

# MARSIM: 一个针对基于 LiDAR 的无人机的轻量级点云真实感仿真器

Fanze Kong<sup>1</sup>, Xiyuan Liu<sup>1</sup>, Benxu Tang<sup>2</sup>, Jiarong Lin<sup>1</sup>, Yunfan Ren<sup>1</sup>,  
Yixi Cai<sup>1</sup>, Fangcheng Zhu<sup>1</sup>, Nan Chen<sup>1</sup>, Fu Zhang<sup>1</sup>

**Abstract**—低成本、小型化和轻量级的固态 LiDAR 传感器的出现，为自主无人机 (UAV) 通过提升导航安全性和计算效率带来了新的机遇。然而，基于 LiDAR 的无人机的成功开发必须依赖于大量的仿真。现有的仿真器由于需要难以获取的密集网格地图，难以进行真实环境的仿真。本文中，我们开发了一个针对基于 LiDAR 的无人机的真实点云仿真器。其关键思想是底层的点云渲染方法，通过直接从点云地图构建深度图像，并进行插值以获得真实的 LiDAR 点测量值。我们开发的仿真器能够在轻量级计算平台上运行，并支持不同分辨率和扫描模式的 LiDAR 仿真（例如，旋转 LiDAR 和固态 LiDAR）、动态障碍物以及多无人机系统。该仿真器基于 ROS 框架开发，可与自主机器人其他关键模块（如感知、状态估计、规划和控制）轻松通信。最后，该仿真器提供了 10 个高分辨率的各种真实环境点云地图，包括不同密度的森林、历史建筑、办公室、停车场和各种复杂的室内环境。这些真实的地图为自主无人机提供了多样化的测试场景。评估结果表明，开发的仿真器在时间和内存消耗方面优于 Gazebo，且模拟的无人机飞行与真实环境中的飞行高度匹配。我们相信，这种真实点云且轻量级的仿真器对于弥合无人机仿真与实验之间的差距至关重要，并将显著促进未来基于 LiDAR 的自主无人机研究的发展。

仿真器能够在轻量级计算平台上实现这些地图的实时高分辨率仿真，无需独立显卡。此外，我们还提供了一个教程，利用一组现有的 LiDAR 建图方法，方便用户使用自己的 LiDAR 采集数据并构建高分辨率点云地图进行仿真，这将极大地促进无人机研究者的实验工作，特别是在运动规划和自主探索方面。

我们的代码已在 [Github](#) 开源<sup>2</sup>。

**Index Terms**—Aerial Systems; Simulator, LiDAR, Perception and Autonomy

## I. 引言

近年来 LiDAR 技术的发展显著降低了 LiDAR 传感器的成本和重量，为无人机 (UAV) 应用创造了诸多机会，如矿山勘探 [1]、生物数据统计 [2]、测绘 [3]、高速导航 [4]

<sup>1</sup>These authors are with the Department of Mechanical Engineering, University of Hong Kong. {kongfz,xliuua,jiarong.lin, renyf,yixicai,zhufc,cnchen}@connect.hku.hk and {fuzhang}@hku.hk. <sup>2</sup>Benxu Tang is with the School of Mechanical Engineering and Automation, Harbin Institute of Technology. 180320222@stu.hit.edu.cn. (Corresponding author: Fu Zhang).

<sup>2</sup>[https://github.com/hku-mars/dyn\\_small\\_obs\\_avoidance.git](https://github.com/hku-mars/dyn_small_obs_avoidance.git)

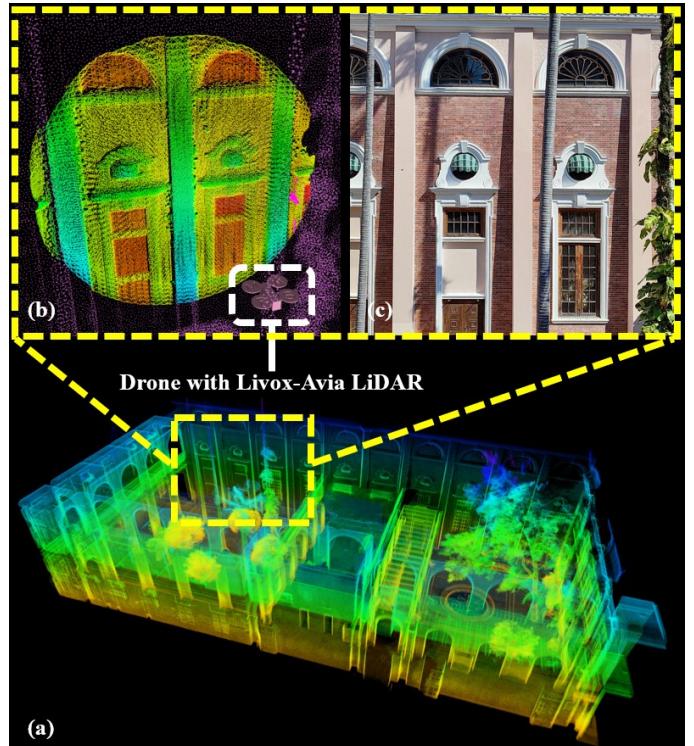


Fig. 1: MARSIM 演示图。(a) 为香港大学主楼的点云地图，是 MARSIM 中十个真实场景之一；(b) 由我们的仿真器直接从点云地图渲染的 Livox Avia LiDAR 点扫描；(c) 为对应的真实场景照片。可见仿真器能够高质量地还原真实场景的结构细节。更多细节可见附带视频（也可见于 <https://youtu.be/hiRtcq-5IN0>）。

和避障等。然而，将无人机部署到这些广泛应用中需要大量测试，这些测试通常成本较高，因为被测试系统仍处于活跃开发中，可能存在明显的失败率（例如与环境碰撞）。一个贴近现实的仿真器可以显著减少无人机测试中产生的时间和设备成本，并已成为无人机开发的重要组成部分。然而，随着无人机应用复杂性的增加，现有仿真器逐渐无法满足基于 LiDAR 的无人机研发需求。

