MARSIM: A light-weight point-realistic simulator for LiDAR-based UAVs

Fanze Kong[1], Xiyuan Liu[1], Benxu Tang[2], Jiarong Lin[1], Yunfan Ren[1],
Yixi Cai[1], Fangcheng Zhu[1], Nan Chen[1], Fu Zhang[1]

***Abstract*— Low-cost, small-form factor LiDAR sensors have enabled new opportunities for autonomous UAVs, but their development relies on extensive simulations. Current simulators are hard to simulate real-world environments due to the need for dense mesh maps. We present a simulator that can generate realistic point cloud scanning from point cloud maps for LiDAR-based UAVs. Our lightweight simulator supports various LiDAR types, dynamic obstacles, and multi-UAV systems in the ROS framework. The simulator provides 10 high-resolution point cloud maps of diverse real-world environments, facilitating diverse testing scenarios. Evaluation results show superior performance in time and memory consumption compared to Gazebo, with simulated UAV flights closely matching real-world performance. This point-realistic, lightweight simulator aims to bridge the gap between UAV simulation and experiments, fostering future research on LiDAR-based autonomous UAVs.**

I. INTRODUCTION

Recent developments of LiDAR technologies have significantly lowered the cost and weight of LiDAR sensors, which creates many opportunities for unmanned aerial vehicle (UAV) applications, such as mine exploration [1], biological data statistics [2], mapping [3], high-speed navigation [4], and obstacle avoidance, etc. However, deploying UAVs to these widespread applications requires extensive tests, which are often cost-demanding since the system under test are still in active development and hence may have a noticeable failure rate (e.g., collision with the environment). A simulator that resembles the reality can significantly reduce the time and equipment cost occurred in UAV tests and has become a crucial component of UAV developments.

Existing simulators (e.g., Gazebo [5], Webots [6], Airsim [7]) have difficulties meeting the demand of high-resolution realistic scene simulation for LiDAR-based UAVs due to the following limitations: (i) their simulated environments are mostly virtual, unrealistically simple, and man-made, which possess a considerable gap from complex real-world scenes; (ii) they only import mesh maps, which are difficult to obtain from real-world environments that are often measured in 3D point clouds by laser scanners or LiDARs. To the best of our knowledge, there are no open-source and mature tools available for generating high-resolution and high-fidelity mesh maps out of point cloud data. The commonly-used Poisson

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

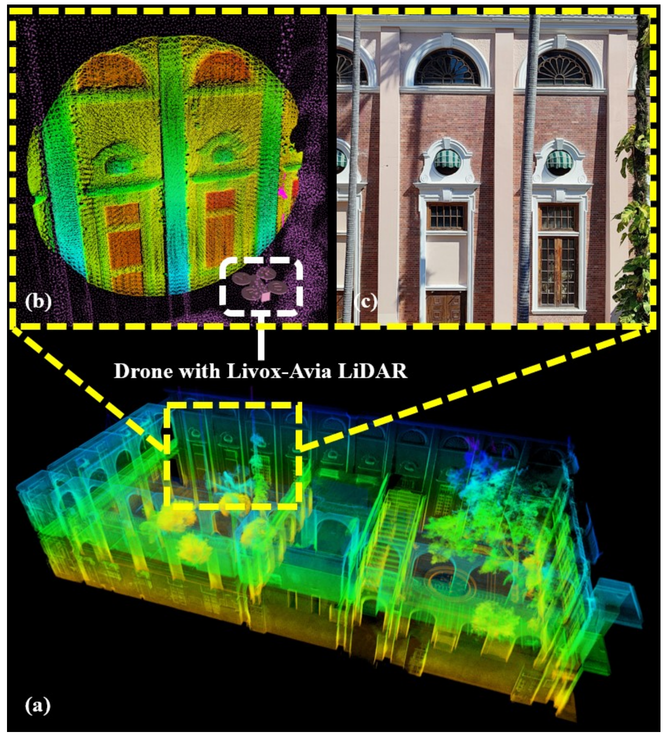

Fig. 1. A demo of MARSIM. (a) The point cloud map of the HKU main building (one of ten real-world scenes of MARSIM). (b) A scan of points of a Livox Avia LiDAR rendered directly from the point cloud map by MARSIM. (c) The photo of the corresponding scene in the real world. It is seen our proposed simulator can restore the structural details of the real scene with high quality (more details can be found on https://youtu.be/dVUi9jQled0.)

reconstruction [8] method is time-consuming and has low-quality meshes on real point cloud data captured by LiDARs due to occlusions and point density variations in large scene scanning; (iii) they often rely on high-performance GPUs to achieve real-time simulations in large complex mesh maps, which puts a high requirement for computing platforms.

