式照片真实感映射和回环组件，同时保持

带有超原始元素的地图。

2.相关工作

视觉定位与地图绘制是一个旨在
建立未知环境的适当表征
通过摄像头来估计它们在该环境中的姿势。与 SfM 技术相比，视觉 SLAM 技术通常在准确性和可预测性之间寻求更好的平衡

和实时性能。在本节中，我们重点介绍视觉SLAM并进行简要回顾。

图解器与神经解算器。经典 SLAM 方法广泛采用因子图来模拟变量（即姿势和地标）之间的复杂优化问题

和测量（即观察和约束）。
为了实现实时性能，SLAM 方法逐步传播其姿态估计，同时避免昂贵的操作。例如，ORB-SLAM 系列方法[2, 23, 24]依赖于提取和跟踪轻量级

连续帧中的几何特征，在本地而不是全局执行捆绑调整。此外，像 LSD-SLAM [7]和 DSO [8]这样的直接 SLAM操作原始图像强度，而无需提取几何特征。它们在线维护由点云表示的稀疏或半密集地图，即使在

资源约束系统。从成功中受益
深度学习模型、可学习参数和模型
被引入到 SLAM 中，使得流程可区分。一些方法，如 DeepTAM [45]通过神经网络[17]端到端预测相机姿态，而

精度有限。为了提高性能，一些方法，例如 D3VO [41]和 Droid-SLAM [34]，引入

单目深度估计[12]或密集光流估计[33]模型作为监督进入 SLAM 流程

信号。因此，它们可以生成明确表示场景几何形状的深度图。通过大规模

合成 SLAM 数据集 TartanAir [37]可用于训练，基于 RAFT [33]构建的 Droid-SLAM实现了最先进的性能。然而，纯基于神经网络的

求解器的计算成本很高，并且其性能
未见过的场景的效果会显著下降。

显式表示与隐式表示。在
为了获得密集的重建,包括 KinectFusion [15]、 BundleFusion [6] 和
Infini-TAM [25]在内的一些方法利用隐式表示 Truncated

符号距离函数（TSDF） [5],用于整合传入的 RGB-D 图像并重建连续表面,
该函数可以在 GPU 上实时运行。尽管它们

可以获得密集重建,视图渲染质量
有限。最近,神经渲染技术代表了
神经辐射场（NeRF） [21]实现了令人惊叹的新视图合成。给定相机姿势,
NeRF
通过多层感知器 (MLP) 隐式建模场景几何和颜色。通过最小化渲染图像和训
练视图的损失来优化 MLP。

然后， iMAP [30]调整 NeRF 进行增量建图,不仅优化了 MLP,还优化了相机
姿势。接下来的工作 Nice-SLAM [46]引入了多分辨率

网格[42]来存储特征,降低深度MLP的成本
查询。Co-SLAM [36]和 ESLAM [16]分别探索 Instant-NGP [22]和 TensoRF
[3],以进一步加快建图速度。然而,相机姿态和几何表示的隐式联合优化仍然不
理想。不可避免地,它们依赖于来自 RGB-D 相机的显式深度信息或额外的模型
预测

用于辐射场的快速收敛。
我们提出的 Photo-SLAM 旨在恢复一个简洁的
沉浸式体验中观察到的环境的表示
探索,而不是重建一个密集的网格。它
在线维护一个带有超基元的地图,该地图利用显式几何特征点进行准确、高效
的定位,同时利用隐式表示

捕捉和建模纹理信息。请参阅
图2a为现有系统的分类。由于 Photo-SLAM 无需依赖

密集的深度信息,它可以支持RGB-D相机
以及单目相机和立体相机。

3. 照片 SLAM

Photo-SLAM 包含四个主要组件,包括定位、几何映射、真实感映射和

循环闭合,如图2b 所示。每个组件都运行在
并行线程并共同维护超原始映射。

3.1. 超基元映射

在我们的系统中,超基元被定义为一组点
云 $P \in R^3$ 与 ORB 特征相关[26] $O \in R^{256}$,旋转 $r \in SO(3)$,缩放 $s \in R^3$,密度 $\sigma \in R^1$,
和球谐系数 $SH \in R^{16}$。从图像帧中提取的 ORB 特征负责建立 2D 到 2D 和
2D 到 3D 的对应关系。一旦

系统成功估计了变换矩阵
基于相邻帧之间足够的二维到二维对应关系,超基元映射通过三角函数初始化

角度,姿势跟踪开始。在跟踪过程中，
定位组件处理传入的图像
并利用二维到三维的对应关系来计算
当前相机姿态。此外,几何映射
组件将逐步创建并初始化稀疏
超基元。最后,真实感组件逐步优化和密集化超基元。

3.2. 定位和几何映射

定位和几何映射组件不仅提供有效的 6-DoF 相机姿态估计

输入图像,还有稀疏的三维点。优化问题被公式化为一个因子图,通过

Levenberg-Marquardt (LM) 算法。
在定位线程中,我们使用仅运动的捆绑包
调整以优化相机方向 $R \in SO(3)$
且位置 $t \in R^3$ 为了最小化匹配的二维几何关键点 pi 之间的
重投影误差
框架和 3D 点Pi 。 令 $i \in X$ 为集合的索引
匹配 X ，我们尝试用 LM 来优化的是

