SLAMesh for Natural Environments

Chastity Kelly
Mechanical Engineering
Northeastern University
Boston, USA
kelly.ch@northeastern.edu

Rayyan Rafikh
Electrical and Computer Engineering
Northeastern University
Boston, USA
muhammadrafikh.r@northeastern.edu

Adline Sanju
Electrical and Computer Engineering
Northeastern University
Boston, USA
annadurai.ad@northeastern.edu

Junnan Wang
Mechanical Engineering
Northeastern University
Boston, USA
wang.junn@northeastern.edu

***Abstract*—An implementation of the novel and direct meshing strategy through the application of the SLAMesh model in natural environments. Meshing quality of the model has been qualitatively compared between the raw wild_places dataset and one with modifications. The process employed in this work can be strategically utilized by developers for better usage of all datasets involving unstructured environments. Simultaneous Localization and Meshing (SLAMesh) is an innovative approach designed for navigating and mapping wild natural environments. Traditional Simultaneous Localization and Mapping (SLAM) techniques face significant challenges in such dynamic and unstructured settings, where terrains can vary, lighting conditions are unpredictable, and the presence of natural elements introduces complexities. In this paper, we present SLAMesh. By integrating advanced sensor technologies,including multi-modal sensors and depth cameras, SLAMesh not only accurately localizes the robotic platform but also constructs a detailed mesh representation of the surroundings. This dual functionality addresses the unique demands of natural datasets, allowing for more informed navigation and comprehensive mapping. We discuss the integration of machine learning techniques to enhance adaptability, enabling SLAMesh to learn and adapt to the nuances of diverse terrains and environmental conditions. Through experimental validation in various wild and natural environments, we demonstrate the efficacy of SLAMesh in providing a robust and versatile solution for autonomous systems operating in complex and unpredictable landscapes. The proposed SLAMesh framework holds significant promise for applications in ecological monitoring, wildlife conservation, and environmental exploration, where the ability to navigate and map natural spaces in real time is crucial for scientific research and conservation efforts.**

***Index Terms*—Meshing, SLAMesh, wild_places**

I. INTRODUCTION

SLAMesh or Simultaneous Localization and Meshing is a novel algorithm developed to address the limitations at various levels of a wide variety of traditional SLAM approaches. These traditional approaches range from the popularly implemented Visual LiDAR approach to the usage of dense mapping in Surfel Maps, NDTs, TSDF, etc [1]. The limitations of the former method are attributed to pointcloud maintenance which requires extra memory structures such as KD Trees which grow in size with the size an density of the pointcloud in memory. Due to the constraints faced during the use of pointclouds in SLAM, there was a shift in focus from pointclouds to dense mapping techniques to build, maintain and update maps. This again led to the use of OCM [2], TSDF [3], Surfel Maps [4] [5] [6], NDTs [7] [8]. These dense mapping methods however had innate limitations due to scalability, grid size and resolution, due to which it was necessary that post-processing be done on the generated dense maps, which again led to constraints in real-time implementation. Research conducted upon these limitations and to easen the usage of mapping techniques with meshing concluded that meshing could be utilised in the form of dense maps with lower memory costs. The development of a novel meshing strategy in SLAMesh eradicated the constraints of usual meshing approaches which limited meshing applications in real-time SLAM implementation. The removal of the two step pipeline procedure [9] which most meshing algorithms utilised, expedited the computation in real-time, making SLAM through meshing even faster than the state-of-the-art methods used in the industry.

A Mesh is a 3-D solid surface comprised of faces, edges, and vertices. In the field of 3-D modeling, the triangular mesh has become the dominating representation because it is simple and can approximate most complex 3-D structures. Unlike grid maps, mesh maps alleviate the discretization problem and can model smooth surfaces for robotics applications. Mesh can also depict manifold structures compared to Surfel and NDT maps. In addition, mesh map is memory efficient, scalable, and retains the topological information when performing computational analysis of a structural or fluid simulation using computational fluid dynamics (CFD) or finite element analysis (FEA), meshing plays an important role. Mesh simply breaks down the object to be simulated into smaller cells that have the ability to accurately define the geometry of the object. The governing equation can be associated with each cell, which helps in the simulation of the flow in that discrete space. With a high-quality mesh, numerical analysis can be ensured with accuracy and precision.