现有大多数仿真器（如 Gazebo [5]、Webots [6]、Airsim [7]）由于以下限制，很难满足基于 LiDAR 的无人机对高分辨率真实场景仿真的需求：(i) 主流仿真器能模拟的环境多为虚拟的、非真实的简单人工环境，与复杂的真实世界场景存在显著差距；(ii) 现有仿真器只支持导入网格地

图, 而真实世界环境通常以激光扫描仪或 LiDAR 采集的 3D 点云数据形式存在, 获取高分辨率且高保真度的网格地图的开源成熟工具尚不可得。常用的泊松重建方法 [8] 由于遮挡和大场景扫描中点密度变化, 处理真实 LiDAR 采集的点云数据时计算耗时且网格质量较低; (iii) 主流仿真器往往依赖高性能 GPU 实现大规模复杂网格地图的实时仿真, 对计算平台要求较高。

针对上述不足, 本文提出了一个轻量级基于 LiDAR 的无人机仿真器, 具备以下特性:

- 1) 直接利用真实环境的点云地图渲染真实的 LiDAR 扫描, 点云地图保留了环境的细节, 且可通过 LiDAR 传感器轻松获取。
- 2) 计算和内存消耗高效, 能够在无独立显卡的个人电脑上运行。
- 3) 功能多样, 支持动态障碍物、多无人机系统以及不同分辨率和扫描模式的多种现有 LiDAR 型号 (Livox AVIA、Livox MID-360、Velodyne VLP-32、IntelRealsense D455 等) 仿真。
- 4) 开源且兼容 ROS (<https://github.com/hku-mars/MARSIM.git>), 用户可轻松将仿真器与 ROS 中开发的模块 (如同步定位与建图 (SLAM) 及路径规划模块) 集成, 并快速在真实仿真环境中进行评估。

## II. 相关工作

基于 LiDAR 的无人机仿真包括无人机运动仿真和 LiDAR 仿真。相比 LiDAR 仿真, 无人机运动仿真相对直观且成熟。MATLAB 支持多种类型无人机的运动仿真与控制器设计, 如四旋翼 [9] 和垂直起降无人机 (VTOL UAV) [10]。常见的仿真器如 Gazebo [5] 和 Airsim [7] 也能实现高频率的无人机运动仿真。Wei 等人 [11] 利用 Gazebo 仿真开发了多无人机路径规划算法, Han 等人 [12] 在 Airsim 中模拟了无人机的 SE(3) 规划并提供了自主无人机竞速的基准。

基于 LiDAR 的无人机仿真主要挑战在于 LiDAR 仿真。目前已有许多支持 LiDAR 仿真的仿真器, 如 Gazebo [5]、Webots [6]、Airsim [7] 和 SVL [13]。Gazebo 可能是移动机器人研究中最常用的仿真平台, 用户可构建自定义机器人和 LiDAR 传感器。该仿真器广泛应用于基于 LiDAR 的自主探索算法验证, 如 GBP [1]、MBP [14]、TARE [15] 和 Splatplanner [16]。SVL 在城市环境中模拟了车辆顶部安装的 LiDAR, 用于自动驾驶应用。

这些仿真器的主要缺点是它们导入的是网格模型格式的地图, 通常只适用于由 3D 建模软件 (如 Sketchup、Blender、

3DS Max 等) 创建的人工环境, 或基于 Gazebo 模型库构建。例如, Splatplanner 在 Gazebo 中制作了十一张人工地图 [16, 17], TARE 则制作了五张较大的人工地图用于算法验证。此外, DARPA 地下竞赛 [18] 提供了一组隧道和矿山地图, 其中大部分为手工制作。人工手工制作的地图虽能满足部分基础仿真需求, 但大多数相对简单且不够真实, 与复杂的真实世界环境存在较大差距。

现实环境更常见的表示形式是点云, 可由 3D 激光扫描仪或 LiDAR 传感器采集。为缩小仿真与现实的差距, 一些工作尝试从点云生成真实的网格地图。例如, LiDARsim [19] 利用基于表面元素 (surfel) 的方法 [20] 生成真实的网格地图, 用于真实感自动驾驶仿真。其他方法也用于从点云构建真实网格地图, 如泊松重建 [8]、截断有符号距离场 (TSDF) 体素方法 [21] 和德劳内三角剖分法 [22, 23]。然而, 这些方法对遮挡和点云密度变化的鲁棒性不足, 而这些现象常出现在 LiDAR 采集的大规模场景中, 可能错误生成不存在的表面, 需要额外人工修正或参数调节。为解决此问题, FlightGoggles [24] 在商业软件 Reality Capture 中分别构建场景中每个独立物体, 然后与背景合成, 打造高度真实的仿真环境。但该方法扩展性差, 建模和合成过程耗时耗力 (FlightGoggles 仅提供两个场景)。此外, 高分辨率网格模型若无高性能 GPU 支持, 实时渲染也较为困难。

是否真的需要网格地图进行 LiDAR 仿真成为一个问题。网格模型的优势在于可附加来自相机图像的材质纹理, 从而渲染用于相机仿真的图像。但对于 LiDAR 仿真, 这种纹理并非必需。且随着低成本高精度 LiDAR 传感器和同步定位与建图 (SLAM) 算法 [25, 26] 的发展, 获取真实环境的高精度点云变得更加经济且易得。基于此趋势, 我们选择直接使用点云地图进行仿真, 而非网格地图。类似思路已在 FUEL [27] 中初步探索, 但该仿真器分辨率和精度有限, 因计算成本高仅适用于小场景深度相机。

## III. 系统概述

如图2所示, 我们的无人机仿真器主要由三个子模块组成: 内置飞行控制器模块、动力学与运动学仿真模块以及 LiDAR 仿真模块 (图中黑色模块, 见图2)。该仿真器能够与 ROS 框架中的规划器、SLAM 算法和可视化模块交互, 构成一个完整的基于 LiDAR 的无人机仿真系统。

使用该仿真器时, 用户应首先选择 LiDAR 型号并提供环境的点云地图。随后, 用户可通过图2中所示的 ROS 话题名称, 将自有的 SLAM (或使用真实里程计) 和规划算法接入无人机仿真器进行验证和可视化。一旦仿真器启动, 动力学与运动学仿真模块开始计算无人机的里程计和 IMU

