Graph-based Global Robot Simultaneous Localization and Mapping using Architectural Plans

Muhammad Shaheer[1], Jose Andres Millan-Romera[1], Hriday Bavle[1], Jose Luis Sanchez-Lopez[1],
Javier Civera[2] and Holger Voos[1]

Abstract— In this paper, we propose a solution for graph-based global robot simultaneous localization and mapping (SLAM) using architectural plans. Before the start of the robot operation, the previously available architectural plan of the building is converted into our proposed architectural graph (A-Graph). When the robot starts its operation, it uses its onboard LIDAR and odometry to carry out an online SLAM relying on our situational graph (S-Graph), which includes both, a representation of the environment with multiple levels of abstractions, such as walls or rooms and their relationships, as well as the robot poses with their associated keyframes. Our novel graph-to-graph matching method is used to relate the aforementioned S-Graph and A-Graph, which are aligned and merged, resulting in our novel informed Situational Graph (iS-Graph). Our iS-Graph not only provides graph-based global robot localization, but it extends the graph-based SLAM capabilities of the S-Graph by incorporating into it the prior knowledge of the environment existing in the architectural plan.

I. INTRODUCTION

The construction industry is increasingly using mobile robots, which offer numerous potential benefits. These robots can significantly reduce costs by regularly inspecting on-going construction sites and monitoring progress. However, most robots used in construction are either teleoperated or operate semi-autonomously, primarily due to the perception challenges associated with the ever-changing nature of construction sites. To enable fully autonomous operation, it would be advantageous for these robots to possess comprehensive prior knowledge of the construction site's geometry. By combining this prior knowledge with sensor readings during real-time operation, these robots could achieve robust and accurate global localization within construction sites.

Digital architectural plans, such as Building Information Modelling (BIM) [1], provide a means of capturing and communicating information about a construction site and incorporating it as prior knowledge about the scene. Works such as [2]–[4] have addressed the problem of extracting relevant structural knowledge from BIM and using it for real-time robot localization. However, these methods only extract geometric information from the BIM and do not leverage the topological and relational information also available in it, which limits the robustness and accuracy in complex and changing construction sites.

To tackle this problem, we present a novel approach to localize robots leveraging not only geometry but also higher-level hierarchical information from architectural plans. We present in this paper how to model the BIM information in the form of a graph that we denote as Architectural Graph (*A-Graph*), and then match and merge with the online Situational Graph (*S-Graph*) [5], [6] that the robot builds as it navigates the environment. As a key aspect, translating low-level geometry into high-level features in both graphs is what allows a robust matching between such different inputs.

Our method can be divided into three main stages. In the first one, an *A-Graph* is created for a given environment with nodes representing the semantic features available in a BIM model, specifically, wall-surfaces, doors, and rooms as the graph nodes, and edges containing the relevant relational information such as two wall surfaces comprising a wall, four wall surfaces connecting room and rooms connected through doors. In the second stage, running in real-time onboard the robot, a *S-Graph* is estimated using 3D LiDAR measurements. The nodes of our *S-Graph* correspond to semantics such as wall surfaces and rooms, and the edges correspond to constraints between these wall surfaces and the relevant room nodes. Finally, to localize the robot within this environment, a graph-matching algorithm is proposed utilizing hierarchical information from both graphs to provide the best match candidates finally resulting in informed (*iS-Graphs*) that fuses the information of both. This last graph will be the one used for global localization.

II. SYSTEM OVERVIEW

Fig. 1 shows an overview of the proposed approach. Firstly the BIM information is extracted for a given environment to create the two-layered **Architectural Graphs** (Section. III) in an offline manner, with all the elements extracted in the BIM frame of reference B. As the robot navigates within the given environment, an online **Situational Graph** (Section. II-A) is estimated by the robot in the map frame of reference M. In parallel, we run our the **Graph Matching** method (Section. IV) to provide match candidates between the existing *A-Graphs* and the current *S-Graph*. Finally, after retrieving the best match candidate, our **Graph Merging**

[1]Authors are with the Automation and Robotics Research Group, Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg (UL). Holger Voos is also associated with the Faculty of Science, Technology and Medicine, University of Luxembourg, Luxembourg. {muhammad.shaheer, jose.millan, hriday.bavle, joseluis.sanchezlopez, holger.voos}@uni.lu

