Evaluation Framework for Indoor Localization Systems in the AEC Environment

Lukas Fritzsche#, 1, Rolf Starke#, 1, Benjamin Felbrich2, Ilija Vukorep1

1 Faculty of Architecture, Civil Engineering and Urban Planning, BTU Cottbus-Senftenberg, Germany
2 Felbrich Holding UG
# contributed equally
lukas.fritzsche@b-tu.de, starke@b-tu.de, benjamin@felbrich.com, ilija.vukorep@b-tu.de

Abstract –

Accurate spatial localization is critical in the construction industry, indoor robotics, and other applications involving actor movement within buildings. This paper introduces a combined software and hardware framework designed to evaluate indoor localization systems leveraging optical tracking integrated with a ROS communication system. It supports recording location data, aligning trajectories and evaluating the recorded samples against the ground truth allowing for real world comparison within a custom setup. While it is designed to evaluate any localization system, a presented case study compares two established SLAM algorithms regarding their suitability for an AEC application.

Keywords –

localization evaluation, indoor localization, mobile robotic, SLAM, ROS

1 Introduction

The rapid advancements in indoor localization technologies have led to their increasing integration across diverse industries, including construction, robotics, and indoor navigation [1]. Accurate spatial localization is pivotal for Architecture, Engineering and Construction (AEC) applications ranging from autonomous robots[2] operating in confined spaces to real-time monitoring of dynamic processes (Figure 1). However, the multitude of available localization methods — each with unique strengths, limitations, and precision levels — poses a challenge for selecting the most suitable system for specific applications [3].

The proposed framework does not aim to create a new benchmark for evaluating SLAM algorithms. Instead, it builds on existing benchmarks, metrics, and computer vision algorithms to help industries adopt and improve localization systems, advancing robotic development in AEC. The system supports a context-aware comparison of suitable localization methods in a realistic environment, rather than validating their performance under laboratory conditions. This novel evaluation framework combines optical tracking with a modular ROS-based software system. Designed to measure and compare the precision of localization systems, the framework provides a robust method for evaluating different approaches, allowing for informed decisions in research and development. While the framework is adaptable to various technologies, this work demonstrates the workflow in a case study focusing on two widely used localization algorithms.

2 Background and Related Work

Precise indoor localization of actors in AEC — like robots, machines, and sensing devices — is crucial for their operation. Indoor localization has the problem of being in the “shadow” of the building and thus cannot utilize satellite localization.

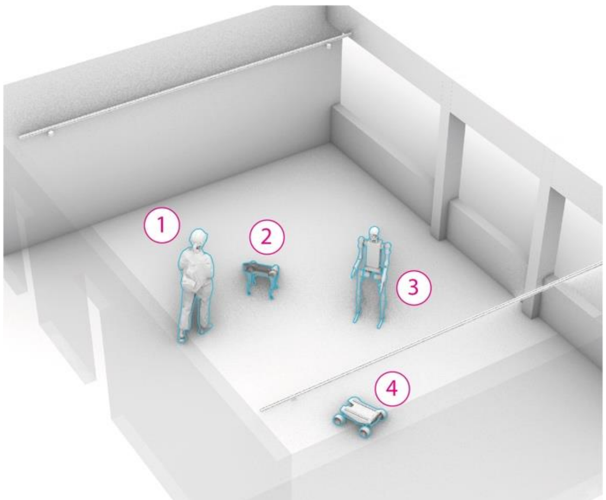

Figure 1. indoor mobile robots or devices that use localization: 1) handheld device; 2) quadruped robot; 3) humanoid robot; 4) four-wheeled robot.

For precise localization inside buildings there are many technologies like Inertial Measurement Units (IMU), Visual Odometry, Simultaneous Localization and Mapping (SLAM) with cameras or Lidar. These different methods are often unified in sensor fusion systems to increase precision [4].