(1)

其中Σg是尺度相关协方差矩阵
关键点,π(·) 是 3D 到 2D 的投影函数,ρ
表示稳健的Huber成本函数。

在几何映射线程中,我们执行本地
对一组共视点PL进行捆绑调整,并且
关键帧KL。关键帧是从
输入摄像机序列并提供良好的视觉信息。我们构建一个因子图,其中每个关键
帧
是一个节点,边表示
关键帧和匹配的 3D 点。我们迭代最小化
通过改进关键帧姿势来重新投影残差,并
使用误差函数的一阶导数来计算 3D 点。我们固定关键帧KF的姿势,这些关键
帧也在观察PL,但不在KL 中。设 K = KL ∪ KF ， Xk为

关键帧 k 中的 2D 关键点之间的匹配集
和 PL 中的 3D 点。优化过程旨在减少 K 和PL 之间的几何不一致性,并且

定义为

(2)

带有重投影残差

3.3. 真实感贴图

真实感贴图线程负责优化由以下方式逐步创建的超原始数据：

几何映射线程。超基元可以是

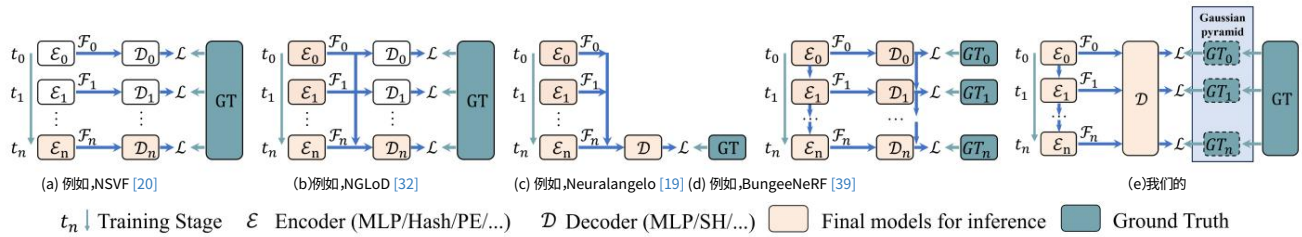


图 3. 不同渐进式训练方法的比较。这里的编码器En表示回归特征Fn 的结构,它
可以是 MLP、体素网格、哈希表、位置编码等。这里的解码器Dn表示将Fn转换为密度的结构,
颜色或其他信息。我们提出了一种基于高斯金字塔的新方法来有效地学习多级特征。

通过基于图块的渲染器进行栅格化,以合成具有关键帧姿势的对应图像。渲染
过程是
表述为

(3)

其中 N 是超基元的数量， ci表示
从 SH ∈ R 16转换而来的颜色， αi等于
σi ·G(R,t, Pi , 里 , si),G 表示 3D 高斯分布
算法[18]。位置P、旋转r、缩放s、密度σ和球谐系数SH的优化是通过最小化光度
损失来实现的

渲染图像Ir与地面真实图像Igt 之间的 L，
表示为

(4)

其中 SSIM(Ir , Igt)表示
两幅图像,λ 是平衡的权重因子。

3.3.1 基于几何的致密化

如果我们将照片级真实感地图视为回归模型
场景中,更密集的超基元,即更多的参数,通常可以更好地模拟场景的复杂性

以提高渲染质量。几何映射组件为了满足实时映射的需求,只建立了稀疏的超
基元,因此几何映射所创建的粗超基元需要

在真实感地图优化过程中变得密集。
除了分裂或克隆大型超原始
类似于[18]的损失梯度,我们引入了一个额外的
基于几何的致密化策略。
实验表明,不到 30% 的二维几何特征
帧的点是活跃的,并有相应的3D
点,特别是对于非 RGB-D 场景,如图所示
图4. 我们认为二维几何特征点在空间上
分布在框架中的基本上代表了该区域
具有需要更多超级原始元素的复杂纹理。
因此,我们积极创造额外的临时超
基于非活动二维特征点的图元
关键帧是为真实感地图绘制而创建的。当我们

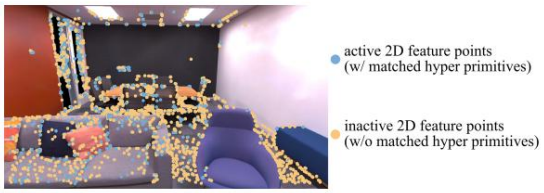


图 4. 我们利用初始几何信息来加密
超原始。

使用 RGB-D 相机,我们可以直接投影非活动 2D
特征点的深度来创建临时的超原始图像。对于单目场景,我们估计

通过解释其深度来识别非活动二维特征点
最近邻域的活跃二维特征点。在立体
场景中,我们依靠立体匹配算法来估计非活动二维特征点的深度。

3.4. 基于高斯金字塔的学习

渐进式训练是神经渲染中广泛使用的一种技术,可以加速优化过程。一些

已经提出了一些方法来减少训练时间,同时
实现更好的渲染质量。基本方法是
逐步增加结构分辨率和模型参数数量。例如,NSVF [20]和

DVGO [31]在训练过程中逐步增加特征网格分辨率,与以前的工作相比,显著
提高了训练效率。较低分辨率的模型用于初始化较高分辨率的

模型,但不会保留用于最终推理,如图所示
图3a.通过多分辨率增强性能
NGLoD [32]逐步训练多个 MLP
作为编码器和解码器,同时只保留最后的
解码器解码集成的多分辨率特征,如
如图3b 所示。此外,Neuralangelo [19]仅
在训练期间维持单个 MLP,如图3c 所示。
它逐步激活不同级别的哈希表[22]
在大规模场景重建中取得了更好的性能。同样,3D Gaussian Splatting [18]
逐步加密 3D Gaussian,取得了最佳性能