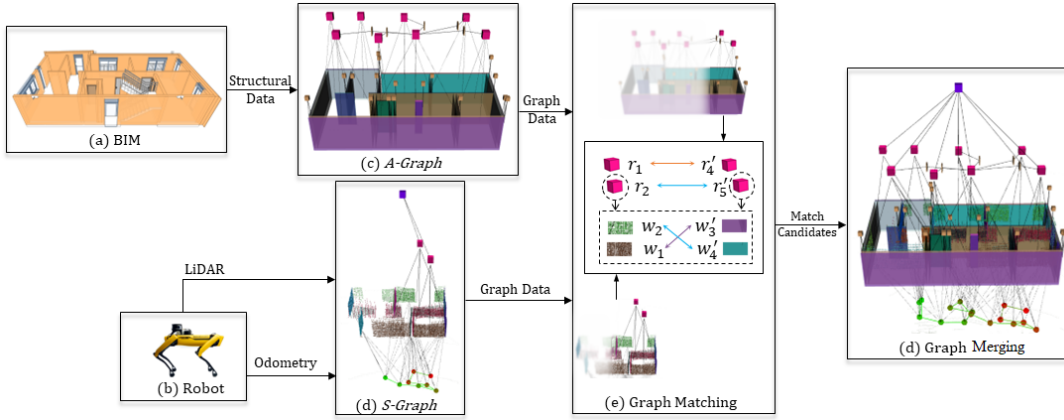[2]Author is with I3A, Universidad de Zaragoza, Spain jcivera@unizar.es

Fig. 1: Overview of our approach. We generate offline an Architectural Graph (*A-Graph*) from a BIM model. A robot estimates online a Situational Graph (*S-Graph*) from its sensors. We do graph matching between the two, align them, and merge their information. This generates the final *iS-Graphs*, which is utilized by the robot to be localized with respect to the BIM.

(Section. V) provides the merged *iS-Graph* utilized for the global localization with respect to the BIM frame of reference B.

A. Situational Graphs (S-Graphs)

S-Graphs are four-layered optimizable hierarchical graphs built online using 3D LiDAR measurements. The full details of the *S-Graphs* we use in this work can be found in [5], [6]. In brief, their four layers can be summarized as:

Keyframes Layer. It consists of the robot poses factored as $^{M}\mathbf{x}_{R_i} \in SE(3)$ nodes in the map frame M with pairwise odometry measurements constraining them.

Walls Layer. It consists of the planar wall-surfaces $^{M}\boldsymbol{\pi}_i \in \mathbb{R}^3$ extracted from the 3D LiDAR measurements and factored using minimal plane parameterization. The planes observed by their respective keyframes are factored using pose-plane constraints.

Rooms Layer: It consists of two-wall rooms $^{M}\boldsymbol{\gamma}_i \in \mathbb{R}^2$ or four-wall rooms $^{M}\boldsymbol{\rho}_i \in \mathbb{R}^2$, each constraining either two or four detected wall-surfaces respectively.

Floors Layer: It consists of a floor node $^{M}\boldsymbol{\xi}_i \in \mathbb{R}^2$ positioned at the center of the current floor level and constraining all the rooms present at that floor level.

III. ARCHITECTURAL GRAPHS (A-GRAPHS)

We extract relevant information from BIM models into two-layered optimizable graphs denoted as *A-Graphs*. In the lowest-level layer, we will model the geometry of the walls, and in the highest level the rooms. Room-to-wall constraints connect the two layers and neighboring rooms are constrained by doorways. The specific formulation is detailed below.

A-Walls Layer. This layer extracts all the information about the walls and wall surfaces from the BIM and connects them with appropriate wall-to-wall-surface edges.