To compare and evaluate the precision of different localization methods common databases and benchmark suites are used that include high-quality sensor data collected from real-world driving scenarios, including stereo camera images, 3D point clouds from LiDAR, and GPS/IMU data, enabling researchers to test and compare algorithms under consistent conditions. While there is an abundance of outdoor datasets like Aachen Day-Night Dataset [5], CrowdDriven Dataset [6], Extended CMU Seasons Dataset, RobotCar Seasons Dataset, Symphony Seasons Dataset some indoor datasets like InLoc Dataset [7], Gangnam Station and Hyundai Department Store Datasets, ETH-Microsoft Dataset [8] or the TUM RGB-D Dataset [9] can be found as well. With the help of these datasets researchers can develop their best localization algorithm or hardware system.

Once a localization algorithm for an application is chosen, the next steps require testing it in the context of the application in order to validate the promised precision. This is where the presented frameworks evaluation and calibration commences. To our knowledge there are no out of the box solutions allowing to determine and compare the suitability of different localization algorithms for a custom environment.

Technical descriptions of applied evaluation and calibration methods in previously mentioned localization software solutions differ widely. For example, in [9] handheld Kinect RGBD cameras were tracked by 8 motion tracking high-speed cameras to accurately track the pose of the localization device. In [10] IMUs were added to camera sensors and a motion tracking system was installed in the starting position where also the scanning ended (loop closure) providing ground truth for the TUM VI dataset. A similar setup was also used for the creation of the TUM VIE dataset [11] where the sensorial layers were enriched with stereo-event data. Other SLAM evaluation frameworks [12], [13] compare different algorithms in one benchmark environment, while the presented framework can be used to compare one algorithm in different environments. Existing frameworks also concentrate on calculating error metrics for specified trajectories and are not equipped to provide a comprehensive workflow for recording the positional data.

3 System description

3.1 Hardware Setup

A crucial part of the proposed system is to provide precise ground truth positional data for the localization actor which is achieved by an outside-in tracking system based on high-speed IR cameras. This section provides a detailed overview of the hardware components used, their specifications, and their integration.

Figure 2. 1) Hardware setup in the case-studies lab environment (7x10m) consisting of 2) four OptiTrack Prime X 13 cameras (ground truth) and 3) a depth camera for localization (e.g. RealSense D455)

Four OptiTrack Prime X 13 cameras are installed on the ceiling to observe a designated tracking area with the size 7 x 10 m (Figure 2). These cameras, chosen for their high-speed capture rates and submillimetre precision of ±0.20 mm [14], are positioned strategically to provide comprehensive coverage of the monitored space. The cameras are connected to the central computer via an 8-port Ethernet PoE/PoE+ switch, specifically the TRENDnet TPE-TG80G model. This switch enables stable data transmission and supplies power to the cameras through Ethernet cables, simplifying the overall cabling requirements and ensuring uninterrupted operation. A workstation running Windows 11 and

OptiTrack's Motive 3.1.4 motion capture software [15] transforms the raw tracking data into user-defined frames. A calibration workflow is applied to align the coordinate spaces of the localization algorithm and the OptiTrack.

For the calibration, as a critical aspect of the setup, marker spheres with a diameter of 9.5 mm are employed. These markers are specifically designed to attach to the camera used for localization, facilitating accurate calibration by ensuring they are easily detectable by the OptiTrack cameras.

3.2 Software Setup

The software architecture supporting the optical tracking and localization system is designed to operate efficiently within a modular, containerized environment. This approach ensures scalability, ease of deployment, and isolation of various system components.

In order to achieve comparable results and to allow offline runs of computationally intensive localization systems, the framework supports a workflow where the localization sensor data is first recorded as *rosbags* [16] and the evaluation is performed at a later stage.

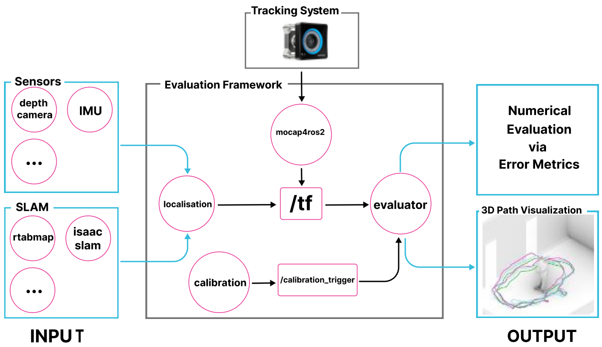

Figure 3. Overview of the proposed framework. Application specific components like sensors and localization algorithms (left) are utilized and extended by the proposed framework's components (centre) which record, align and evaluate positional data. Results are presented in the form of well-established error metrics as well as a visualization of the recorded trajectories (right).