辐射场渲染。训练不同层次的模型

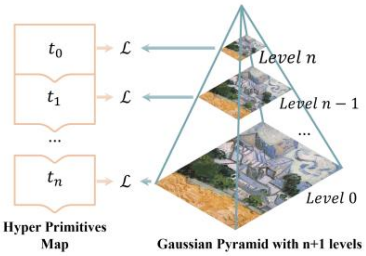


图5.基于高斯金字塔的训练过程。

这些方法中的 els 受到相同训练的监督
图像。相反,第四种方法(图3d)用于
BungeeNeRF [39]是应用不同的模型来解决
不同分辨率的图像。BungeeNeRF 演示了
明确分组多分辨率训练的效率
图像来让模型学习多层次的特征。然而,
这种方法并不通用,因为多分辨率图像
在大多数情况下不可用。

为了充分利用各种优点,我们提出了基于高斯金字塔(GP)的学习(图3e),一种新的渐进式
训练方法。如图5 所示,高斯金字塔是图像的多尺度表示,包含

不同层次的细节。它通过反复应用高斯平滑和下采样操作构建

到原始图像。在开始训练步骤中,
超级原始人受到最高级别的
金字塔,即第 n 层。随着训练迭代的增加,我们不会
仅对超原始元素进行致密化,如第3.3.1节所述,但
还可以降低金字塔等级并获得新的地面实况
直到到达高斯金字塔的底部。
使用 n+1 高斯金字塔的优化过程
水平可以表示为

(5)

其中 L(Ir , GP(Igt))为公式4,而 GPn (Igt)表示
高斯金字塔第 n 层中的地面图像。在
实验中,我们证明了 GP 学习显著提高了照片级真实感地图绘制的性能,特别是对于单目相机而言。

3.5. 环路闭合

回环闭合[11]在 SLAM 中至关重要,因为它有助于解决可能

发生在定位和几何映射过程中。检测到闭合环后,我们可以纠正局部

通过相似变换,将关键帧和超基元转换为真实感图像。通过校正相机姿势,真实感地图绘制组件可以进一步消除由

通过里程计漂移来提高地图绘制质量。

4.实验

在本节中,我们将 Photo-SLAM 与其他最先进的 (SOTA) SLAM 和实时 3D 重建系统在各种场景中进行比较,包括单目、立体、

RGB-D 相机以及室内和室外环境。在
此外,我们还评估了各种硬件配置上的 Photo-SLAM 性能以证明其效率。

最后,我们进行消融研究以验证所提出算法的有效性。

4.1. 实施和实验设置

我们完全用 C++ 和 CUDA 实现了 Photo-SLAM,
利用 ORB-SLAM3 [2]、3D Gaussian splating [18]和 LibTorch 框架。优化

我们使用固定学习率和 λ = 0.2,通过随机梯度下降算法实现照片级真实感贴图。考虑到

测试数据集,高斯金字塔的级别设置为
三个,即默认 n = 2。比较的基线包括 SOTA 经典 SLAM 系统 ORB-SLAM3 [2],
实时 RGB-D 密集重建系统 BundleFu-sion [6]、基于深度学习的系统 DROID-SLAM [34],
以及最近支持视图合成的 SLAM 系统,即
Nice-SLAM [46]、 Orbeez-SLAM [4]、ESLAM [16]、 Co-SLAM [36]、 Point-SLAM [27]和Go-SLAM [44]。
硬件。我们在一台配备 NVIDIA 的台式机上运行了 Photo-SLAM,并使用其官方代码比较了所有方法
RTX 4090 24 GB GPU、英特尔酷睿 i9-13900K CPU,
和 64 GB RAM。我们进一步在笔记本电脑和 Jetson AGX Orin 开发套件上测试了 Photo-SLAM。笔记本电脑
配备 NVIDIA RTX 3080ti 16 GB 笔记本 GPU,
英特尔酷睿 i9-12900HX 和 32 GB RAM。

数据集和指标。我们在著名的 RGB-D 数据集上对单目和 RGB-D 传感器类型进行了测试
数据集:Replica 数据集[28, 30]和 TUM RGB-D 数据集[29]。至于立体测试,我们使用了 EuRoC
MAV 数据集[1]。除了室内场景,我们还利用 ZED 2
立体摄像机收集户外场景以进行额外评估。