Wall-Surfaces: Wall-surfaces are planar entities $^{B}\boldsymbol{\pi}$ extracted in the BIM frame of reference B. All the wall-surfaces are converted to their Closest Point (CP) represen-

tation, as in [6]. Wall-surface normals with their component $^{B}n_x$ greater than $^{B}n_y$ are classified as x-wall-surfaces, and wall surfaces whose normal component $^{B}n_y$ is greater than the normal component $^{B}n_x$ are defined as y-wall-surfaces. These are initialized in the graph as $^{B}\boldsymbol{\pi} = [^{B}\phi, {}^{B}\theta, {}^{B}d]$, where $^{B}\phi$ and $^{B}\theta$ stand for the azimuth and elevation of the plane in frame B and ^{B}d is the perpendicular distance in B.

Walls: We introduce a novel semantic entity with respect to [6] in the form of a *Wall* $\boldsymbol{\omega} \in \mathbb{R}^3$, consisting of two planar wall-surfaces. Two opposed planar wall-surface entities either in x-direction or y-direction with similar perpendicular distance ^{B}d can be classified as a part of a single wall entity. The wall center $^{B}\boldsymbol{\omega}_{x_i}$ for two opposed x-direction wall-surfaces is computed as:

$$
{}^{B}\mathbf{w}_{x_i} = \frac{1}{2}\big[|{}^{B}d_{x_1}| \cdot {}^{B}\mathbf{n}_{x_1} - |{}^{B}d_{x_1}| \cdot {}^{B}\mathbf{n}_{x_2}\big] + |{}^{B}d_{x_2}| \cdot {}^{B}\mathbf{n}_{x_2}
$$
$$
{}^{B}\boldsymbol{\omega}_{x_i} = {}^{B}\mathbf{w}_{x_i} + \big[{}^{B}\mathfrak{s}_i - \big[\,{}^{B}\mathfrak{s}_i \cdot {}^{B}\hat{\mathbf{w}}_{x_i}\big] \cdot {}^{B}\hat{\mathbf{w}}_{x_i}\big] \quad (1)
$$

where $^{B}\mathfrak{s}_i \in \mathbb{R}^3$ is the starting point for a given BIM wall and $\mathbf{n}$ and d are the plane normals and distance. For Eq. III to hold true, all plane normals are converted to point away from the BIM frame of reference as in [6]. The wall center along with it wall-surfaces is factored in the graph as:

$$
c_{\boldsymbol{\omega}}\big(^{B}\boldsymbol{\omega}_i, \big[^{B}\boldsymbol{\pi}_{x_1}, {}^{B}\boldsymbol{\pi}_{x_1}, {}^{B}\mathfrak{s}_i\big]\big)
$$
$$
= \sum_{i=1}^{K} \|^{B}\hat{\boldsymbol{\omega}}_i - f(^{B}\tilde{\boldsymbol{\pi}}_{x_1}, {}^{B}\tilde{\boldsymbol{\pi}}_{x_1}, {}^{B}\mathfrak{s}_i)\|^2_{\boldsymbol{\Lambda}_{\tilde{\boldsymbol{\omega}}_{i,t}}} \quad (2)
$$

Where $f(^{B}\tilde{\boldsymbol{\pi}}_{x_1}, {}^{B}\tilde{\boldsymbol{\pi}}_{x_1}, {}^{B}\mathfrak{s}_i)$ is the function mapping the wall center using the wall-surfaces and its starting point following Eq. III. Wall factors add an additional layer of structural consistency to the graph. A Wall center for opposed planes in y-direction is computed following Eq. III.

A-Rooms Layer. The second layer of the graph extracts all the information about the rooms along with the door-ways

interconnecting the rooms.

Rooms: We use the similar concept of a four-wall room $^B\boldsymbol{\rho} \in \mathbb{R}^2$ as presented in [6], where each room comprises the four-wall surfaces extracted in the first layer of the graph.

