RVWO: A Robust Visual-Wheel SLAM System for Mobile Robots in Dynamic Environments

Jaafar Mahmoud[1]*, Andrey Penkovskiy[1]*, Ha The Long Vuong*, Aleksey Burkov, and Sergey Kolyubin[1]

***Abstract*— This paper presents RVWO, a system designed to provide robust localization and mapping for wheeled mobile robots in challenging scenarios. The proposed approach leverages a probabilistic framework that incorporates semantic prior information about landmarks and visual re-projection error to create a landmark reliability model, which acts as an adaptive kernel for the visual residuals in optimization. Additionally, we fuse visual residuals with wheel odometry measurements, taking advantage of the planar motion assumption. The RVWO system is designed to be robust against wrong data association due to moving objects, poor visual texture, bad illumination, and wheel slippage. Evaluation results demonstrate that the proposed system shows competitive results in dynamic environments and outperforms existing approaches on both public benchmarks and our custom hardware setup. We also provide the code as an open-source contribution to the robotics community[2].**

I. Introduction

The demand for wheeled mobile robots in logistics, inspection, and monitoring applications is on the rise, and as such, the need for reliable Simultaneous Localization and Mapping (SLAM) systems for autonomous operation is becoming increasingly critical. Since commercial mobile robots are manufactured in large quantities, it is essential to equip them with cost-efficient sensor setups such as IMUs, encoders, and cameras, rather than expensive LiDAR localization systems. However, real-world environments can pose challenges for each of aforementioned modalities, for moving objects and challenging lighting conditions, which can reduce the effectiveness of visual odometry algorithms. In addition, inertial fusion, particularly with commercial IMUs, may encounter degenerate motion issues. Similarly, wheel odometry may suffer from slippage.

Visual SLAM algorithms are typically designed with the assumption of a static environment. Therefore, their performance may be limited in dynamic environments due to the presence of moving objects. As a result, it is essential for the SLAM system to be able to handle such scenarios and other challenging visual cases. Visual Inertial (VI) systems are popular in the SLAM community, and most state-of-the-art systems, such as [1], [2], provide good performance on benchmark datasets recorded on UAVs, handheld devices, or VR/AR equipment. However, in VI systems, it is important to consider the unobservability of scale and global orientation relative to gravity direction, particularly when the robot performs basic planar movements with constant acceleration (e.g., purely straight or rotational movements). This issue becomes particularly pronounced during the inertial initialization process [3]. This makes accuracy and robustness of VI systems on wheeled robots questionable, as we demonstrate in Section V.

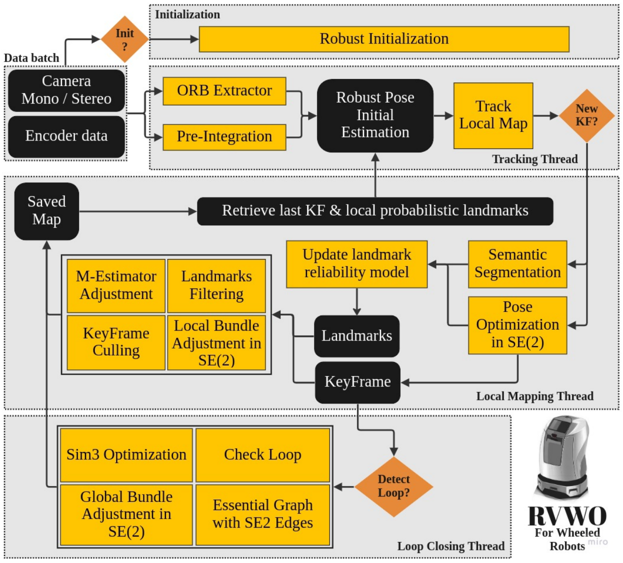

Fig. 1. RVWO pipeline

In this study, we aimed to develop a specialized system for wheeled robots with cost-efficient sensor setups that can operate in challenging environments. Our proposed system operates in real-time, and is based on the architecture of the ORB-SLAM3 system; however, we use only visual and encoder data as input.