Motivated by these gaps, in this paper, we propose a light-weight LiDAR-based UAV simulator, which has the following features:

1) Directly utilizing point cloud maps reconstructed from real environments for LiDAR scan rendering. The point cloud map contains fine details of the environments and could be easily obtained with a LiDAR.
2) High efficiency in computation and memory consumption, and the ability to run on personal computers without a dedicated graphics processing unit (GPU).
3) Versatility in supporting the simulation of three types of dynamic obstacles, multi-UAV systems (with con-

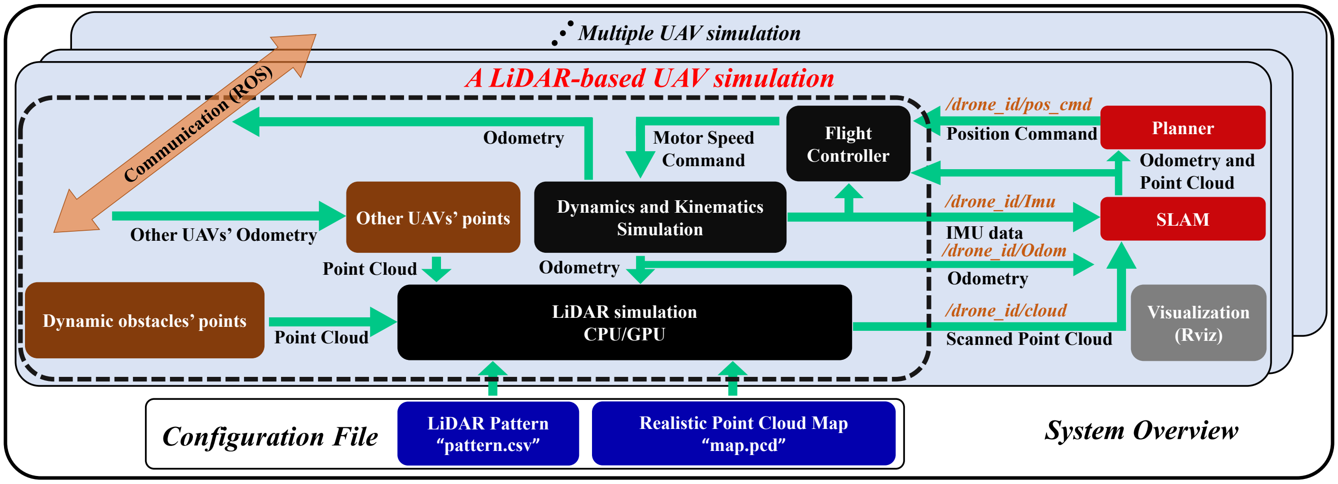

Fig. 2. The overall framework of our simulator (black dashed box) and how it interacts with external modules in ROS.

figurable flight control modules), and various solid-state and mechanical spinning LiDAR models with adjustable parameters (angular resolutions, scanning patterns, field-of-views, sensing ranges, etc.).

4) We open-source our code on GitHub[1], which is ROS-compatible and can be easily integrated with SLAM and path-planning algorithms to benefit the community.

II. System Overview

As shown in Fig. 2, our UAV simulator is mainly composed of three submodules: a built-in flight controller module, a dynamics and kinematics simulation module, and a LiDAR simulation module (modules in black, see Fig. 2). The simulator is able to interact with planners, SLAM algorithms, and visualization modules in the ROS framework, forming a complete LiDAR-based UAV simulation system.

To use the simulator, users should first choose a LiDAR model and supply a point cloud map of the environment. Users can then plug in their own SLAM (or use ground-truth odometry) and Planner algorithms to the UAV simulator via the ROS topic names shown in Fig. 2 for verification and visualization. Once the simulator starts, the dynamics and kinematics simulation module starts to compute the UAV's odometry and IMU data, according to which the LiDAR simulation module then renders the LiDAR scanned point cloud. The simulated IMU data and LiDAR scans are published in ROS, which could be used by the SLAM and then by the planner module. Besides the static environments represented by the point cloud map, the LiDAR simulation also simulates point measurements on dynamic obstacles and other UAVs in real time.