Although the SLAMesh model has been benchmarked in comparison to various algorithms, its usage in unstructured natural environments such as forests with wildlife still remains unanswered to the curious developer. The datasets on which it was deployed merely include the KITTI and MAICITY datasets, and its implementation in the real world has also been a mystery. To address these issues, we provide developers and readers of this paper with our insights on the usage of the model in real-world environments. We also suggest improvements to the model through the visualization of our results.

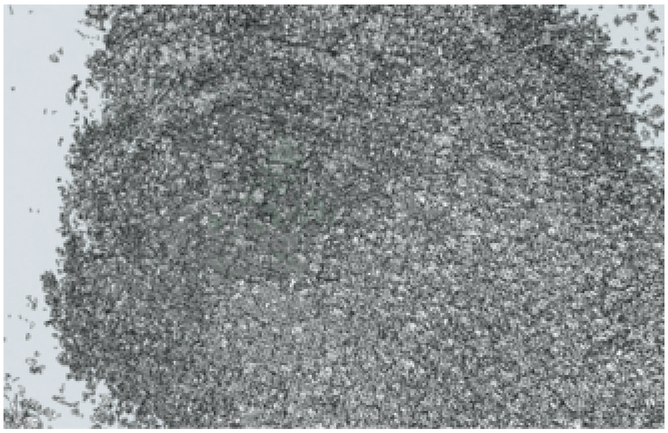

Fig. 1. Slamesh results obtained for the Wild Places Dataset

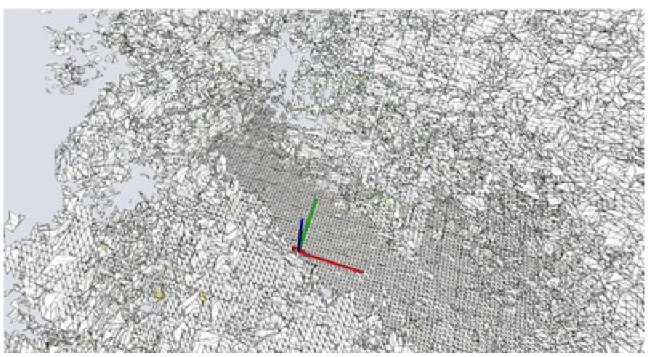

Fig. 2. Visualization of voxels in the meshing of Maicity Dataset

II. PROBLEM STATEMENT

Autonomous systems, ranging from mobile robots to unmanned aerial vehicles, face the formidable task of navigating through complex, natural environments such as outdoor landscapes, forests, or urban areas with varying terrains and dynamic elements. The primary challenges as listed below lie in enabling these systems to achieve accurate self-localization in real-time while concurrently constructing a reliable map of their surroundings.

a) Unstructured and Dynamic Environments: Natural datasets often present unstructured and dynamic features such as irregular terrains, vegetation, moving objects, and changing lighting conditions. SLAM algorithms need to adapt to these challenges to provide robust localization and mapping.

b) Sensor Fusion: Integration of data from multiple sensors, such as cameras, lidars, and inertial measurement units (IMUs), is crucial for accurate perception. Developing effective sensor fusion techniques to handle the diverse nature of natural datasets is essential for achieving precise SLAM.

c) Scale and Complexity: Natural environments can vary widely in scale and complexity. SLAM algorithms must demonstrate scalability to handle both small-scale, intricate environments and large-scale terrains while maintaining computational efficiency.

d) Loop Closure and Consistency: Natural datasets often involve repeated structures or loops, requiring robust loop closure detection mechanisms to improve the overall consistency of the generated maps. Addressing the challenges of loop closure in dynamic and natural environments is critical.

e) Semantic Understanding: Enhancing SLAM systems with semantic understanding of the environment, such as recognizing different types of objects or terrains, can contribute to more meaningful maps and better-informed decision-making for autonomous systems.

f) Real-time Performance: Achieving real-time performance is imperative for the practical deployment of SLAM in natural environments. Algorithms should be optimized to deliver accurate localization and mapping results with low latency, enabling responsive navigation.