Door-Ways: We incorporate an additional entity in the graph called door-ways interconnecting the room nodes, easily available from BIM. The position of a door-way node $^B\boldsymbol{\mathcal{D}} \in \mathbb{R}^3$ is directly extracted from BIM in the frame of reference B. Using the semantic information from BIM of the rooms connected by a given door-way, the door-way-to-rooms factor can be formulated as:

$$c_{\boldsymbol{\mathcal{D}}}(^B\boldsymbol{\rho}_1, {}^B\boldsymbol{\rho}_2, {}^B\boldsymbol{\mathcal{D}}_i) = \\ \| f(^B\hat{\boldsymbol{\rho}}_1, {}^{\rho_1}\hat{\boldsymbol{\mathcal{D}}}_i) - f(^B\hat{\boldsymbol{\rho}}_2, {}^{\rho_2}\hat{\boldsymbol{\mathcal{D}}}_i)\| \quad (3)$$

Where $^B\boldsymbol{\rho}_1$ and $^B\boldsymbol{\rho}_2$ are the four-wall rooms connected to the door-way $^B\boldsymbol{\mathcal{D}}_i$. $^{\rho_1}\boldsymbol{\mathcal{D}}_i$ and $^{\rho_2}\boldsymbol{\mathcal{D}}_i$ are the positions the door-way nodes estimated with respect to rooms $^B\boldsymbol{\rho}_1$ and $^B\boldsymbol{\rho}_2$.

IV. GRAPH MATCHING

Our second contribution to this paper is a novel approach to graph matching in which we match an architectural *A-graph*, $\mathcal{G}_a$, and a *S-Graph*, $\mathcal{G}_s$. We compare and match the room (ρ) entities and for each room entity its corresponding wall-surfaces (π), extracted from both graphs. The correspondences form a bipartite graph connecting the nodes of some parts of $\mathcal{G}_a$ with all, or almost all nodes, in $\mathcal{G}_s$, at the rooms and walls layers. $\mathcal{G}_s$ is built incrementally as the robot navigates the environment, the graph matching is run after every map update until a successful match is obtained. A schema of the entire graph-matching process is described in Fig. 2.

Notation. Let $\boldsymbol{V}$ be any set of nodes, with $\boldsymbol{V}_a$ and $\boldsymbol{V}_s$ the sets of nodes in the *A-Graph* and the *S-Graph* respectively. Let $\boldsymbol{m} = (v_a, v_s) : v_a \in V_a, v_s \in V_s$, and $\boldsymbol{M}$ be any set of m such as $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, ..., \boldsymbol{m}_n\}$. Let $\mathcal{M}$ be any set of M such as $\mathcal{M} = \{\boldsymbol{M}_1, \boldsymbol{M}_2, ..., \boldsymbol{M}_n\}$. Local candidates are $\boldsymbol{M}$ including $\boldsymbol{m}$ referring to a small part of the input graphs. Global candidates are $\boldsymbol{M}$ including $\boldsymbol{m}$ referring to the entire input graphs.

V. GRAPH MERGING

Our global state $\mathbf{s}$ at time T, before graph merging, contains all the nodes of the *A-Graph*, generated offline, as well as the current nodes estimated online by the *S-Graph*

$$\mathbf{s} = [^M\mathbf{x}_{R_1}, \ldots, {}^M\mathbf{x}_{R_T}, \\ {}^M\boldsymbol{\pi}_1, \ldots, {}^M\boldsymbol{\pi}_P, {}^B\boldsymbol{\pi}_1, \ldots, {}^B\boldsymbol{\pi}_Q, \\ {}^M\boldsymbol{\rho}_1, \ldots, {}^M\boldsymbol{\rho}_S, {}^B\boldsymbol{\rho}_1, \ldots, {}^B\boldsymbol{\rho}_R \quad (4) \\ {}^M\boldsymbol{\gamma}_1, \ldots, {}^M\boldsymbol{\gamma}_G, {}^M\boldsymbol{\xi}_1, \ldots, {}^M\boldsymbol{\xi}_E, \\ {}^B\boldsymbol{\omega}_1, \ldots, {}^B\boldsymbol{\omega}_W, {}^B\boldsymbol{\mathcal{D}}_1, \ldots, {}^B\boldsymbol{\mathcal{D}}_D, \\ {}^B\mathbf{x}_M]^\top,$$

where $^B\mathbf{x}_M$ is the estimated transformation between the map frame M of the *S-Graph* and the BIM frame B of the *A-Graph*, which is set to identity before graph merging. The

graph matching method from Section. IV provides match candidates between the room nodes and the wall-surface nodes of the *S-Graph* and the *A-Graph*. To efficiently merge the two graphs to generate the *iS-Graph*, we introduce room-to-room constraints as well as a wall-surface-to-wall-surface constraints between the matched candidates. The room-to-room constraint is defined as