3.2.1 Containerized ROS2 Instances

The foundation of the software system consists of ROS2 instances deployed as individual Docker containers. Each container encapsulates a specific functionality (Figure 3), ensuring a clean separation of responsibilities and enabling seamless communication between components via ROS2 nodes. This architecture leverages Docker's containerization capabilities to isolate dependencies, simplify system updates, and allow independent testing of each component.

3.2.2 OptiTrack Integration in ROS2 Docker

The motion capture system recording the ground truth localization data is wrapped in a ROS2 node hosted within a Docker container, which acts as a receiving counterpart to the Motive Software as NatNet client [17]. This node handles the optical tracking processes, transforming the tracking data captured from the OptiTrack Prime X 13 cameras and publishing the calculated positions of tracked objects.

The component is based on the mocap4ros2 package [18] and extended to send transform data in a format that is compatible with the overall setup. The container communicates with other system components by broadcasting positional data in real time, ensuring high accuracy and low latency.

3.2.3 Localization system in ROS2 Docker

A separate Docker environment encapsulates any given localization system, thereby ensuring its isolation and enabling seamless integration with diverse sources of localization. Within this environment the localization system needs to be configured to publish the measured pose via a ROS2 node. The localization component operates in parallel with the OptiTrack container, enabling the system to combine optical tracking and visual localization seamlessly.

3.2.4 Calibration and Evaluation System in ROS2 Docker

In a third dockerized environment the evaluation system runs as a custom ROS2 node. By subscribing to localization data published by the other system components and reacting to user input, as described in the calibration section, this unit handles the core evaluation logic. It provides a real time visualization of the different poses via the default 3D visualizer for ROS2, captures the data frames and provides logic for evaluating the captured data.

3.2.5 Inter-Component Communication

The ROS2 ecosystem facilitates communication between the Docker containers through ROS2 topics and nodes. Each container publishes data, such as positional information or error metrics, which other containers or scripts can subscribe to for further processing. The modularity provided by ROS2 ensures that each component remains loosely coupled while maintaining synchronization and data consistency across the system. Additionally, ROS2's design for distributed systems enables low-latency configurations across various computing environments, making the integration of a new localization component particularly straightforward due to the minimal setup required.

In order to maximize compatibility with different localization systems, the framework utilizes the tf2 package as a well-established standard for capturing coordinate frames over time [19]. The ground truth as well as the localization system component publish the measured poses as transform. The evaluation component is continuously consuming those transforms and processes them to be used for evaluation.

3.3 Initialization, Calibration and Trajectory Alignment

To evaluate the accuracy of the localization system, the deviation between the poses as measured respectively by the OptiTrack system and by the localisation system must be determined.

Since these originate from different systems their coordinate spaces will differ. The evaluation framework provides a registration workflow as a systematic approach to abstract from the substantial complexity inherent in aligning these poses and transforming them into a shared coordinate space.

3.3.1 Registering the Test Environment and Calibrating Ground Truth

In the initial step the test environment needs to be prepared. The motion capture system serving as ground truth must be calibrated by traversing the tracking volume with a standardized calibration device.

As part of this procedure, the origin position and orientation of the global coordinate space must be defined in the physical world by placing a specific marker on the ground floor. The alignment of the origin with a physical feature e.g. a wall or corner is recommended to greatly simplify subsequent evaluation, as it enables the alignment of a 3D model with minimal effort.

3.3.2 Measuring and Specifying Physical Configuration of the Localization Device

To track the localization device with OptiTrack, a sufficient number (3-5pcs) of markers must be fixed statically to the device.

Prior to the OptiTrack system tracking the pose of the device, it is necessary to register the marker configuration within the Motive Software. As part of this process a *rigid body* is created, which captures the static offsets between the markers as one physically connected object. After the rigid body is created, the corresponding coordinate frame should be adjusted to match ROS coordinate space definition as defined in as defined in REP103/105: x → forward, y → left, z → up [20], [21]

To facilitate the alignment of poses calculated by both localization and ground truth, the physical configuration of the tracking device must be specified and loaded into the evaluation framework. This can be achieved by measuring the offsets between the origin of the rigid body and the origin of the used tracking solutions e.g. the camera centre.