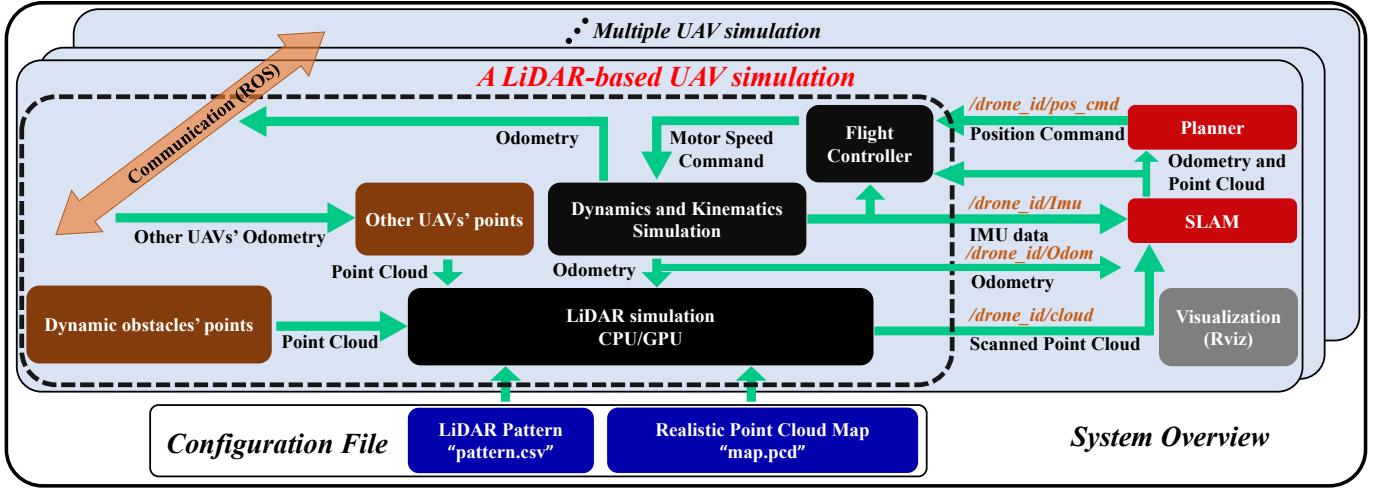


Fig. 2: 我们仿真的整体框架（黑色虚线框）及其与 ROS 中外部模块的交互方式。

数据，LiDAR 仿真模块则根据这些数据渲染 LiDAR 扫描点云。仿真产生的 IMU 数据和 LiDAR 扫描点云在 ROS 中发布，可被 SLAM 模块使用，进而传递给规划模块。除了由点云地图表示的静态环境，LiDAR 仿真还实时模拟动态障碍物和其他无人机上的点测量。

在仿真器中，最重要的模块是 LiDAR 仿真模块，它能够高效地从点云地图中提取 LiDAR 扫描的点云作为传感器的输出数据。该模块通常是无人机仿真中最耗时的模块，因此我们提出了自有的点云渲染器，包含 CPU 版本和 GPU 版本。CPU 版本仅使用 CPU 进行计算，适合低分辨率地图仿真；GPU 版本利用 GPU 资源以实现更高效的渲染。仿真器基于 OpenGL [28] 框架，使用 GPU 硬件加速渲染。OpenGL 是目前最常用的开放图形库，市面上几乎所有集成显卡和独立显卡均支持该库。相比 CUDA 图形加速库（仅支持 NVIDIA 独立显卡），硬件要求更低。

#### IV. 方法论

##### A. LiDAR 仿真

给定环境的点云表示（见第V节）及当前 LiDAR 位姿，LiDAR 仿真模块旨在渲染当前 LiDAR 扫描中应测得的点。为此，我们首先将点云地图的所有点投影到当前 LiDAR 视场 (FoV) 内，并进行插值以获得尺寸为  $\theta_{\text{res}}$  角分辨率的稠密深度图像，然后对非扫描模式上的点进行掩码处理。最终剩余的深度像素加入 LiDAR 测量噪声后转换为点云进行发布（见图3）。为了加速点投影过程，我们在导入点云地图时进行了两个关键预处理步骤：1) 为限制地图点数，对所有地图点执行空间下采样，空间分辨率为  $r_{\text{map}}$ ；2) 将地图空间划分为大小相等的体素（边长为  $l$ ），并将每个体素包含的地图点保存到对应的点列表中。渲染 LiDAR 点云时，我们筛选与当前 LiDAR 视场相交的体素，仅投影这些体素内的点。

1) 遮挡剔除：空间下采样虽有效限制了地图点数，但带来两个问题。一是由于投影原理，投影点无法均匀覆盖深度图：靠近 LiDAR 的点非常稀疏，远处点非常密集，导致大量深度像素为空（如图3(a) 所示）。另一个问题是，投影到深度图后，背景物体上的点会干扰前景物体上的点，导致错误的深度测量。为解决这些问题，我们执行如下遮挡剔除过程。

由于空间下采样分辨率为  $r_{\text{map}}$ ，地图中的一个点实际上应代表一个边长为  $r_{\text{map}}$  的实体立方体（见图4(a)）。因此，深度图中围绕地图点  $\mathbf{p}$  的插值范围  $\theta_{\text{max}}$  为该点对应立方体的投影面积：

$$\theta_{\text{max}} = \arcsin \left( \frac{\sqrt{3} r_{\text{map}}}{d} \right) \quad (1)$$

其中  $d$  为地图点  $\mathbf{p}$  的深度值。插值范围  $\theta_{\text{max}}$  内的像素其深度值被设置为  $d$ 。若不同地图点的插值范围重叠，重叠像素保留最小深度值。

2) 平面校正：当 LiDAR 激光射线垂直于物体表面时，遮挡剔除效果良好。然而，当激光射线不垂直于表面时，插值点会偏离表面（见图4(b) 中的蓝色点云），导致意外的错误点测量。为解决此问题，在插值邻近地图点时，我们将地图点视为一个小平面，并通过计算像素射线与平面的交点来插值插值范围  $\theta_{\text{max}}$  内的像素（见图4(c))。每个地图点周围的小平面在预处理阶段由其邻近点拟合得到，拟合得到的平面法向量随地图点一同保存。由于从平面计算深度较耗时，我们仅对真实处于平面上的地图点执行此平面校正。该判定通过为每个地图点引入新变量  $\gamma$  实现， $\gamma$  表示该点的平面质量。 $\gamma$  值为平面拟合过程中的平面厚度，并与平面法线及地图点一同保存。在线渲染时，只有  $\gamma$  低于某阈值的地图点才执行平面校正。