按照惯例,我们使用绝对轨迹误差(ATE)度量[13]来估计定位精度,同时报告ATE的RMSE和STD。

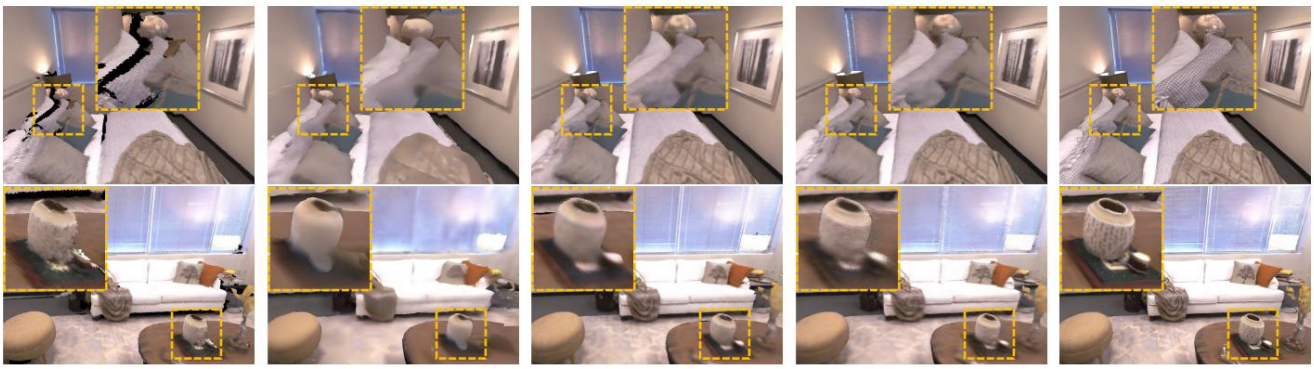
PSNR、SSIM 和
采用LPIPS [43]来分析照片级真实感地图绘制的性能。我们还通过展示跟踪
FPS、渲染

FPS、GPU 内存使用情况。关于
网格重建超出了本文的范围。此外,为了降低非确定性的影响,

多线程和机器学习系统,我们分别运行了
序列五次并报告每次的平均结果
公制。详情请参阅补充材料。

在副本数据集上		定位（厘米）			映射			资源			
凸轮	方法	RMSE ↓	STD ↓	PSNR ↑	SSIM ↑	LPIPS ↓	操作时间 ↓	跟踪 FPS ↑	渲染 FPS ↑	GPU 内存使用情况 ↓	
▪	ORB-SLAM3 [2]	3.942	3.115	DROID-SLAM [34]	-	-	-	<1 分钟 <2	58.749	-	0
	0.725	0.308	Nice-SLAM* [46]	99.9415	35.336	16.311	-	-	-	分钟 >10	35.473	-	11 GB
	Orbeez-SLAM [4]						0.720	0.439	分钟 <5 分钟	2.384	0.944	12 GB	
		-	-		23.246	0.790	71.054	0.336	<5 分钟 <5	49.200	1.030	6 GB	
	去 SLAM [44]	24.593	21.172	0.703	0.756	29.284	0.883	0.524	0.421	分钟 <5 分	25.366	0.821	22 GB
	我们的（杰森一家）	1.235		33.049	0.926	0.892	33.302	0.926	0.139	钟 <2 分钟	18.315	95.057	4GB
	我们的（笔记本电脑）	0.713							0.086		19.974	353.504	4GB
	我们的	1.091							0.078		41.646	911.262	6 GB
BGB-	ORB-SLAM3 [2]	1.833	1.478		-	-	-			<1 分钟 <2	52.209	-	0
	机器人猛击[34]	0.634	0.248		-	-	-			分钟 <5 分	36.452	-	11 GB
	束融合[6]	1.606		0.969	23.839	0.822	1.590	26.158	0.197	钟 >10 分	8.630	-	5 GB
	尼斯大满贯[46]	2.350		0.832	0.562	32.516	0.916	0.274	0.232	钟 <5 分钟	2.331	0.611	12 GB
	Orbeez-SLAM [4]	0.888		30.594	0.866	0.602	30.246	0.864	0.112	<5 分钟 <5	41.333	1.401	6 GB
	伊斯兰教[16]	0.568		0.218	24.158	0.第766章			0.162	分钟 <5 分	6.687	2.626	21 GB
	联合SLAM [36]	1.158							0.175	钟 <5 分钟	14.575	3.745	4GB
	去 SLAM [44]	0.571							0.352	>2 小时 <5	19.437	0.444	24 GB
	点式 SLAM [27]	0.596		0.249	34.632	0.927	0.289	31.978	0.083	分钟 <5	0.345	0.510	24 GB
	我们的（杰森一家）	0.581		0.916	0.289	34.853	0.944	0.298	0.101	分钟 <2 分	17.926	116.395	4GB
	我们的（笔记本电脑）	0.590		34.958	0.942				0.062	钟	20.597	396.082	4GB
	我们的	0.604							0.059		42.485	1084.017	5 GB

表 1. Replica 数据集上的定量结果。我们将最好的两个结果标记为第一和第二。Nice-SLAM* 表示深度
监督被禁用。“-”表示系统不支持视图渲染或无法跟踪相机姿势。Photo-SLAM 的结果
在笔记本电脑和 Jetson 平台上运行的分别表示为“我们的（笔记本电脑）”和“我们的（Jetson）”。



(一)BundleFusion [6] (b)Nice-SLAM [46] (c) 伊斯兰教法[16] (d)协同 SLAM [36] (e)我们的

图 6. 使用来自数据集 Replica 的 RGB-D 图像对不同系统进行定性比较。Photo-SLAM 可以重建高保真
有些场景过于平滑，而有些场景则有明显的瑕疵。放大可获得更好的视图。