Placing the origin at the position of one of the used markers will simplify measuring the offsets.

The framework utilizes the well-established *universal robot description format (urdf)* [22] for reading the devices physical configuration.

3.3.3 Aligning Coordinate Frames

After the ground truth has been calibrated and the device has been set up for being localized in OptiTrack, an evaluation run can be conducted.

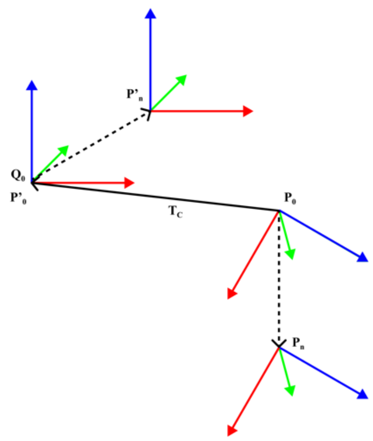

Figure 4. Alignment of the different coordinate systems of ground truth *Q* and the localization system *P* at calibration data sample *t*=0 and some subsequent sample *t*=*n*. Using the calibration transform T_C the calibrated transform *P*’ can be obtained at any point in time.

As soon as both localization systems are running and the tracking device’s markers are fixed, the alignment procedure of the evaluation system must be triggered. By pressing a key, the user starts the evaluation run. The evaluation system stores the tracked transform P_0 as well as the ground truth transform Q_0 at this timestamp *index* = 0. The recording of poses is started.

From now on the system calculates a new pose P'_0 representing the calibrated tracked pose (i.e. the pose measured by the localization system transformed into OptiTrack’s coordinate space). In every subsequent sample with *index n* the poses Q_n and P'_n can be utilized to calculate the deviation as shown in Figure 4.

3.3.4 Timestamp Alignment

The data packages sent via NatNet streaming protocol are enriched with the timestamp when this data was recorded, measured in turn with the OptiTrack PC’s clock. The OptiTrack ROS2 node publishes the corresponding transform with the original timestamp, compensating for latency.

Any pose evaluation needs to be calculated with exact matching timestamps for it to allow conclusions about the trajectory. Possible timestamp mismatches between the ground truth and the localization system can arise due to differences in sensor frequencies, outliers through sensor measurement errors, tracking losses in either the ground truth or the localization system, and delays introduced by the processing times of raw image data recording, the SLAM algorithm, and the transform alignment calculation. To compensate for those mismatches the delay in the localization system's timestamps, which is assumed to be static, needs to be determined and corrected for a static value or the timestamps need to be upsampled, downsampled or interpolated before the evaluation.

3.4 Means of Evaluation

To support a wide range of localization applications the proposed framework implements several means of evaluating the chosen localization system's precision. With the availability of the first RGB-D image datasets, which were infused with ground truth camera poses [9] the absolute trajectory error (ATE) as well as the relative error or relative pose error (RPE) became the predominant error metrics for localization systems. Adaptations of ATE and RPE calculations from public repositories [12] are included in the proposed framework.

3.4.1 Numerical Evaluation

RPE describes the difference between the estimated motion and the true motion for each pose, and is used to calculate drift, which describes the accumulation of sensor measurement errors during visual odometry [23]. While computationally expensive and difficult to interpret, RPE is less sensitive to estimation errors and allows for a detailed evaluation for any chosen segment size of the trajectory. Since it relies on motion, it is used in scenarios where accurate short term motion estimation is more important than precise global positioning e.g. wearable VR or AR devices.

ATE is a simple yet comprehensive evaluation metric. It calculates the absolute pose difference from position error, rotation error and velocity error at the time step $_i$ between two trajectories. Specifically, the localization systems trajectory P is aligned with the ground truth trajectory Q using the rigid-body transformation S .

$$F_i := Q_i^{-1} S P_i \tag{1}$$

The resulting error F is commonly used to calculate the Root Mean Squared Error which reduces the errors over all time steps to a single number metric. The requirement for a ground truth trajectory and the lack of compensation for drift, which disproportionately emphasizes errors in the initial segment of the localizations system's trajectory compared to those occurring later, are recognized as limitations of the ATE metric. Nonetheless, it is a prominent metric for SLAM evaluation due to its simplicity, ease of comparison, and intuitive visual interpretability [23].