III. Results

A. High-resolution realistic Point Cloud Maps

This paper provides high-resolution (0.01 m) point cloud maps of ten real scenes for users to simulate, as shown in Fig. 3. The 0.01-m resolution map here refers to the original point cloud processed by the 0.01-m spatial downsampling. The scenes of the ten maps are three forests, three indoor scenes, a historical building (the HKU main building), two parking garages, and a large office.

[1]https://github.com/hku-mars/MARSIM.git

B. Breakdown of computation resources consumption

We compare the time consumption between MARSIM simulator and the Gazebo simulator. Since Gazebo can only use mesh models, we transform the point cloud maps of respective resolutions (see below) to mesh models using Poisson reconstruction method [8]. We select five typical scenes and compare the time and memory consumption of rendering one scan of a Livox AVIA LiDAR ($77^\circ \times 70^\circ$ FoV, 385×350 resolution, 30-m sensing range), respectively. For each map, we test two cases: a high-resolution map (0.05-m resolution) and a low-resolution map (0.2-m resolution). The data is generated by randomly selecting 10 positions and yaw angles of the UAV. The running time comparison on a light-weight computing platform NUC 10 Kit (with an i7-10710U max frequency 4.70-GHz CPU, 32-GB RAM) is shown in Fig. 4. It can be seen that in low-resolution maps, even the CPU version of MARSIM can achieve slightly less computation time than the GPU-accelerated Gazebo simulation. With GPU acceleration, MARSIM is two times faster than Gazebo. In high-resolution maps, the difference is even more obvious: the CPU version of MARSIM is two times faster than the GPU-accelerated Gazebo simulation while the GPU version of MARSIM is ten times faster. The reason why Gazebo performed poorly in the experiments is because of the large number (over 2 million) of triangular faces in the generated mesh maps, which is necessary to retain a level of detail similar to the corresponding point cloud. In contrast, most existing robot simulations use very simple mesh maps, which are mainly composed of large planes and have a small number of triangular faces, which can be simulated in real time. Moreover, as a simulator specifically designed for point cloud, MARSIM does not need to process the whole render pipeline to render meshes (e.g., reducing the process of fragment shader, ray tracing, etc.) and complex