4.2. 结果与评价

在 Replica 上。正如定量比较所示
表1，Photo-SLAM 在以下方面取得了最佳表现
地图绘制质量。Photo-SLAM 具有极具竞争力的定位精度，能够实时跟
踪相机姿态。
此外，Photo-SLAM 可以每秒 1200×680 的分辨率渲染数百个逼真
的视图，并且
GPU 内存使用情况。即使在嵌入式平台上，
Photo-SLAM的渲染速度约为100 FPS。

在单目场景中，Photo-SLAM 明显压制了其他方法。当我们禁用
Nice-SLAM [46]的深度监督时，其定位精度

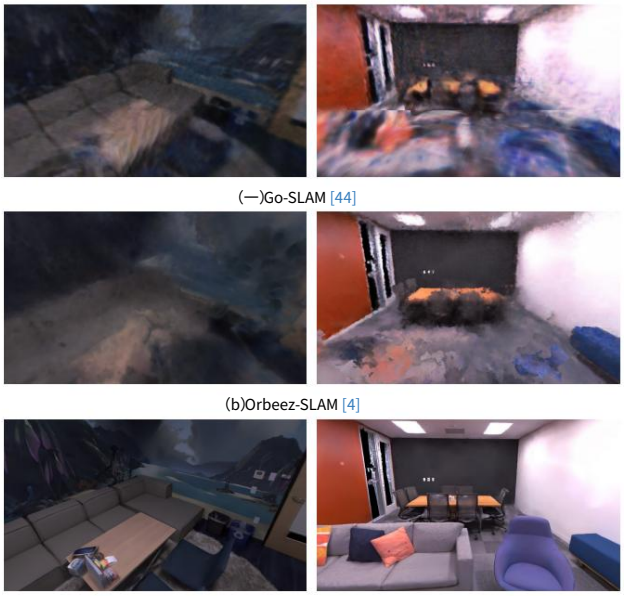
而映射为 16.311
PSNR。我们在图7中进行了定性比较。
Photo-SLAM 的测绘结果具有照片级真实感。

在 RGB-D 场景中，我们运行了 BundleFusion [6]
RGB-D 序列，然后提取纹理网格。并且
然后我们使用网格渲染来渲染相应的图像
对比分析结果，如图6 所示，重建网格
通过经典方法得到的图形可能会产生混叠和空洞。
ESLAM [16]和 Go-SLAM [44]具有最佳定位
精度，但映射缺乏高频细节。
相比之下，Photo-SLAM 可以渲染高保真图像，并且
渲染速度大约快了三百倍。
在慕尼黑工业大学。我们对三个方面进行了定量分析
表 2 中 TUM 数据集的序列。与
与基于学习的方法（例如 DROID-SLAM [34]和 Go-SLAM [44]）相比， ORB-SLAM3
无需 GPU 即可运行得更快，并且在定位方面具有更高的准确性。结果表明，经典方法在
鲁棒性和泛化方面仍然具有优势。图8是

Photo-SLAM 地图绘制图库。
立体摄像机可以提供更稳健的跟踪能力。

在 TUM 数据集上		fr1-办公桌				fr2-xyz				fr3-办公室			
凸轮	方法	RMSE（厘米）↓	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE（厘米）↓	PSNR ↑	SSIM ↑	LPIPS ↓	RMSE（厘米）↓	PSNR ↑	SSIM ↑	LPIPS ↓
▪	ORB-SLAM3 [2]	1.534	-	-	-	0.720	-	-	-	1.400	-	-	-
	机器人猛击[34]	78.245	33.122	1.757	1.549	36.050	-	-	-	154.383	-	-	-
	去 SLAM [44]	1.539	11.705	0.406	0.614	18.811	28.584	14.807	0.443	0.572	105.755	13.572	0.480
	我们的（杰森一家）		0.681	0.329	20.515	0.733	0.558	21.347	0.727	0.187	21.575	1.687	18.884
	我们的（笔记本电脑）		0.241	20.972	0.743	0.228	0.852	0.739	0.157	21.072	0.726	1.542	19.138
	我们的						0.984	0.166				1.257	19.591
RGB-	ORB-SLAM3 [2]	1.724	-	-	-	0.385	-	-	-	1.698	-	-	-
	机器人猛击[34]	91.985	19.317	3.359	3.094	41.833	-	-	-	160.141	-	-	-
	尼斯大满贯[46]	2.119	12.003	0.417	0.510	17.497	36.103	18.200	0.603	0.313	31.448	22.225	25.309
	伊斯兰教[16]	4.571	0.561	0.484	16.419	0.482	0.727	0.233	31.347	19.176	0.595	0.374	25.808
	联合SLAM [36]	1.891	0.591	15.794	0.531	0.538	31.788	16.118	0.534	0.419	0.360	23.127	25.374
	去 SLAM [44]	2.603	18.273	0.663	0.338	20.403	0.780	0.149	0.361	22.570	0.777	0.158	26.802
	我们的（杰森一家）		0.728	0.251	20.870	0 .743	0.346	22.094	0.765	0.169		1.874	19.781
	我们的（笔记本电脑）		0.239									1.315	21.569
	我们的											1.001	22.744