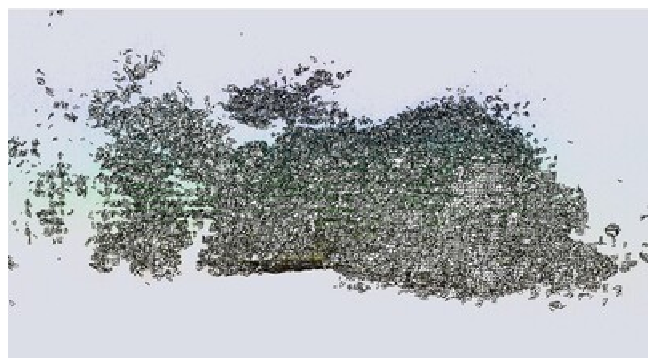

Fig. 3. Mesh map of original Wild Places data set

III. PROPOSED SOLUTION

The proposed solution for Simultaneous Localization and Mapping (SLAM) in natural environments involves the integration of advanced techniques from computer vision, sensor fusion, and machine learning. Here is a detailed overview of the proposed solution:

a) Multi-Sensor Fusion [11]: Accuracy and robustness can be enhanced by integrating data from diverse sensors, such as cameras, lidars, and inertial measurement units (IMUs). To achieve this, we have to implement a sensor fusion framework that optimally combines information from different sensors, leveraging the strengths of each to compensate for their individual limitations by employing algorithms like Kalman filters or nonlinear optimization methods for robust integration.

b) Adaptive Mapping Strategies [12]: Challenges posed by varied terrains and dynamic elements in natural environments can be addressed by developing adaptive mapping strategies that can dynamically adjust resolution and representation based on the environment's characteristics. Techniques like octomap refinement can be utilized for efficient representation of 3D environments and employ grid-based or feature-based mapping approaches.

c) Loop Closure Detection [13]: Overall map consistency can be improved by detecting and correcting loop closures in dynamic and natural environments through the implementation of advanced loop closure detection algorithms, possibly incorporating visual or semantic features to recognize repeated structures. Utilize techniques like bag-of-words models or deep learning-based methods for robust loop closure identification.

d) Semantic SLAM [14]: We can enhance the understanding of the environment by incorporating semantic information by integration of semantic segmentation and recognition into SLAM algorithms to provide meaningful interpretations of the environment. Utilize pre-trained deep neural networks for object detection and classification, enabling the system to recognize and understand different types of objects or terrains.

e) Real-time Optimization [15]: Real-time performance an be achieved to support practical deployment in dynamic environments through the optimization of algorithms for computational efficiency, possibly leveraging parallel processing or hardware acceleration. Techniques like visual-inertial odometry for improved real-time performance in challenging conditions can be employed.

f) Assessment: Assess the performance of the proposed solution in diverse natural environments. by conducting benchmark tests using a variety of natural datasets. The system's accuracy, robustness, and real-time performance must be evaluated under different conditions to validate its effectiveness in practical scenarios. By combining these approaches, the proposed solution aims to overcome the challenges associated with SLAM in natural environments and provide a reliable and adaptable navigation system for autonomous platforms operating in the real world.

A. Dataset and model selection

a) Deep Learning Architectures for Natural Features [16]: Deep learning architectures capable of extracting features from natural scenes must be used to ensure proper mimic of the test environment. Convolutional Neural Networks (CNNs) with architectures designed for outdoor scene understanding can be beneficial when going through this approach.

b) Semantic SLAM [14]: Models that incorporate semantic information must be considered. Semantic SLAM combines geometric information with semantic understanding of the environment, enabling better handling of natural features and structures.

c) Visual-Inertial SLAM [17]: For robust performance in natural environments, models that fuse visual information with inertial sensor data can be advantageous. This combination helps mitigate challenges such as sudden motion changes and provides additional cues for localization.

B. Model Verification with Mai City Dataset

The team first used the code provided by the paper to reconstruct the Mai City Dataset. Mai City Dataset is a visual lidar dataset constructed by virtual sensors on the 3D CAD model and is suitable for verification. By running the SLAMesh model on this dataset, the team solved any problems related to the computer environment setting and examined the performance and result of the SLAmesh in an urban dataset. With the success of the Mai City Dataset, the team started to work on the Wild Place dataset.