In order to mitigate the issue of erroneous data association arising from moving objects, we propose a reliability model for visual landmarks. This model is updated by incorporating prior semantic information of the landmark and its reprojection error value in our back-end optimization process. We introduce an adaptive M-estimator as an optimization kernel to counter the impact of outliers during the Bundle Adjustment process. Additionally, we incorporate the planar motion assumption into the optimization process by imposing two constraints. The first one involves the utilization of preintegrated data from wheel odometry, while the second one involves projecting filtered visual residuals from our probabilistic model onto $SE(2)$ space. This integration leads to a reduction in visual odometry drift, enhances the accuracy of long-term tracking, and improve robustness in challenging scenarios.

* Equal contribution

[1] Faculty of Control Systems and Robotics, ITMO University, St. Petersburg, Russia. {jaafar.a.mahmoud, aapenkovskiy, s.kolyubin}@itmo.ru

[2] https://github.com/be2rlab/rvwo

Manuscript 2133 submitted to 2023 IEEE/RSJ International Conference
on Intelligent Robots and Systems (IROS). Received March 2, 2023.

To ensure the effectiveness of our proposed RVWO system, we conducted extensive evaluations using various benchmarks and a real-world service robot. Firstly, we tested our probabilistic approach on the TUM RGB-D dataset [4], which is a widely recognized benchmark for visual SLAM systems in dynamic environments. Additionally, we evaluated our system on the OpenLoris benchmark [26], which is a well-known benchmark for wheeled robots. Unlike most visual-wheel solutions that only evaluate on their own setup, we chose to evaluate our system on this benchmark to assess its generalizability.

Furthermore, we conducted experiments on a real service robot named Courier (see Fig. 2) to test our system in real-world environment, including detailed analysis of a performance in challenging situations. Our experiments demonstrate that RVWO achieves robust and accurate tracking and mapping results, thus demonstrating the effectiveness of proposed system.

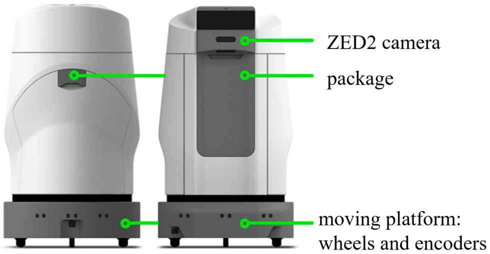

Fig. 2. Wheeled mobile robot used in the evaluation

II. Related work

We split analysis of previous works closely related to our main contributions into two parts.

A. SLAM in Dynamic Environments

There is significant interest in enhancing the robustness of visual SLAM systems by mitigating noise caused by the invalid assumption of a static environment around the robot. Several systems employ RGB-D cameras and utilize depth information, such as Dyna-SLAM [11], which incorporates Mask R-CNN for detecting potentially moving objects and a multi-view geometry-based approach. Another example is SOF-SLAM [12], which uses semantic optical flow and builds upon the RGB-D version of ORB-SLAM2 [13]. DS-SLAM [14] and OFM-SLAM [15] do not rely on depth and instead use a semantic segmentation model in combination with moving consistency checks or epipolar constraints. DRE-SLAM [5] employs an object detection model (YOLO) and K-means clustering for segmentation over depth data from the RGB-D sensor, while Detect-SLAM [16] uses a DNN-based object detector and propagates probabilities of features. These systems primarily focus on outlier rejection during the front-end stage of visual SLAM, whereas proposed approach is based on adaptive M-estimator and designed to directly handle outliers during the optimization stage of the back-end, which is a generalized approach that can be adapted to any setup or system. We believe that increasing the robustness of optimization in such a manner can provide more stable performance in challenging situations where the system may otherwise diverge.