表 2. TUM RGB-D 数据集上的定量结果。我们将最好的两个结果标记为第一和第二



(一)Go-SLAM [44]

(b)Orbeez-SLAM [4]

(c)我们的

图 7. Replica office3 场景上与其他单目相机系统的映射比较。

但几乎没有得到前实时
密集 SLAM 系统。然而，Photo-SLAM 已经
设计为与立体相机兼容。我们在表3中提供了 EuRoC 数据集的定量结果
和定性结果的补充。结果表明
我们的系统在立体场景中仍然可以表现良好。
此外，我们使用手持立体相机收集了一些
户外场景，以及Photo-SLAM的建图结果
如图9所示。

4.3. 消融研究

我们提出了基于几何的致密化（Geo)和
基于高斯金字塔（GP)的学习来增强系统
实时真实感贴图的性能。在此

(a)基本事实　(b)我们的（单声道）　(c)我们的（RGB-D）

图 8.Photo-SLAM 在数据集 TUM 上的定性结果。

在 Euroc 立体声音响上		ORB-SLAM3	DROID-大满贯	我们的（杰森一家）	我们的（笔记本电脑）	我们的
MH-01	均方根误差（厘米）↓	4.379	39.514	4.207	4.049	4.109
	峰值信噪比↑	13.979	13.962	13.952	-	
	SSIM ↑	0.426	0.421	0.420	-	
	低息贷款利息 ↓	0.428	0.378	0.366	-	
MH-02	均方根误差（厘米）↓	4.525	39.265	4.193	4.731	4.441
	峰值信噪比↑	14.210	14.254	14.201	-	
	SSIM ↑	0.436	0.436	0.430	-	
	低息贷款利息 ↓	0.447	0.373	0.356	-	
V1-01	均方根误差（厘米）↓	8.940	21.646	8.830	峰值信	8.836 8.821
	噪比 ↑	16.933	17.025	17.069	-	
	SSIM ↑	-	-	0.626	0.622	0.618
	LPIPS ↓	-	-	0.321	0.284	0.266
V2-01	均方根误差（厘米）↓	26.904	15.344	26.643	26.736	26.609
	峰值信噪比↑	-	-	16.038	16.052	15.677
	是 ↑	-	-	0.643	0.635	0.622
	低功率IPS ↓	-	-	0.347	0.314	0.323

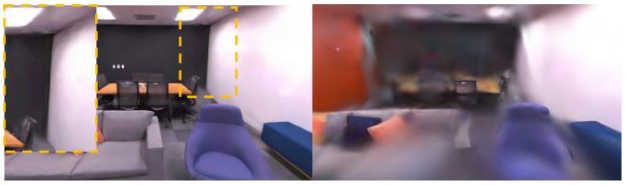
表 3. EuRoC MAV 数据集的定量结果，使用
立体输入。我们的 Photo-SLAM 是第一个支持使用立体相机进行在线真实感地图绘制的系统。

部分，我们构建了一个消融研究来衡量每种算法的有效性，可以通过 PSNR 来量化，
渲染速度 (FPS) 和最终模型大小（以兆字节为单位）
（MB）。定量结果如表4所示。

图 9. 使用手持设备的 Photo-SLAM 测绘结果
室外无边界场景中的立体摄像机。

在副本上	单核细胞增多症	RGB-D
# 地理 GP PSNR ↑ FPS ↑ MB PSNR ↑ FPS ↑ MB		
(1)含/开 = 2 31.274 994.2 10.742 33.296 923.0 18.199		
(2)有 有 20.002 353.2 44.100 33.696 860.0 31.856		
(3) 不含/不含 22.913 645.0 5.782 32.551 1010.8 13.901		
(4)含数=1 30.903 803.8 21.819 34.634 953.7 31.552		
(5)含数=3 31.563 877.6 22.510 33.305 946.2 31.039		
默认w/n = 2 33.302 911.3 31.419 34.958 1084.0 35.211		

表 4.基于几何的致密化（Geo)和基于高斯金字塔（GP)的学习效果的消融研究。



(a)无地理　　(b)无GP

图 10. 不同烧蚀系统在
单目复制场景。

渲染质量与
速度,以及代表的超基元的数量
模型大小。模型通常很小,没有基于非活动 2D 的超基元主动致密化

几何特征（Geo）。然而,如果没有 Geo,PSNR
单眼和双眼上分别下降 2.028 和 1.662
RGB-D场景,如表4（1)所示。
与图7c 相比,没有 Geo
（图10a)显示了一些瑕疵,比如天花板上的瑕疵。在 RGB-D 场景中,更多的超基元可以获得更高的 PSNR。
然而,如果没有基于高斯金字塔的学习,
超原始元素通过 Geo 进行致密化,从而导致
尤其是地图质量和渲染速度的下降
在单目场景中,如图10b所示,并在表 4(2) 和 (3) 中报告。这是因为没有精确深度信息的致密超基元位置不准确。如果没有彻底的优化,不准确的
超原始元素成为累赘。值得注意的是
采用 GP 学习的系统通常表现更好,
强调 GP 学习的有效性。此外,
增加高斯金字塔级别可以改善映射
质量,我们的 Photo-SLAM 默认使用 3 级 GP 学习,取得了最佳效果。然而,我们发现
使用 4 级高斯金字塔的结果变差,如表4(5) 所示,可能是由于过度拟合
增量映射过程中的低级特征。由于
图像是通过绘制超原始图形来渲染的,渲染速度理论上与可见