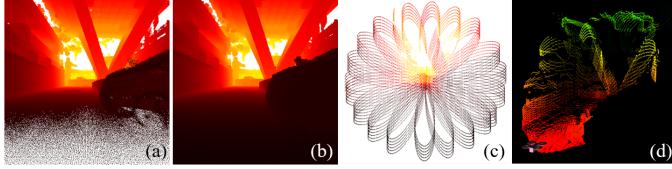


Fig. 3: LiDAR 仿真示意图。(a) 为直接将地图点投影到图像上的稀疏深度图。(b) 为插值后的稠密深度图。(c) 为根据 LiDAR 扫描模式 (Livox Avia LiDAR) 掩码后的深度图。(d) 为最终输出的 LiDAR 点云。

3) GPU 加速: 随着地图分辨率的提升, 点云规模逐渐增至数百万, 投影地图点到深度图像的时间显著增加。此时, 仅使用中央处理器 (CPU) 已难以满足实时需求。为解决该问题, 我们利用图形处理单元 (GPU) 硬件加速 LiDAR 仿真过程。经济实惠的集成或独立 GPU 在标准个人计算机中广泛存在, 因而不会降低我们仿真器的通用性。GPU 也得到了跨平台通用图形渲染库 Open Graphics Library (OpenGL) [28] 的良好支持, 该库可高效利用 GPU 资源完成我们任务中的点云投影至深度图。我们基于 OpenGL 提出了一种点云并行渲染器以加速深度图构建。借助 GPU 加速, 仿真器能够在拥有千万级点云规模的地图上实现实时运行 (超过 10 Hz)。

#### B. 动力学与运动学仿真

为了模拟真实的无人机飞行, 仿真器基于标准刚体模型 [29] 提供无人机的动力学与运动学仿真。动力学模型中的推力和转矩由二阶电机模型生成, 该模型的控制指令为期望电机转速 [30]。模型参数 (如惯性矩阵、质量、桨叶转矩、推力系数、电机 KV 值) 取自文献 [29], 并汇总于独立配置文件, 用户可根据需要修改。计算得到的角速度和特殊加速度加入测量噪声及偏置后, 生成用于发布的 IMU 测量数据。完整的真实无人机状态也被发布供外部参考。

#### C. 飞行控制器设计

在无人机运动仿真之后, 需要控制器来精确控制无人机飞行。我们的仿真器采用了文献 [31] 中所示的级联双环 PID 控制器, 其中内环为姿态控制器, 外环为位置控制器。根据无人机动力学和运动学模型参数, 控制器增益基于期望的自然频率和系统阻尼比进行调节, 以实现良好的位置控制性能。

#### D. 动态障碍物仿真与碰撞检测

为了模拟可能出现动态障碍物的真实环境, 仿真器支持动态障碍物的仿真。我们随机生成一定数量的球形动态物体及其表面点云。每个球形物体以恒定速度沿随机方向运

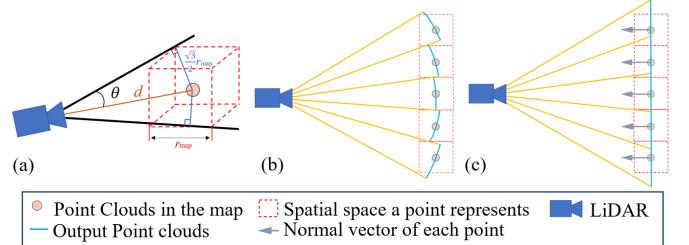


Fig. 4: 遮挡剔除和平面校正原理。(a) 展示了计算每个点插值范围的几何原理。(b) 展示了遮挡剔除在大平面上的问题。(c) 展示了平面校正。

动。当某球体移出地图边界时, 系统会在地图中随机位置生成新的球体。用户可根据需要修改球体的数量、大小和速度。

为了尽可能真实地模拟环境, 仿真器在每个仿真步进行碰撞检测, 判断无人机是否与环境中的障碍物 (静态或动态) 发生碰撞。为此, 仿真器建立两个 KD 树, 一个在预处理阶段由全局地图点构建, 另一个在每个仿真步由动态物体上的点构建。随后, 仿真器搜索距离无人机尺寸范围内的最近邻点。如在该范围内找到任意点, 则判断无人机发生碰撞, 并输出碰撞警告。

#### E. 去中心化多无人机仿真

无人机研究中的一个显著趋势是群体导航与控制。为支持此类研究, 我们的仿真器支持无人机群系统的仿真。为分摊计算负载并提升仿真对群体规模的扩展性, 仿真完全去中心化, 每架无人机在独立线程 (作为 ROS 节点) 或独立计算机中运行。不同线程或计算机通过 ROS 通信进行交互。如此, 多架无人机仿真可分布在多台通过局域网 (LAN) 连接的轻量级计算机上。为更真实地模拟多无人机间的交互, 仿真器新增互相观测功能: 在每个仿真步骤中, 于无人机表面 (近似为无人机尺寸的立方体) 采样一组点, 并将这些采样点添加至全局点云地图中, 用于渲染其他无人机的 LiDAR 扫描。

#### V. 真实高分辨率点云地图构建

为了使无人机在仿真器中与环境实现真实交互, 我们从实际场景构建点云地图。为尽可能还原真实环境, 我们对点云地图提出了两项要求: 高分辨率和高精度。为满足这些要求, 我们使用携带 Livox Avia 传感器的手持设备 (详见 [32]) 扫描环境。Livox Avia LiDAR 的非重复扫描特性使得即使 LiDAR 处于静止状态, 地图点仍可累积到高分辨率, 从而减少了由 LiDAR 运动引起的点密度变化影响。为了将所有 LiDAR 扫描注册到同一全局坐标系, 我们使用 FAST-LIO2 [25] 构建粗略地图, 随后利用 [26] 通