3.4.2 Graphical Evaluation

The proposed framework is specifically designed to utilize localization algorithms among others for construction site monitoring, 3D scanning, surveying, autonomous equipment navigation, indoor inspections.

Therefore it includes a custom ROS2 node which records every frame update, providing a trajectory or path visualization in *rviz* [24], as well as a grasshopper 3d [25] script, which translates the recorded path into control points of a polyline for a seamless integration with commercial CAD software (see Figure 5), allowing non-experts to quickly assess if a localization system is operating within desired tolerances for a chosen application.

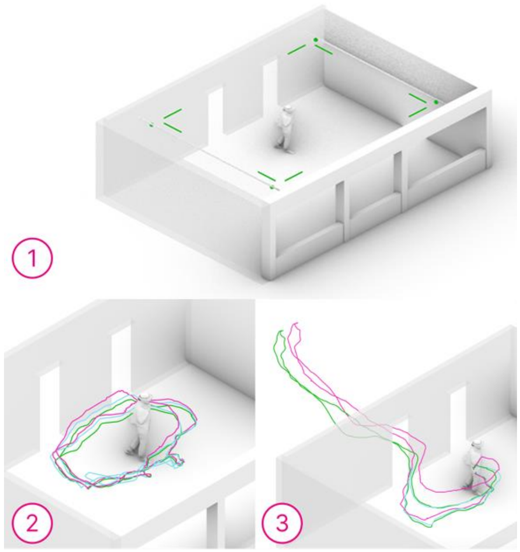

Figure 5. Visualization of two exemplary runs using the SLAM algorithms within a 1) lab environment. 2) Bottom shows 2) a circular path through a single room and 3) a run through multiple rooms in Rhino 3d. The ground truth is shown in *cyan*, results from RTAB-Map SLAM in *magenta* and ISAAC SLAM's trajectory in green.

4 Application of the Evaluation Framework

4.1 Description of a Case Study

To demonstrate and validate the proposed framework, an exemplary scenario is examined where a handheld device based on the NVIDIA Jetson AGX Orin platform equipped with a Stereolabs ZED X stereo camera is used for semi-automated building material and component inspection as part of so-called pre-demolition audits. In this scenario, machine learning based object recognition is combined with a visual inertial localization system to create an accurate material inventory. To achieve reliable results in this application, it is critical that the localization component of the system provides an accurate location of the actor. With RTAB-Map SLAM [26] and the Isaac Visual SLAM [27] two state-of-the-art visual-inertial SLAM systems were selected for their high on-paper accuracy as well as their high performance and flexibility within the ROS ecosystem. To determine which algorithm to use, an in-depth evaluation was performed in a realistic environment using the proposed evaluation framework.

4.2 Results

The results of the localization were then evaluated for the test run with a length of 28.4m. The ATE and RPE were calculated between the aligned trajectories of the SLAM systems and the OptiTrack serving as the ground truth. (see Figure 5, Table 1 and Table 2)

Table 1 Absolute Trajectory Error (ATE)

error metric	RTAB-Map	ISAAC
RMSE	0.194m	0.256 m
MEAN	0.159 m	0.224 m
STD	0.111 m	0.123 m
MIN	0.000 m	0.000 m
MAX	0.586 m	0.686 m

With an ATE Root Mean Squared Error *RMSE* of 19.4 cm RTAB-MAP outperforms the ISAAC SLAM.

Table 2 Relative Pose Error (RPE)

error metric	RTAB-Map	ISAAC
RMSE	0.058 m	0.047 m
MEAN	0.028 m	0.026 m
STD	0.051 m	0.039 m
MIN	0.000 m	0.000 m
MAX	0.458 m	0.337 m

For the Relative Pose Error *RPE*, when calculating drift per frame at $\Delta = 1$, the ISAAC SLAM outperforms the RTAB-Map SLAM, achieving a Root Mean Square Error *RMSE* of 4.7 cm.