当前视图中的超基元,而不是模型中的超基元
整个场景的大小。虽然较小的模型
精确重建的模型应该能够准确地捕捉其本质细节,同时使用简洁的参数,这是

高速渲染的前提。此外,减少
渲染所需的时间使得在线映射期间能够通过更多迭代来优化超基元,
最终提高准确性和质量。
总之,消融研究证实
基于几何的致密化策略允许系统
获得足够的超基元,而基于高斯金字塔的学习保证了超基元的彻底优化,增强了在线真实感地图绘制

性能。毫无疑问,默认的Photo-SLAM能够用更合适的超基元重建地图,并实现更好的渲染质量和高

渲染速度比烧蚀系统快。

5. 结论

在本文中,我们提出了一种新颖的 SLAM 框架
称为 Photo-SLAM,用于同时定位和照片级真实感地图绘制。我们引入了超原始地图,而不是高度依赖资源密集型隐式表示和神经求解器。它使我们的系统

利用明确的几何特征进行定位和
隐式地捕获场景的纹理信息。
除了基于几何的致密化之外,我们还提出了
基于高斯金字塔的学习,一种新的渐进式训练方法,进一步提升了地图构建性能。大量实验表明,Photo-SLAM

显著优于现有的在线 SOTA SLAM
真实感映射。此外,我们的系统验证
通过在
嵌入式平台,凸显其在先进
机器人在现实场景中的应用。
致谢:这项工作部分由创新和技术支持

创新及科技基金计划
（编号:ITS/200/20FP）、海洋保护促进基金（MCEF20107 和 MCEF23EG01）、
以及香港科技大学的内部资助（R9429）。

Photo-SLAM:实时同步定位和真实感地图构建
适用于单目、立体和 RGB-D 相机

补充材料

图 11. 所提出的系统具有实时性能
嵌入式平台,例如 Jetson AGX Orin 开发套件。

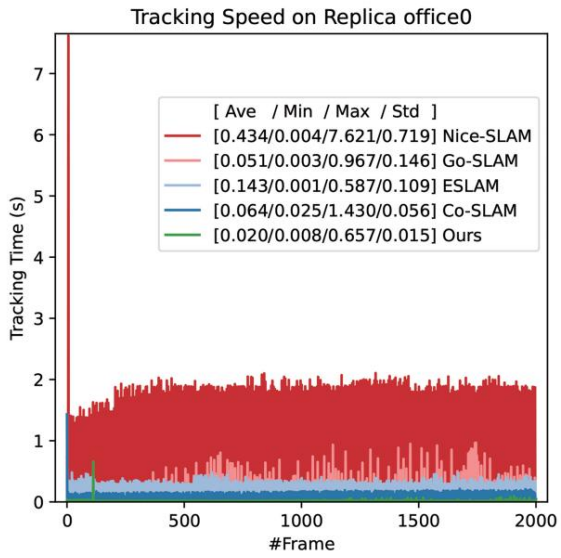


图 12. 使用场景 office0 的跟踪速度比较
RGB-D相机。纵轴表示处理时间
每一帧,横轴表示帧号。
[Ave/Min/Max/Std]分别代表平均、最小、最大跟踪时间及其标准差。

Photo-SLAM 是一种用于同时定位和真实感地图绘制的新型系统,甚至可以在
嵌入式平台实时速度,如
图11.在此补充中,我们提供了额外的结果
关于定位和映射性能。

6. 本地化

稳定性。作为在线系统,SLAM 需要处理传入的帧并估计当前的相机姿势
及时。因此,跟踪延迟和稳定性
除了位姿估计之外,平均处理时间也是评价系统性能的一个重要因素
准确度。如主要论文表 1 所示,Photo-SLAM 每秒能够处理超过 40 帧
第二名,姿态估计准确。平均跟踪速度比 ESLAM [16]快六倍左右,

比 Co-SLAM [36] 快三倍。在这里,我们提供
对跟踪稳定性进行附加分析,同时举例说明
绘制在图12中。

尽管 Go-SLAM [44]的平均跟踪时间
小于Co-SLAM和ESLAM,处理延迟
由于频繁进行昂贵的全局优化,因此成本很高。如图12 所示, Go-SLAM 通常需要
1 秒来处理帧并估计姿势。此外,Nice-SLAM [46]和 Co-SLAM 都需要更长的时间
时间来准确地初始化跟踪。显然,我们的
系统可以快速稳定地处理传入的帧,
具有最小的平均跟踪时间和标准偏差。我们系统的峰值处理时间发生在
检测回环以校正姿势估计漂移。
准确性。图13 展示了 Photo-SLAM 的一些定性跟踪结果。