$$c_{\boldsymbol{\rho}}(^B\boldsymbol{\rho}_1, {}^M\boldsymbol{\rho}_2) = \| {}^B\hat{\boldsymbol{\rho}}_1 - {}^M\hat{\boldsymbol{\rho}}_2\|^2_{\boldsymbol{\Lambda}_{\hat{\boldsymbol{\rho}}_{1,2}}}, \quad (5)$$

where $^B\boldsymbol{\rho}_1$ is the room node in the *A-Graph* and $^M\boldsymbol{\rho}_2$ is the corresponding room node in *S-Graph*. Similarly, for all correspondences between wall-surface candidates, the constraint is formulated as

$$c_{\boldsymbol{\pi}}(^B\boldsymbol{\pi}_1, {}^M\boldsymbol{\pi}_2) = \| {}^B\boldsymbol{\pi}_1 - {}^M\boldsymbol{\pi}_2\|^2_{\boldsymbol{\Lambda}_{\tilde{\boldsymbol{\pi}}_{1,2}}}, \quad (6)$$

where $^B\boldsymbol{\pi}_1$ and $^M\boldsymbol{\pi}_2$ are the wall-surfaces in the *A-Graph* and the *S-Graph*.

With the constraints between the two graphs, $^B\mathbf{x}_M$ can be estimated and all the robot poses, the wall-surfaces, rooms, and floors of the *S-Graph* can be referred accurately with respect to the BIM frame of reference B of the *A-Graph*, resulting in the final improved situational graph *iS-Graph*. In this manner the robot is localized with respect to the global reference of the architectural plan. Our approach thus can perform global localization exploiting the hierarchical high-level information in the environment without the need for appearance-based loop closure constraints at keyframe level, which are more variable.

VI. EXPERIMENTAL EVALUATION

A. Experimental Setup

We validated our proposed approach in multiple simulated and real-world construction environments. In all experiments, we generated *A-Graphs* from various building models that were created in Autodesk Revit. The Boston Dynamics *Spot* robot, equipped with a Velodyne VLP-16 3D LiDAR, was teleoperated for data collection. We compared our approach to two 2D LiDAR-based localization algorithms (AMCL [7] and Cartographer [8]) and one 3D LiDAR-based localization (UKFL [9]) algorithm. In simulated datasets we measured the Absolute Pose Error (APE) [10] with respect to the ground truth, while in real-world due to the absence of ground truth pose information, we compare the point cloud RMSE of the generated map with the available ground truth map from BIM. The proposed methodology was implemented in C++, and the experiments were validated on an Intel i9 16-core workstation.

B. Results and Discussion

Simulated Datasets. Table. I presents the APE of our proposed *iS-Graphs* and state-of-the LiDAR-based localization algorithms for six simulated datasets. Each simulated dataset represents a single floor level of a given construction environment with varying configurations of wall-surfaces and rooms. Given the output from 2D LiDAR algorithms, the
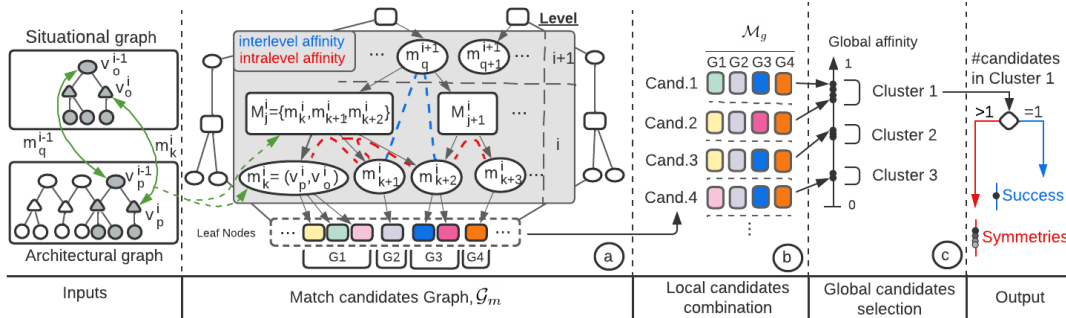
Fig. 2: Graph matching schema. a) Downwards and at each level, different combinations of matches are proposed and selected by their geometrical affinity either at the same level or with the associated upper-level pair. b) The match graph is traversed upwards while combined with same-level nodes to define all-level match candidates. c) The lowest-level pairs of every candidate are scored in global affinity. That score is clustered to find symmetries in the best cluster.