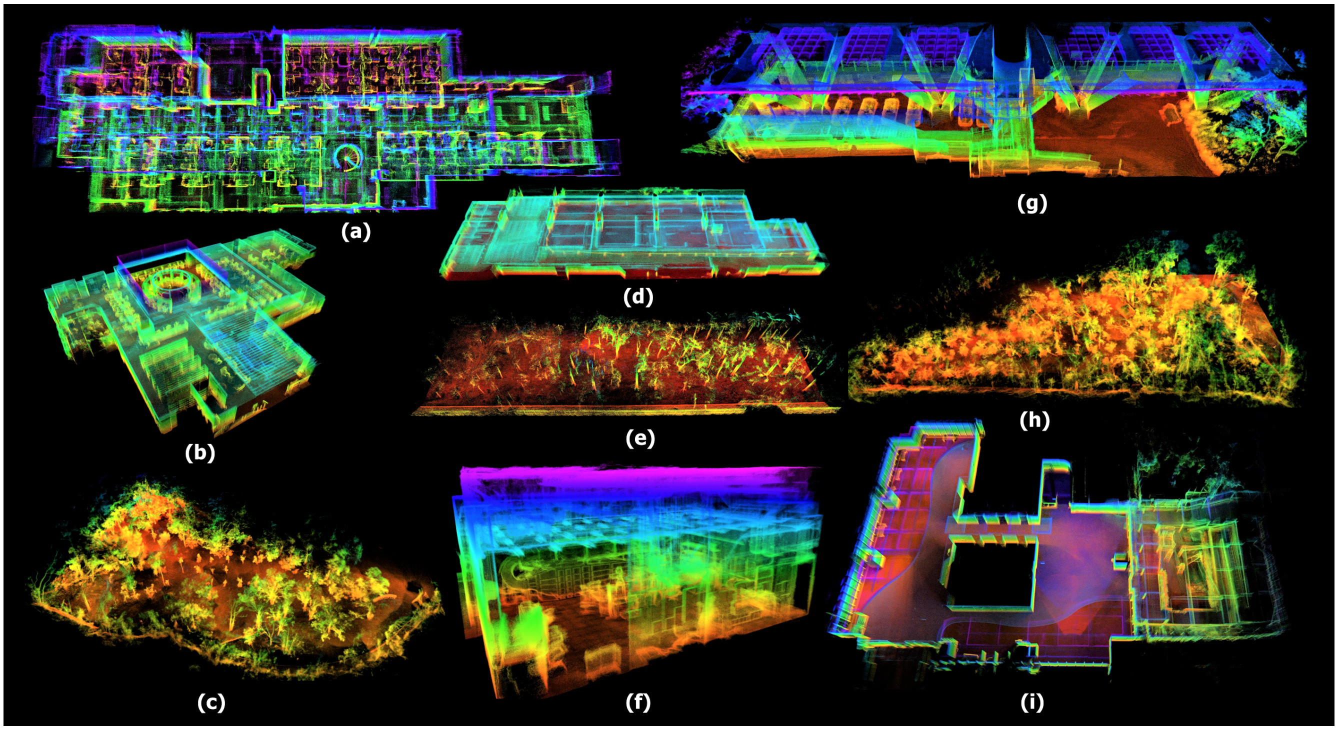

Fig. 3. Ten high-resolution point cloud maps provided by the simulator.

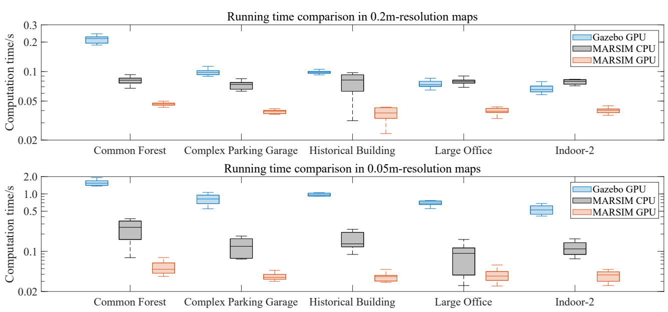

Fig. 4. Time consumption for rendering one Livox AVIA scan on a light-weight computation platform (NUC).

physics simulation (like collision simulation), which decrease the consumption of computation resources significantly.

In addition to the time consumption comparison, we also collected the RAM consumption as shown in Table I. The RAM consumption of our simulator is about half that of the Gazebo simulator in both the CPU and GPU versions, which also demonstrates the light-weight characteristics of our simulator.

C. Support of different types of LiDARs and other functions

In order to increase the simulator's versatility, a variety of common LiDAR and depth camera models are also provided in the simulator. As shown in Figure 5, the simulator supports sensors such as Livox Avia, Livox Mid-360, VLP-32, VLP-64, OS1-32, and Intel realsense D455. The simulator can reproduce the scanning patterns of these sensors so that users can use them directly without tuning any parameters.

TABLE I

MEMORY CONSUMPTION COMPARISON WITH GAZEBO ON A LIGHT-WEIGHT COMPUTATION PLATFORM (NUC).

Resolution	Map	RAM Consumption (GB)		
		Gazebo	MARSIM	
			CPU	GPU
0.2 m	Historical building	1.64	1.01	1.31
	Complex Parking Garage	1.46	1.1	1.25
	Large Office	1.68	1.04	1.13
	Common Forest	1.48	1.37	1.73
	Indoor-2	2.09	0.94	1.14
0.05 m	Historical building	6.97	4.04	3.43
	Complex Parking Garage	6.72	3.51	3.06
	Large Office	4.93	2.87	2.17
	Common Forest	16.88	7.35	3.53
	Indoor-2	3.73	2.51	1.84

D. Practical applications of the simulator

This simulator is mainly used to provide a testing and verification platform for the algorithm development of LiDAR-based UAVs, especially motion planning and autonomous exploration algorithms that require interaction with the environments. While previous experiments have shown the application of our simulator in UAV motion planning, we also carried out simulation experiments of autonomous UAV exploration. Fig. 8 shows the autonomous exploration process of a UAV carrying Livox Avia using FUEL [9] algorithm in the indoor-2 map. It is worth mentioning that the simulator has been successfully used to assist the development of multi-UAV mutual location in [10] and motion planning algorithm in [4], [11].