7.讨论

在线制图与离线制图。对于在线制图,制图过程与

本地化过程。因此,它需要持续和
随着机器人每次新的观察,及时更新
或相机移动并观察周围环境。一般来说,在线真实感地图绘制比

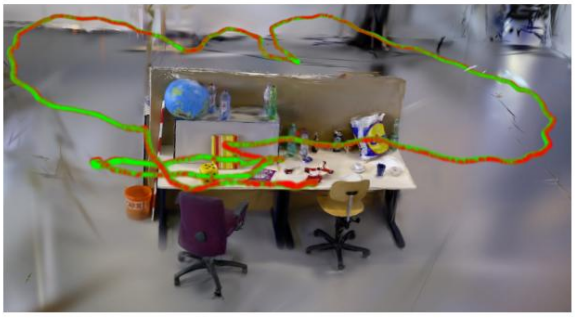
离线真实感地图绘制,因为平衡计算效率和

渲染质量。正如在主要论文中提到的,我们
提出了一种基于几何的致密化策略和
基于高斯金字塔 (GP)学习方法实现
高质量的在线建图。为了进一步支持这一说法,我们比较了我们的
Photo-SLAM 和 3D 高斯分层 (3DG) [18]之间的照片级真实感建图
性能。3DG是 SOTA 离线方法,它

拍摄一组具有已知姿势和稀疏点的图像
云作为输入来学习用于视图合成的辐射场。
在实验中,3DG 使用关键帧姿势估计

(a)复制 office0

(b)副本 room0



(c)TUM fr3办公室

图 13. 重建地图中的轨迹。绿点表示地面真实轨迹,红点表示 Photo-SLAM 的估计轨迹。

与 Photo-SLAM 配对并进行相同的训练
持续时间与 Photo-SLAM 相同。所需的点云输入是
以三种不同的方式初始化:1)随机初始化
100 个点;2)随机初始化 10,000 个点;3)
从 Photo-SLAM 的超原始地图进行初始化。
结果如表5所示。如果不输入细粒度点云,3DG需要更多时间进行优化,导致渲染质量下降。此外,

为了提高渲染质量,3DG 倾向于密集化点
云导致模型尺寸更大,渲染速度更慢
速度。无论是使用单目相机还是 RGB-D 相机,Photo-SLAM 都能始终提供引人注目的渲染质量和更快的渲染速度,这要归功于所提出的算法的有效性。

方法	PSNR ↑ SSIM ↑ LPIPS ↓	渲染 ↑	模型尺寸 (MB)
1) 3DG	27.844 0.861 0.213		745.480	36.141
2) 3DG	34.555 0.942 0.065		483.904	144.196
3) 3DG	37.055 0.962 0.032		448.109	219.470
我们的 Mono	33.302 0.926 0.078		911.262	31.419
我们的 RGB-D	34.958 0.942 0.059		1084.017	35.211

表 5. 3D 高斯分层 (3DG) 和我们的系统 Photo-SLAM 在不同条件下的映射性能比较副本数据集上的设置。

在 TUM 数据集上			资源		
场景	摄像机	方法	追踪 ↑	渲染 ↑	模型尺寸 (MB)
办公室	单核细胞增多症	我们的（Jetson）	28.267	我们的	340.507	4.610
		（笔记本电脑）	28.330	我们的	1105.062	7.421
		57.781			2016.690	10.027
	RGB-D	我们的（杰森一家）	27.970		380.622	5.743
		我们的（笔记本电脑）	28.930	我	1061.040	8.432
		们的58.378			2083.896	9.963
客厅	单核细胞增多症	我们的（Jetson）	24.005	我们的	169.321	14.286
		（笔记本电脑）	24.922	我们的	619.554	16.102
		58.241	我们的	(Jetson)	1405.797	20.380
	RGB-D	21.032	我们的（笔记本电脑）		274.718	6.319
		22.665	我们的52.904		701.590	13.850
					1790.120	21.399
机房	单核细胞增多症	我们的（Jetson）	36.700	我们的	291.398	10.669
		（笔记本电脑）	38.929	我们的	824.658	16.249
		81.575	我们的	(Jetson)	1522.120	19.211
	RGB-D	18.039	我们的（笔记本电脑）		291.907	12.726
		19.636	我们的43.650		764.342	15.349
					1540.757	17.009

表 6.TUM 数据集上不同平台的 Photo-SLAM 的附加结果。

在 EuRoC 数据集上		资源		
场景	方法	追踪 ↑	渲染 ↑	模型尺寸 (MB)
MH-01	我们的（Jetson）	21.359	我们的	93.762	43.385
	（笔记本电脑）	25.019	我们的	316.403	89.700
	44.977			613.958	123.528
MH-02	我们的（Jetson）	22.355	我们的	101.021	36.263
	（笔记本电脑）	26.189	我们的	332.174	81.569
	46.556			675.508	113.116
V1-01	我们的（Jetson）	21.332	我们的	106.008	28.444
	（笔记本电脑）	25.403	我们的	367.903	55.263
	44.763			835.119	74.457
V2-01	我们的（Jetson）	23.872	我们的	99.988	27.840
	（笔记本电脑）	27.556	我们的	307.025	62.588
	48.911			595.234	82.600

表 7. 不同平台的 Photo-SLAM 在 EuRoC MAV 立体数据集上的附加结果。

8. 更多结果

副本数据集各场景结果详解
表8 TUM 数据集上的附加定性结果
在图14和图15 中进行了演示,而图16则说明了 EuReC Stereo 上的 Photo-SLAM 的定性结果