TABLE II: Point cloud RMSE [m] for our real-world dataset. Best results are boldfaced. '−' stands for localization failure.

Method	Alignment Error ↓		
	Datasets		
	D1	D2	D3
AMCL [7]	-	0.90	0.98
UKFL [9]	-	0.86	0.69
Cartographer [8]	-	0.58	0.64
iS-Graphs (ours)	**0.17**	**0.20**	**0.21**

TABLE I: Absolute Pose Error (APE) [m] for several LiDAR-based localization baselines and our *iS-Graphs* Datasets have been recorded in simulated environments. '−' stands for localization failure.

Method	APE [m] ↓					
	Datasets					
	D1	D2	D3	D4	D5	D6
AMCL [7]	2.04	1.71	2.03	2.01	-	-
UKFL [9]	0.97	0.78	-	0.74	0.70	0.88
Cartographer [8]	0.10	0.16	0.12	-	0.13	-
iS-Graphs (ours)	**0.08**	**0.01**	**0.02**	**0.04**	**0.09**	**0.12**

APE of all the algorithms is computed in 2D (x, y, θ). Table. I shows that our *iS-Graphs* shows higher robustness against localization failure and outperforms all the localization baselines using both 2D and 3D LiDARs.

Real-World Datasets. Table. II presents the point cloud RMSE for three different construction environments, where our *iS-Graphs* is able to localize the robot correctly while providing a more accurate 3D map of the environment as compared to the ground truth. Because of noisy LiDAR measurements and the clutter in a real construction environment traditional approaches to localization fail to localize the robot (see D1 in Table. II). Since our approach relies only on high-level entities like wall-surfaces and rooms and their topological relationship instead of directly relying on low-level LiDAR measurements, it is more robust to noise and clutter in the environment than the baselines.

VII. CONCLUSION

In this paper, we presented a novel method for global robot localization utilizing prior information from architectural plans. We embedded the architectural data from BIM models into optimizable graphs called Architectural Graphs (*A-Graphs*). Using Situational Graphs (*S-Graphs*) estimated by a robot as it navigates its environment, we present a novel graph matching strategy to match the *A-Graphs* with the *S-Graphs* and we also present a graph merging strategy to fuse both graphs. The result of the fusion is an informed *iS-Graph*, which enables the robot to localize itself within the architectural plan for the given environment.

REFERENCES

[1] S. Azhar, M. Hein, and B. Sketo, "Building information modeling (bim): benefits, risks and challenges," in *Proceedings of the 44th ASC Annual Conference*, 2008, pp. 2–5.

[2] R. W. M. Hendrikx, P. Pauwels, E. Torta, H. J. Bruyninckx, and M. J. G. van de Molengraft, "Connecting semantic building information models and robotics: An application to 2d lidar-based localization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 654–11 660.

[3] M. S. Moura, C. Rizzo, and D. Serrano, "Bim-based localization and mapping for mobile robots in construction," in *2021 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2021, pp. 12–18.

[4] M. Shaheer, H. Bavle, J. L. Sanchez-Lopez, and H. Voos, "Robot localization using situational graphs and building architectural plans," 2022. [Online]. Available: https://arxiv.org/abs/2209.11575

[5] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "Situational graphs for robot navigation in structured indoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9107–9114, 2022.

[6] ——, "S-graphs+: Real-time localization and mapping leveraging hierarchical representations," 2022. [Online]. Available: https://arxiv.org/abs/2212.11770

[7] D. Fox, "Kld-sampling: Adaptive particle filters," *Advances in neural information processing systems*, vol. 14, 2001.

[8] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1271–1278.

[9] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, Mar. 2019.

[10] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," https://github.com/MichaelGrupp/evo, 2017.