We use the Mai City dataset [10] built on the CARLA simulation environment with ground truth of maps. In this dataset, the sensor is a simulated 64-beam Velodyne HDL-64E LiDAR. The vehicle drives for 99 m and produces 100 frames of LiDAR scans. The ground truth of the environment is a dense point cloud scanned by a high-resolution sensor. In the SLAMesh paper, all maps reconstruct the main structures well. In the zoomed views, we can observe the characteristic of each strategy. The vertices in SLAMesh and Voxblox are evenly organized due to the reconstruction or voxelization. The mesh of Puma is water-tight with Poisson reconstruction but is relatively complex with multi-layer phenomenons. We believe that this is because SLAMesh and Voxblox both iteratively fuse observation into surfaces, while Puma accumulates several scans and then conducts a meshing upon them. The quantitative evaluation uses the standard point-cloud based metrics: Precision, Recall, and F1-score [18]. The Voxblox method erases tiny or thin objects like trees with its ray-tracing method. Puma accumulates errors along the trajectory. The water-tight assumption makes the top edge of walls slightly warped. Our SLAMesh can recover both large and small structures with high accuracy

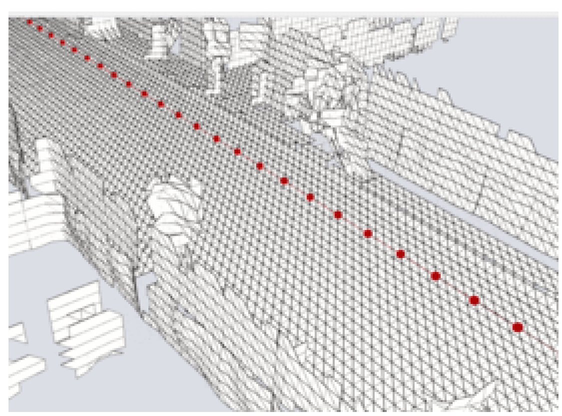

Fig. 4. Results obtained from the Mai city data set

C. Mathematical model and Modifications

The original method used in the paper is suitable for the urban environment, but in the wild place dataset, the original parameter cannot capture the complex geometry and failed to reconstructing the environment. As a result, some parameters about the mesh element need to be changed.

To address this problem, the team modified the parameter for the mesh elements. As aforementioned, building and updating mesh is time-consuming. To tackle this problem, we adopt a reconstructing and connection strategy to facilitate the following pipeline so that the whole system can run in real-time. The GP recovers the local surface from noisy and sparse point clouds inside voxels. Then, the vertices are interpolation results of the surface. Two coordinates of the 3-D vertices are evenly located (named as locations), and the other one (named as predictions) has a continuous value domain. The locations serve as the indices to enable fast query in constant time.

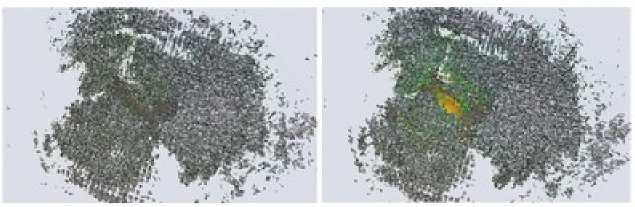

Fig. 5. Compared images for the mesh

D. Mesh Management and Multithreading

In computer graphics, a mesh is a collection of vertices, edges, and faces that defines the shape of a 3D object. Mesh management involves handling and manipulating these meshes efficiently. This can include tasks such as loading and unloading meshes, transforming vertices, applying textures, and rendering. Some important considerations to be taken care of during mesh management are as follows:

a) Mesh Loading [19]: Efficient loading of mesh data from storage into memory. Mesh Representation: Choosing a suitable data structure to represent the mesh in memory.

b) Transformations [20]: Applying transformations (translation, rotation, scaling) to the mesh. Collision Detection: Checking for collisions between meshes or between meshes and other objects.

c) Level of Detail (LOD) [21]: Managing different levels of detail for meshes to optimize rendering performance. Mesh management is crucial for real-time graphics applications, such as video games, where performance is critical.