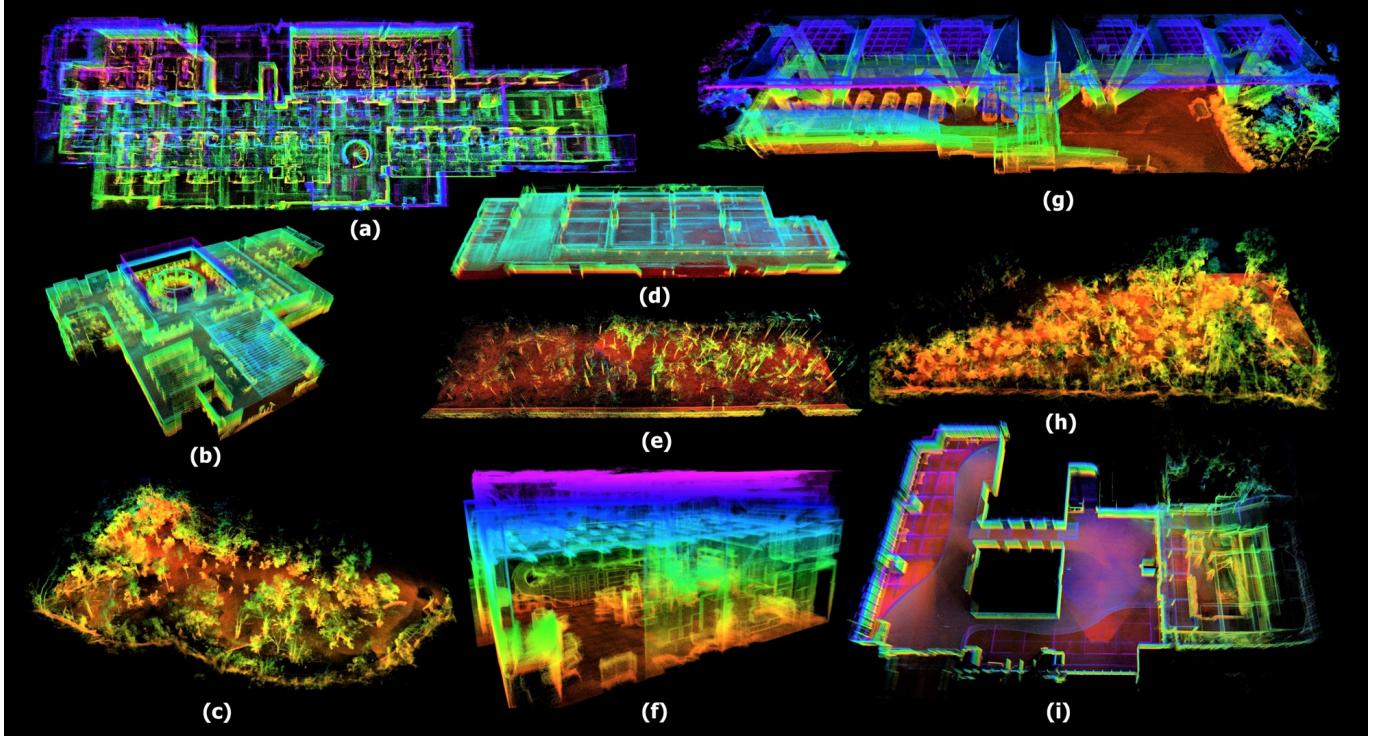


Fig. 5: 仿真器提供的高分辨率点云地图，详细信息见表I。

TABLE I: 十个真实点云地图及其特征。

MARSIM 地图	尺寸 (米)	大规模	多层	杂乱障碍物	狭窄通道	细长结构	真实环境
历史建筑 (图 1 (c))	47×20×23			✓	✓	✓	✓
大型办公室 (图 5 (a))	45×16×6	✓	✓	✓	✓	✓	✓
室内-2 (图 5 (b))	27×40×7			✓	✓		✓
普通森林 (图 5 (c))	48×27×19	✓		✓		✓	✓
简易停车场 (图 5 (d))	45×15×5	✓					✓
简易森林 (图 5 (e))	45×16×6	✓		✓		✓	✓
室内-1 (图 5 (f))	17×13×9			✓	✓	✓	✓
复杂停车场 (图 5 (g))	62×12×10	✓		✓			✓
密集森林 (图 5 (h))	30×72×19	✓		✓	✓	✓	✓
室内-3 (图 5 (i))	21×48×4			✓		✓	✓

过 LiDAR 束调整进行全局地图质量优化。全局配准后的点云地图导入 CloudCompare 软件，使用统计异常点移除 (SOR) 滤波器及空间下采样生成均匀且干净的点云地图。SOR 滤波器通过计算每点与邻近点的平均距离，剔除距离较大的点 (即孤立噪声点)。在滤除异常点后，我们手动填补了一些 LiDAR 扫描无法覆盖的角落点。

通过上述方法，我们扫描了十个真实环境，获得相应的高精度点云地图供仿真器使用。该方法同样适用于获取以网格模型表示的人造环境的点云地图 (如有需要)。

## VI. 结果

### A. 高分辨率真实点云地图

本文提供了十个真实场景的高分辨率 (0.01 m) 点云地图供用户仿真使用，如图5所示。其部分环境特征见表I，可作为选择仿真地图时的参考。此处的 0.01 米分辨率地图指经过 0.01 米空间下采样处理的原始点云。十个地图的场景包括三片森林、三个室内场景、一栋历史建筑 (香港大学主楼)、两个停车场及一个大型办公室。图6中所示的局部点云清晰展示了真实环境中的细节结构。