Based on the presented Evaluations (Table 1 and Table 2), a logical inference is to prioritize RTAB-Map SLAM for tasks requiring high stability due to its low ATE, while selecting ISAAC SLAM for tasks demanding high accuracy, given its superior performance in RPE.

In the above-mentioned application of semi-automated building inventory, the prevention of double counting objects in consecutive frames is pivotal. Only with a low RPE the same object can be safely identified in different frames and correctly positioned in the localization algorithms map. Therefore, ISAAC SLAM should be selected for this application due to its superior RPE in the tested environment.

5 Conclusion, Limitations and Future Work

In order to validate the correctness of the results produced by the framework in the given case study, the error metrics are compared with the accuracy claimed in the literature.

Of all the datasets on which RTAB-Map has been run in [28], the TUM room sequence is the most comparable to the single room setup presented, as it is the only indoor handheld scenario. Within the benchmark, an ATE (RMSE) of 6.6 cm was achieved, which is approximately one third of the accuracy (19.4 cm) determined within the case study. Preliminary investigations have indicated that the observed discrepancy in accuracy is most likely due to the limited processing power of the embedded device and the significantly longer path length.

For ISAAC SLAM, it should be noted that available benchmarks are exclusively for the KITTI dataset. Despite the considerable differences between this scenario and the case study (long-distance outdoor, high-speed vehicle data), a comparison is nevertheless made to validate the feasibility. The translation error with respect to the trajectory length in [29] is given as 0.94%. The ATE (RMSE) of 25.6 cm corresponds to a translation error of 0.9% on the case study trajectory.

A notable difference between the case study results and previously published results highlights the need for individual evaluations to align with specific applications and environments. The proposed framework has been demonstrated to achieve comparable and practical results for this application.

One limitation of the proposed framework is the OptiTrack maximum tracking area size, which effectively limits the availability of ground truth data to one room and makes it difficult to track across different spaces. By having a trajectory start and end within the tracking volume – as shown in 3) in Figure 5 – some fundamental assessments can still be made. For a detailed evaluation in multi-room (or even outdoor) environments another method needs to be chosen.

The accuracy of the OptiTrack system, as stated by the manufacturer, was neither independently verified nor incorporated into the ground truth trajectory of the proposed framework. However, given that the translational error rates of state-of-the-art SLAMs are an order of magnitude larger than OptiTrack's precision (even for shorter trajectories, e.g. 93 mm error with 10 m trajectory for ISAAC SLAM) the influence on the evaluation results is negligible.

Future work could evaluate different localization systems like LIDAR, Bluetooth, Wi-Fi systems and others to develop the proposed framework into a localization system primer for other applications.

Acknowledgments

We gratefully acknowledge the funding from Forschungsinitiative Zukunft Bau des Bundesministeriums für Wohnen, Stadtentwicklung und Bauwesen (reference number SWD-10.08.18.7-24.46).

References

[1] T. Aziz and I. Koo, "A Comprehensive Review of Indoor Localization Techniques and Applications in Various Sectors," *Appl. Sci.*, vol. 15, no. 3, p. 1544, Feb. 2025, doi: 10.3390/app15031544.

[2] J. M. Davila Delgado *et al.*, "Robotics and automated systems in construction: Understanding industry-specific challenges for adoption," *J. Build. Eng.*, vol. 26, p. 100868, Nov. 2019, doi: 10.1016/j.jobe.2019.100868.

[3] J. Li, Y. Song, Z. Ma, Y. Liu, and C. Chen, "A Review of Indoor Localization Methods Leveraging Smartphone Sensors and Spatial Context," *Sensors*, vol. 24, no. 21, p. 6956, Oct. 2024, doi: 10.3390/s24216956.

[4] J. Zhu, H. Li, and T. Zhang, "Camera, LiDAR, and IMU Based Multi-Sensor Fusion SLAM: A Survey," *Tsinghua Sci. Technol.*, vol. 29, no. 2, pp. 415–429, Apr. 2024, doi: 10.26599/TST.2023.9010010.