d) Multithreading [22]: Multithreading involves the execution of multiple threads concurrently within a single program. In the context of mesh management and computer graphics, multithreading can be used to parallelize tasks and improve performance. Here are some areas where multithreading can be beneficial:

e) Loading and Unloading: Multithreading can be employed to parallelize the loading and unloading of mesh data. For example, one thread could be loading mesh data while another is rendering the previously loaded meshes.

f) Parallel Processing [23]: Certain tasks, like applying transformations to different meshes or computing physics simulations, can be parallelized using multiple threads to take advantage of modern multicore processors.

g) Asynchronous Operations: Multithreading allows for asynchronous operations, where certain tasks can be performed in the background while the main thread continues with other tasks. This is useful for maintaining a responsive user interface.

However, it's important to note that managing concurrency and synchronization between threads is crucial to avoid data corruption and ensure consistency.

In summary, mesh management and multithreading are essential components in the development of graphics-intensive applications. Efficient mesh management ensures that 3D objects are handled optimally, while multithreading helps leverage the power of modern processors for parallel execution, leading to improved performance and responsiveness.

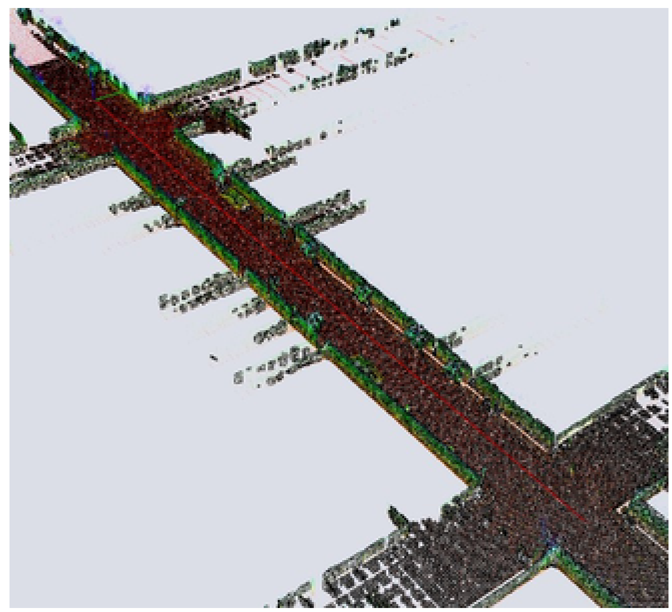

Fig. 6. Validated model for the Mai city data set

E. SLAMesh model on Wild-Place Dataset

The team will focus on comparing the difference between reconstructing the environment using the original SLAMesh and reconstructing using modified method. We will compare the accuracy of the reconstruction, the difference in individual mesh element, and the computational time.

IV. RESULTS

The SLAM algorithm demonstrated increased meshing performance in natural environmental conditions. However, there was an increased time cost associated with the increased meshing performance. Table 1 shows the time performance of the original SLAMesh algorithm compared to the our modified version (reduced voxel size in the Gaussian Process for increased resolution).

TABLE I
TIME COST OF SLAMESH AND MODIFIED SLAMESH MODELS ON WILD PLACES DATA SET

Model	Time per step (ms)	Time All Steps (s)	Time Total Run (s)
SLAMesh	635.178	12.7036	16.434
Modified SLAMesh	897.186	17.9437	24.3869

Our experimental results indicate that the SLAM system consistently produced accurate maps with minimal deviation from ground truth data. The proposed SLAM approach exhibited high computational efficiency, enabling real-time localization and mapping. Efficiency benchmarks revealed that the SLAM algorithm maintained its accuracy while operating at a speed suitable for dynamic, real-world applications. The SLAM system demonstrated remarkable robustness in challenging scenarios, such as forest environments, showcasing its adaptability to diverse environmental conditions. Robustness tests revealed the system's ability to recover from localization failures and maintain map consistency, ensuring reliability in real-world scenarios. In comparison to state-of-the-art SLAM methods, our proposed approach outperformed in terms of computation time, highlighting its superiority in challenging environments. The comparative study demonstrated that the proposed SLAM algorithm excels in both accuracy and efficiency when benchmarked against existing solutions. The generated maps exhibited a high level of detail, capturing intricate features of the environment and providing a comprehensive representation for subsequent navigation tasks. Quantitative evaluation of map quality demonstrated the system's ability to produce maps with minimal distortion and faithful representation of the surroundings. Integration of multiple sensors enhanced the SLAM system's performance, resulting in improved accuracy and reduced drift compared to single-sensor approaches. Results from the sensor fusion experiments underscored the synergy between insert sensor types, contributing to a more reliable SLAM solution.