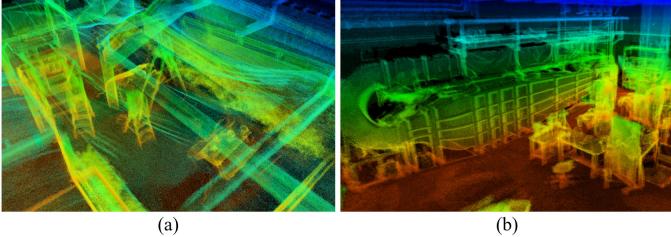


Fig. 6: 室内-1 和室内-2 地图的细节结构。(a) 展示了室内-1 地图中几处清晰的梯子;(b) 展示了一处机械车间内复杂设备的细节。

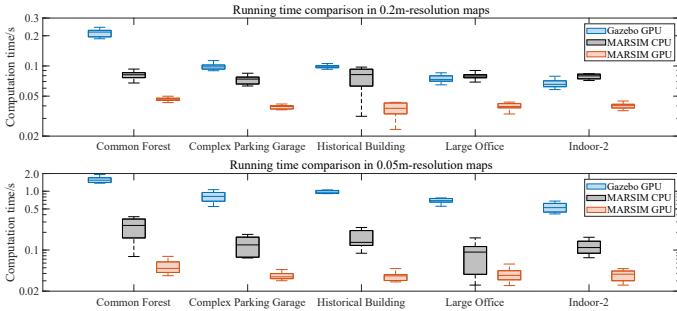


Fig. 7: 在轻量级计算平台 (NUC) 上渲染一次 Livox AVIA 扫描所需时间。

### B. 计算资源消耗拆解

我们比较了 MARSIM 仿真器与 Gazebo 仿真器的时间消耗。由于 Gazebo 只能使用网格模型，我们采用泊松重建方法 [8] 将对应分辨率的点云地图转换为网格模型（见下文）。我们选取五个典型场景，分别比较渲染一次 Livox Avia LiDAR 扫描（视场  $77^\circ \times 70^\circ$ ，分辨率  $385 \times 350$ ，感知距离 30 米）所需的时间和内存消耗。每张地图测试两种情况：高分辨率地图（0.05 米分辨率）和低分辨率地图（0.2 米分辨率）。数据通过随机选择无人机的 10 个位置和偏航角生成。轻量级计算平台 NUC 10 Kit（搭载最高 4.70 GHz 的 i7-10710U CPU，32 GB 内存）上的运行时间对比见图7。可见在低分辨率地图上，即使是 MARSIM 的 CPU 版本，其计算时间也略低于 GPU 加速的 Gazebo 仿真。借助 GPU 加速，MARSIM 速度是 Gazebo 的两倍。在高分辨率地图中，这一差异更加明显：MARSIM 的 CPU 版本速度是 GPU 加速 Gazebo 的两倍，GPU 版本则快十倍。

Gazebo 表现较差的原因在于生成的网格地图中三角面数量庞大（超过 200 万），这是为了保留与对应点云相似的细节水平。相比之下，大多数现有机器人仿真使用非常简单的网格地图，主要由大平面组成，三角面数量较少，能实现实时仿真。此外，作为专为点云设计的仿真器，MARSIM 无需处理完整的网格渲染管线（如减少片段着色器、光线追踪等过程）及复杂物理仿真（如碰撞模拟），显著降低了计算资源消耗。

最后，我们对所有十个地图在 0.05 米和 0.2 米分辨率下使用相同传感器进行了时间消耗测试。如图8所示，在所有

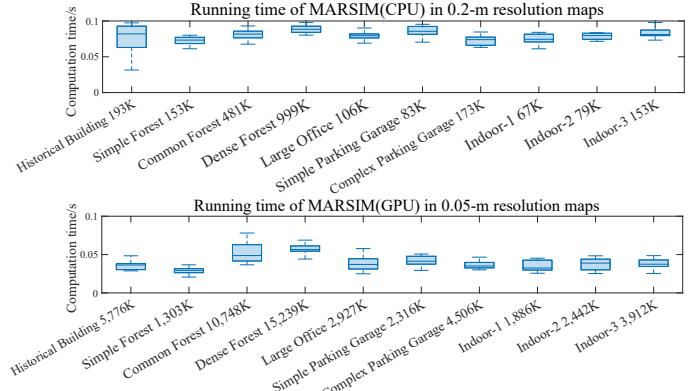


Fig. 8: 在所有 10 个真实点云地图上渲染一次 Livox AVIA 扫描所需时间。横轴标注了点云地图的名称和尺寸。

TABLE II: 在轻量级计算平台 (NUC) 上与 Gazebo 的内存消耗对比。

分辨率	地图	内存消耗 (GB)		
		Gazebo	MARSIM	
		CPU	GPU	
0.2 米	历史建筑	1.64	1.01	1.31
	复杂停车场	1.46	1.1	1.25
	大型办公室	1.68	1.04	1.13
	普通森林	1.48	1.37	1.73
	室内-2	2.09	0.94	1.14
0.05 米	历史建筑	6.97	4.04	3.43
	复杂停车场	6.72	3.51	3.06
	大型办公室	4.93	2.87	2.17
	普通森林	16.88	7.35	3.53
	室内-2	3.73	2.51	1.84

测试中，仿真器均能以 10 Hz 的实时频率运行。

除时间消耗对比外，我们还收集了内存消耗数据，如表II所示。无论 CPU 版本还是 GPU 版本，仿真器的内存消耗约为 Gazebo 的一半，体现了仿真器的轻量级特性。

### C. 实验验证

为验证仿真器能提供与真实实验相似的飞行仿真，我们在真实环境中验证了一种无人机规划方法 Bubble planner [4] 的仿真效果。具体做法是，构建一台搭载 Livox Mid360 LiDAR 的四旋翼无人机，参照文献 [4]。随后手持无人机对真实环境（森林场景）进行扫描，构建点云地图。利用采集的点云地图，在仿真器中模拟 Bubble planner，并将仿真无人机飞行轨迹与实际实验中相同起点和目标位置的飞行轨迹进行对比。对比结果见图9。可见，仿真轨迹与实际轨迹非常接近，验证了仿真器的实用性。

### D. 支持多种类型 LiDAR 及其他功能

为了提升仿真器的通用性，仿真器内置了多种常见 LiDAR 及深度相机模型。如图10所示，支持 Livox Avia、Livox Mid-360、VLP-32、VLP-64、OS1-32 及 Intel Realsense D455 等传感器。仿真器能复现这些传感器的扫描模式，用户可直接使用，无需调整参数。此外，仿真器还支持动态障碍物和多无人机仿真，见图11和图12。