[5] T. Sattler *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610. Accessed: Jan. 14, 2025. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Sattler_Benchmarking_6DOF_Outdoor_CVPR_2018_paper.html

[6] A. Jafarzadeh *et al.*, "Crowddriven: A new challenging dataset for outdoor visual localization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9845–9855. Accessed: Jan. 14, 2025. [Online]. Available: http://openaccess.thecvf.com/content/ICCV2021/html/Jafarzadeh_CrowdDriven_A_New_Challenging_Dataset_for_Outdoor_Visual_Localization_ICCV_2021_paper.html

[7] H. Taira *et al.*, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," in *CVPR*, 2018.

[8] ETH Zurich Computer Vision Group and Microsoft Mixed Reality & AI Lab Zurich, "The ETH-Microsoft Localization Dataset." 2021. [Online]. Available: https://github.com/cvg/visloc-iccv2021

[9] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal: IEEE, Oct. 2012, pp. 573–580. doi: 10.1109/IROS.2012.6385773.

[10] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," in *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[11] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "TUM-VIE: The TUM Stereo Visual-Inertial Event Dataset," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[12] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM." 2017. Accessed: Jan. 14, 2025. [Online]. Available: https://github.com/MichaelGrupp/evo

[13] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI: IEEE, Jun. 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.

[14] OptiTrack, "Primex 13 Datasheet." Accessed: Jan. 14, 2025. [Online]. Available: https://optitrack.com/cameras/primex-13/specs.html

[15] OptiTrack, "Motive – Optical motion capture software." Accessed: Jan. 14, 2025. [Online]. Available: https://optitrack.com/software/motive/

[16] "rosbag." Accessed: Jan. 14, 2025. [Online]. Available: https://github.com/ros2/rosbag2

[17] OptiTrack, “NatNet SDK.” Accessed: Jan. 14, 2025. [Online]. Available: https://optitrack.com/software/natnet-sdk/

[18] F. Martin, J. M. Guerrero, A. A. Garcia, F. Rodriguez, and V. Matellan, “MOCAP4ROS2: An Open Source Framework for Motion Capture Systems in Robotics,” in *Proceedings of the 18th International Symposium on Open Collaboration*, in OpenSym ’22. New York, NY, USA: Association for Computing Machinery, Sep. 2022, pp. 1–3. doi: 10.1145/3555051.3555076.

[19] T. Foote, “tf: The transform library,” in *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, Apr. 2013, pp. 1–6. doi: 10.1109/TePRA.2013.6556373.

[20] “REP 105 -- Coordinate Frames for Mobile Platforms (ROS.org).” Accessed: Dec. 20, 2024. [Online]. Available: https://www.ros.org/reps/rep-0105.html

[21] T. Foote and M. Purvis, “REP 103 -- Standard Units of Measure and Coordinate Conventions (ROS.org),” REP 103 -- Standard Units of Measure and Coordinate Conventions (ROS.org). Accessed: Jan. 14, 2025. [Online]. Available: https://www.ros.org/reps/rep-0103.html

[22] “urdf - ROS Wiki.” Accessed: Jan. 14, 2025. [Online]. Available: https://wiki.ros.org/urdf

[23] Z. Zhang and D. Scaramuzza, “A Tutorial on Quantitative Trajectory Evaluation for Visual(-Inertial) Odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid: IEEE, Oct. 2018, pp. 7244–7251. doi: 10.1109/IROS.2018.8593941.

[24] Dave Hershberger, “3D visualization tool for ROS.” Accessed: Jan. 14, 2025. [Online]. Available: http://wiki.ros.org/rviz

[25] McNeel, “Grasshopper.” Accessed: Jan. 15, 2025. [Online]. Available: https://www.grasshopper3d.com/

[26] M. Labbé and F. Michaud, “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, 2019, doi: 10.1002/rob.21831.

[27] NVIDIA, “NVIDIA Isaac SLAM.” Accessed: Jan. 14, 2025. [Online]. Available: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_visual_slam

[28] M. Labbé and F. Michaud, “RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation,” *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, Mar. 2019, doi: 10.1002/rob.21831.

[29] A. Korovko, D. Robustov, D. Slepichev, E. Vendrovsky, and S. Volodarskiy, “Realtime Stereo Visual Odometry,” 2018.