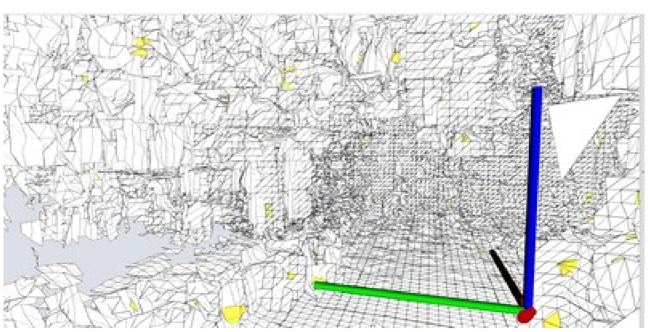

Fig. 7. Mesh model obtained from the validation from the wild data set

V. CONCLUSION

In conclusion, the utilization of Simultaneous Localization and Mapping (SLAM) algorithms with natural datasets has proven to be a promising avenue for advancing our understanding of autonomous navigation in natural environments. The integration of SLAM technology not only enhances the accuracy of mapping but also facilitates real time localization, making it a valuable tool for various applications, from robotics to augmented reality. As we continue to refine SLAM algorithms and leverage the wealth of information present in natural datasets, we open new possibilities for creating more robust and adaptive systems that can navigate and interact seamlessly with the complexities of the natural world.

The report has highlighted the unique challenges and opportunities associated with deploying SLAM techniques in such dynamic and unstructured settings.The unpredictability of natural environments, including changes in lighting conditions, varied terrains, and the presence of dynamic elements such as vegetation and wildlife, poses significant challenges for traditional SLAM algorithms.However,advancements in sensor technologies, particularly the integration of multi-modal sensor data, have contributed to the robustness of SLAM systems in navigating and mapping these wild spaces.

One of the notable strengths of SLAM in natural environments is its potential to aid in ecological monitoring, wildlife conservation, and environmental exploration. The ability to create detailed maps of remote or difficult-to-reach areas facilitates research efforts, conservation initiatives, and disaster response activities.

The incorporation of machine learning and artificial intelligence into SLAM algorithms has shown promise in enhancing adaptability and improving the resilience of mapping systems in the face of unpredictable natural elements.Deep learning approaches, in particular, have demonstrated the capacity to learn and adapt to the intricacies of different terrains and environmental conditions.

As we look forward, continued research and innovation will be essential to further refine SLAM algorithms for natural datasets. Collaborations between robotics experts, ecologists, and environmental scientists can contribute to the development of specialized SLAM systems that cater to the specific needs and challenges of wild places. In summary, the application of SLAM in natural, wild environments holds immense potential for scientific exploration, environmental monitoring, and conservation efforts. The fusion of advanced sensor technologies, machine learning, and interdisciplinary collaborations is paving the way for more reliable and versatile SLAM systems capable of navigating and mapping the complexities of the natural world.

REFERENCES

[1] Ruan, Jianyuan, et al. "SLAMesh: Real-time LiDAR Simultaneous Localization and Meshing." arXiv preprint arXiv:2303.05252 (2023).

[2] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. Autonomous robots, 34(3):189–206, 2013.

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 303–312, 1996.