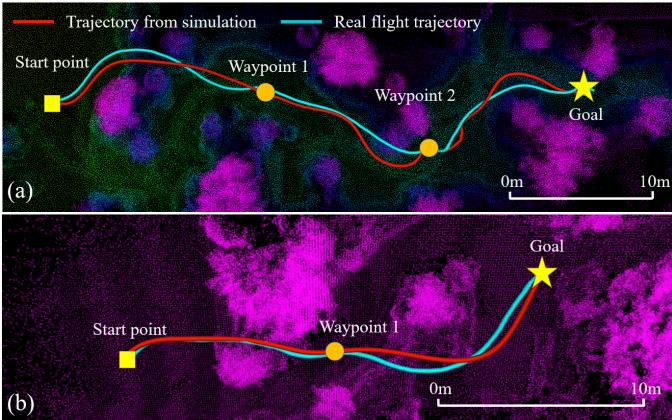


Fig. 9: 仿真与实际环境中飞行轨迹的对比。(a) 和 (b) 分别代表在不同环境下的两组实验。红线为仿真结果轨迹，蓝线为实际飞行轨迹。

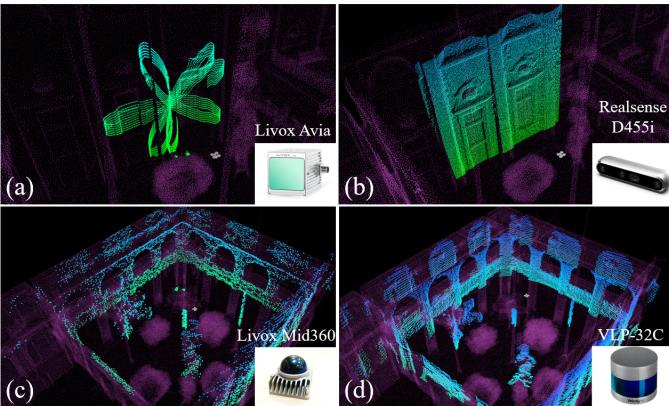


Fig. 10: 多种 LiDAR 扫描模式支持，分别为 Livox Avia (a)、D455 (b)、Livox Mid-360 (c) 和 VLP 32 (d)。

#### E. 仿真器的实际应用

本仿真器主要基于 LiDAR 的无人机算法开发提供测试与验证平台，尤其是需要与环境交互的运动规划和自主探索算法。此前实验已展示了仿真器在无人机运动规划中的应用，我们还进行了自主无人机探索的仿真实验。图13展示了一台搭载 Livox Avia 的无人机在室内-2 地图上，利用 FUEL 算法 [27] 进行自主探索的过程。值得一提的是，仿真器已成功辅助完成多无人机相互定位 [33] 和运动规划算法 [4, 34] 的开发。

## VII. 结论与讨论

本文提出了一种基于 LiDAR 的无人机仿真器，可在轻量级计算平台上进行真实环境仿真。该仿真器直接在点云地图上渲染 LiDAR 扫描，相较于现有仿真器使用的网格模型，更易于捕获真实环境数据。此外，得益于现代 3D LiDAR 和激光扫描仪的高精度，基于真实环境扫描得到的点云地图能真实地表示环境，大大缩小了仿真与现实之间的差距。为最大化仿真器的实用性，我们还提供了十个高分辨率点云地图，并支持多种类型 LiDAR 传感器、动态

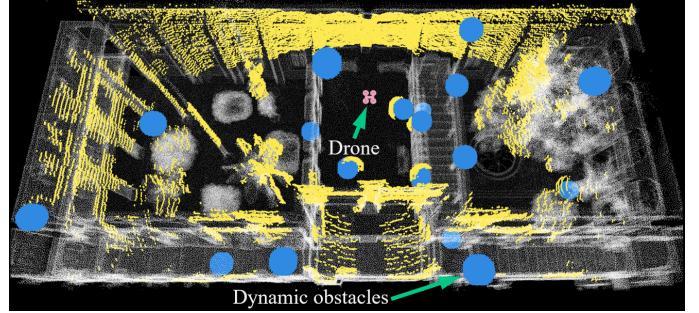


Fig. 11: 历史建筑地图中的动态障碍物仿真。蓝色球体为动态障碍物，黄色点云为 VLP-32 LiDAR 扫描点。

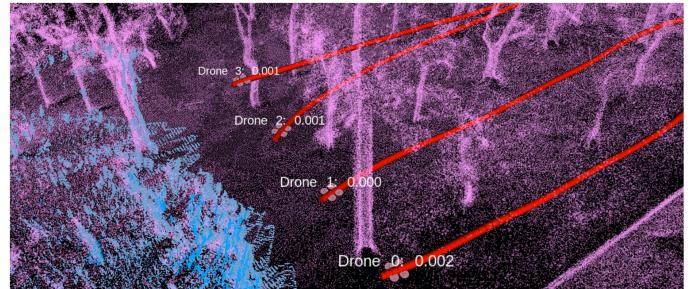


Fig. 12: 多无人机规划仿真。粉色模型为无人机，红色曲线为无人机轨迹，避开真实森林地图中的障碍物。

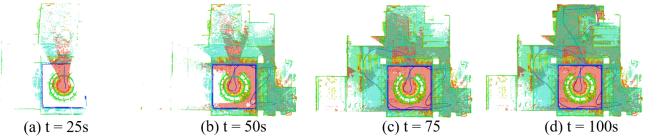


Fig. 13: 基于室内-2 地图的无人机自主探索仿真演示，使用 FUEL 算法。展示了无人机在不同执行时间后扫描的区域。

障碍物及多无人机仿真。这些特性能够满足单机或多机无人机运动规划及自主探索算法的研发需求。

由于仿真器基于点云地图，当地图精度不足或存在噪声点时，仿真器无法还原真实环境的正确细节。此外，当 LiDAR 扫描较稀疏时，仿真器在计算效率上的优势可能减弱，因为 Gazebo (及其他现有仿真器) 基于网格模型的射线投射时间将显著减少。相比之下，我们的渲染模块需生成扫描模式外的稠密深度图，导致计算不必要深度像素浪费计算资源。未来可从该方向对仿真器进行改进。