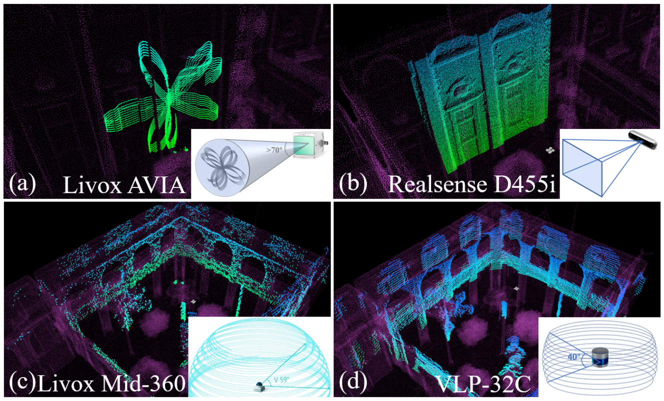

Fig. 5. Various LiDAR scan pattern support including Livox Avia (a), D455 (b), Livox Mid-360 (c), and VLP 32 (d), respectively.

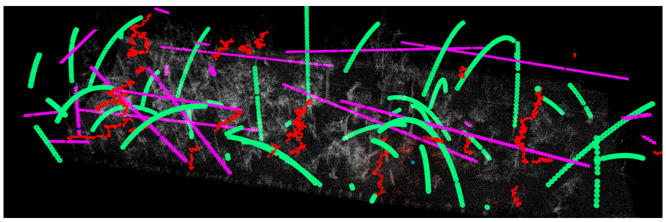

Fig. 6. Dynamic obstacles simulation in the simple forest map. The purple lines represent UAV models moving in constant speeds, the green curves represent spherical obstacles moving in free falling trajectories, and the red points represent cubes moving in random walk.

IV. Conclusion

This paper proposes a LiDAR-based UAV simulator for real environment simulation on light-weight computing platforms. The simulator renders LiDAR scans directly on point cloud maps, which is way easier to capture for real environments than mesh models used by existing simulators. Moreover, due to the high accuracy of modern 3D LiDARs and laser scanners, a point cloud map scanned from real environments can truthfully represent the environment, which dramatically bridges the gap between simulation and reality. To maximize the practicality of the simulator, we further provide ten high-resolution point cloud maps and support the simulation of various types of LiDAR sensors, dynamic obstacles, and multi-UAV simulation. These features can meet the research and development needs of motion planning algorithms and autonomous exploration algorithms of single or multiple UAVs.

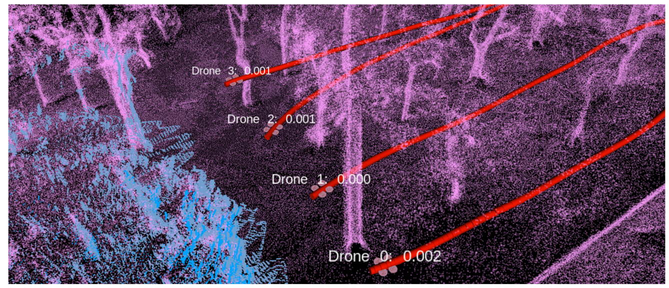

Fig. 7. Multi-UAV planning simulation. The pink models are the UAVs, and the red curves are the trajectories of the UAVs, avoiding the obstacles of a realistic forest map.

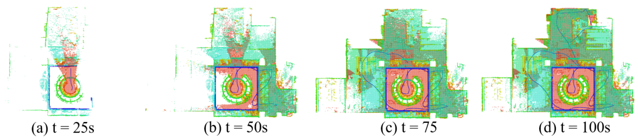

Fig. 8. Demonstration of a UAV autonomous exploration simulation in indoor-2 map, utilizing FUEL algorithm. The area scanned by the UAV after different executing times are shown.

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] E. Karachaliou, E. Georgiou, D. Psaltis, and E. Stylianidis, "Uav for mapping historic buildings: From 3d modelling to bim," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 397–402, 2019.

[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," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 6332–6339.

[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] 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.

[10] 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.

[11] 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.