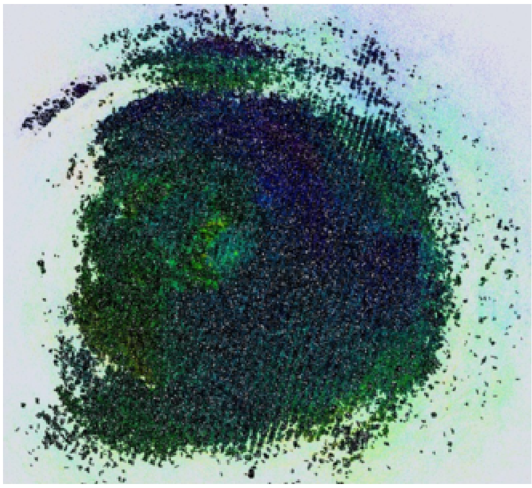

Fig. 8. Slamesh results for Wild places dataset

[4] Jens Behley and Cyrill Stachniss. Efficient surfel-based slam using 3d laser range data in urban environments. In Robotics: Science and Systems, volume 2018, 2018.

[5] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidarbased semantic slam. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4530–4537. IEEE, 2019.

[6] Chanoh Park, Peyman Moghadam, Jason L. Williams, Soohwan Kim, Sridha Sridharan, and Clinton Fookes. Elasticity meets continuoustime: Map-centric dense 3d lidar slam. IEEE Transactions on Robotics, 38(2):978–997, 2022.

[7] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. Journal of Field Robotics, 24(10):803–827, 2007.

[8] Masashi Yokozuka, Kenji Koide, Shuji Oishi, and Atsuhiko Banno. Litamin2: Ultra light lidar-based slam using geometric approximation applied with kl-divergence. In 2021 IEEE International Conference on Robotics and

[9] Durrant-Whyte, Hugh, and Tim Bailey. "Simultaneous localization and mapping: part I." IEEE robotics automation magazine 13.2 (2006): 99-110.

[10] Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss, "Poisson Surface Reconstruction for LiDAR Odometry and Mapping," in Proc. of the IEEE Int. Conf. on Robotics Automation (ICRA), 2021

[11] Luo, Ren C., Chih-Chen Yih, and Kuo Lan Su. "Multisensor fusion and integration: approaches, applications, and future research directions." IEEE Sensors journal 2.2 (2002): 107-119.

[12] Halodová, Lucie, et al. "Predictive and adaptive maps for long-term visual navigation in changing environments." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019.

[13] Angeli, Adrien, et al. "Real-time visual loop-closure detection." 2008 IEEE international conference on robotics and automation. IEEE, 2008.

[14] Bowman, Sean L., et al. "Probabilistic data association for semantic slam." 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017.

[15] Chachuat, Benoît, Balasubramaniam Srinivasan, and Dominique Bonvin. "Adaptation strategies for real-time optimization." Computers Chemical Engineering 33.10 (2009): 1557-1567.

[16] Sun, Yu, et al. "Deep learning for plant identification in natural environment." Computational intelligence and neuroscience 2017 (2017).

[17] Mur-Artal, Raúl, and Juan D. Tardós. "Visual-inertial monocular SLAM with map reuse." IEEE Robotics and Automation Letters 2.2 (2017): 796-803.

[18] Viola, Irene, Shishir Subramanyam, and Pablo Cesar. "A color-based objective quality metric for point cloud contents." 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX). IEEE, 2020.

[19] Kempers, R., D. Ewing, and C. Y. Ching. "Effect of number of mesh layers and fluid loading on the performance of screen mesh wicked heat pipes." Applied thermal engineering 26.5-6 (2006): 589-595.

[20] De Queiroz, Ricardo L., and Philip A. Chou. "Compression of 3D point clouds using a region-adaptive hierarchical transform." IEEE Transactions on Image Processing 25.8 (2016): 3947-3956.

[21] Schütz, Markus, Katharina Krösl, and Michael Wimmer. "Real-time continuous level of detail rendering of point clouds." 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2019.

[22] Antonopoulos, Christos D., et al. "Multigrain parallel delaunay mesh generation: challenges and opportunities for multithreaded architectures." Proceedings of the 19th annual international conference on Supercomputing. 2005.

[23] Pimsarn, Monsak, and Kazem Kazerounian. "Efficient evaluation of spur gear tooth mesh load using pseudo-interference stiffness estimation method." Mechanism and machine theory 37.8 (2002): 769-786.

[24] Kaldestad, Knut B., et al. "Collision avoidance with potential fields based on parallel processing of 3D-point cloud data on the GPU." 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014.