## References

- [1] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter, “Graph-based subterranean exploration path planning using aerial and legged robots,” *Journal of Field Robotics*, vol. 37, no. 8, pp. 1363–1388, Dec. 2020.
- [2] K. Shah, G. Ballard, A. Schmidt, and M. Schwager, “Multidrone aerial surveys of penguin colonies in antarctica,” *Science Robotics*, vol. 5, no. 47, p. eabc3000, 2020.
- [3] Y. Vacanas, K. Themistocleous, A. Agapiou, and D. Hadjimitsis, “Building information modelling (bim) and unmanned aerial vehicle (uav) technologies in infrastructure construction project management and delay

- and disruption analysis,” in *Third International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2015)*, vol. 9535. SPIE, 2015, pp. 93–103.
- [4] Y. Ren, F. Zhu, W. Liu, Z. Wang, Y. Lin, F. Gao, and F. Zhang, “Bubble planner: Planning high-speed smooth quadrotor trajectories using receding corridors,” *arXiv preprint arXiv:2202.12177 (accepted by 2022 IROS)*, 2022.
- [5] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.
- [6] O. Michel, “Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, Mar. 2004, publisher: SAGE Publications.
- [7] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and service robotics*. Springer, 2018, pp. 621–635.
- [8] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.
- [9] F. Şenkul and E. Altuğ, “Modeling and control of a novel tilt-roll rotor quadrotor uav,” in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2013, pp. 1071–1076.
- [10] L. Ximin, H. Gu, J. Zhou, Z. Li, S. Shen, and F. Zhang, “Simulation and flight experiments of a quadrotor tail-sitter vertical take-off and landing unmanned aerial vehicle with wide flight envelope,” *International Journal of Micro Air Vehicles*, vol. 10, pp. 303–317, 12 2018.
- [11] W. Luo, Q. Tang, C. Fu, and P. Eberhard, “Deep-sarsa based multi-uav path planning and obstacle avoidance in a dynamic environment,” in *International Conference on Swarm Intelligence*. Springer, 2018, pp. 102–111.
- [12] Z. Han, Z. Wang, N. Pan, Y. Lin, C. Xu, and F. Gao, “Fast-racing: An open-source strong baseline for SE(3) planning in autonomous drone racing,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8631–8638, 2021.
- [13] G. Rong, B. H. Shin, H. Tabatabaei, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim, “LGSVL Simulator: A High Fidelity Simulator for Autonomous Driving,” *arXiv:2005.03778 [cs, eess]*, June 2020, arXiv: 2005.03778.
- [14] M. Dharmadhikari, T. Dang, L. Solanka, J. Loje, H. Nguyen, N. Khedekar, and K. Alexis, “Motion primitives-based path planning for fast and agile exploration using aerial robots,” 05 2020, pp. 179–185.
- [15] C. Cao, H. Zhu, H. Choset, and J. Zhang, “TARE: A Hierarchical Framework for Efficiently Exploring Complex 3D Environments,” in *Robotics: Science and Systems XVII*. Robotics: Science and Systems Foundation, July 2021.
- [16] A. Brunel, A. Bourki, C. Demonceaux, and O. Strauss, “Splatplanner: Efficient autonomous exploration via permutohedral frontier filtering,” in *IEEE International Conference on Robotics and Automation (ICRA 2021)*, 2021.
- [17] A. Brunel, A. Bourki, O. Strauss, and C. Demonceaux, “FLYBO: A Unified Benchmark Environment for Autonomous Flying Robots,” in *2021 International Conference on 3D Vision (3DV)*. London, United Kingdom: IEEE, Dec. 2021, pp. 1420–1431.
- [18] N. Koenig, “Darpa subt virtual competition software,” 2019. [Online]. Available: [https://github.com/osrf/subt/tree/master/subt\\_ign/worlds](https://github.com/osrf/subt/tree/master/subt_ign/worlds)
- [19] S. Manivasagam, S. Wang, K. Wong, W. Zeng, M. Sazanovich, S. Tan, B. Yang, W.-C. Ma, and R. Urtasun, “Lidarsim: Realistic lidar simulation by leveraging the real world,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 167–11 176.
- [20] H. Pfister, M. Zwicker, J. Baar, and M. Gross, “Surfels: Surface elements as rendering primitives,” *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 05 2000.
- [21] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena, “C-blox: A scalable and consistent tsdf-based dense mapping approach,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [22] P. Labatut, J.-P. Pons, and R. Keriven, “Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts,” in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.
- [23] J. Lin and F. Zhang, “R3live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.
- [24] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, “Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6941–6948.
- [25] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “Fast-lio2: Fast direct lidar-inertial odometry,” *IEEE Transactions on Robotics*, pp. 1–21, 2022.
- [26] X. Liu, Z. Liu, F. Kong, and F. Zhang, “Large-scale lidar consistent mapping using hierarchical lidar bundle adjustment,” *arXiv preprint arXiv:2209.11939*, 2022.
- [27] B. Zhou, Y. Zhang, X. Chen, and S. Shen, “Fuel: Fast uav exploration using incremental frontier structure and hierarchical planning,” *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–1, 01 2021.
- [28] D. Shreiner, B. T. K. O. A. W. Group, et al., *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. Pearson Education, 2009.
- [29] T. Lee, M. Leok, and N. H. McClamroch, “Geometric tracking control of a quadrotor uav on  $se(3)$ ,” in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.
- [30] R. Mahony, V. Kumar, and P. Corke, “Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [31] K. Ogata et al., *Modern control engineering*. Prentice hall Upper Saddle River, NJ, 2010, vol. 5.
- [32] C. Zheng, Q. Zhu, W. Xu, X. Liu, Q. Guo, and F. Zhang, “Fast-livo: Fast and tightly-coupled sparse-direct lidar-inertial-visual odometry,” *ArXiv*, vol. abs/2203.00893, 2022.
- [33] F. Zhu, Y. Ren, F. Kong, H. Wu, S. Liang, N. Chen, W. Xu, and F. Zhang, “Decentralized lidar-inertial swarm odometry,” *arXiv preprint arXiv:2209.06628*, 2022.
- [34] Y. Ren, S. Liang, F. Zhu, G. Lu, and F. Zhang, “Online whole-body motion planning for quadrotor using multi-resolution search,” *arXiv preprint arXiv:2209.06761*, 2022.