B. Visual-Wheel (VW) Fusion

Several recent works elaborate on the fusion of optical data with wheel odometry. DRE-SLAM [5] addresses the task of building a static map, while odometric measurements from wheel encoders are tightly-coupled with optical data using an optimization-based method. SE2CLAM [6] implemented visual SLAM for $SE(2)$ planar motion as a unary constraint on $SE(3)$ robot pose. SE2LAM [7] proposed a novel constraint $SE(2)-XYZ$ that allows parameterizing robot pose on $SE(2)$ along with considering the $out-of-SE(2)$ motion perturbation. Both systems [6], [7] only support monocular data and require synchronization of wheel odometry with camera data to establish correspondence between visual frames and wheel encoder poses. Moreover, these systems have only been evaluated on a custom setup where the camera is directed towards the roof, reducing the likelihood of encountering dynamic objects. This setup is not widely applicable, as mobile robots may operate with cameras facing forward or backward as well. VINS-on-wheels [3] investigates problems of VI systems on wheeled robots, and extends fusion to involve wheel odometry data. Most of these methods have not undergone extensive testing under realistic conditions, which may involve challenging scenarios such as reflections, sudden changes in light intensity, or moving objects in the field of view. As [26] states, SLAM systems for service robots are really challenging, and need to be properly tested and evaluated to ensure reliability of a system.

III. System overview

The RVWO pipeline is primarily based on the architecture of the ORB-SLAM system, as illustrated in Fig. 1. The system comprises three principal threads, namely Tracking, Local Mapping, Loop Closing. We note major modifications that have been introduced by RVWO below.

A. Input Data Processing

The input data for our system consists of a stream of images (monocular/stereo/RGB-D) along with data from encoders. To effectively utilize the encoder data, we adopted the approach presented in [17], which involves performing preintegration of wheel odometry. This process uses the encoder motion model to acquire relative poses between two consecutive image frames. Unlike other approaches, such as [6] and [7], our method does not require interpolation to be performed, and accept any frequency as input.

B. Robust Initialization

The visual initialization process in ORB-SLAM3 [1] finds the relative initial pose by estimating the fundamental or essential matrices. Additionally, we incorporate measurements from encoders to validate of the initial map and overcome situations when poor visual initialization is performed, especially when the mobile robot performs purely rotational movements.

C. Tracking Thread

The tracking thread is the front-end of the system and is responsible for performing the initial pose estimation. In cases where visual tracking fails due to visual challenges, preintegrated measurements from wheel odometry are utilized to provide better pose estimation. This thread performs the pose optimization in the $SE(2)$ space (see section IV-B), and incorporates the map landmarks with corresponding covariance, which represents their reliability. The probability model of the landmarks is updated in the back-end (Local Mapping Thread).

D. Mapping and Loop Closing Threads

When the tracking thread publishes a new keyframe to the back-end, we perform pose optimization in $SE(2)$ based on local landmarks using the adaptive kernel from our probabilistic model (see Section IV-A). Simultaneously, in a parallel thread, we perform fast semantic segmentation to classify new landmarks using Segformer [18], which has been pre-trained on the ADE20K [19] indoor dataset. This model is capable of producing output images at a rate of 30 Hz. Based on both the semantic class and projection error of each landmark, we update its probability as an inlier. Then, we perform local bundle adjustment in $SE(2)$.

In the loop closing thread, we incorporate an additional odometric constraint to the essential graph optimization between sequential keyframes, and global bundle adjustment is also performed in $SE(2)$.

IV. Robust Visual Wheel Odometry

The objective of our study is to provide a SLAM system intended for wheeled mobile robots (Fig. 2). To achieve this, we modify the non-linear optimization pipeline, specifically in the Pose Optimization and Bundle Adjustment modules. Our approach is based on the methodology presented in [7]. Using the planar motion assumption, we reduce the estimated robot state vector from $SE(3)$ to $SE(2)$ and denote it as

$$\mathbf{x} = [\mathbf{R}, \mathbf{p}] \in \mathrm{SE}(2), \tag{1}$$

where $\mathbf{R} \in \mathrm{SO}(2)$ is the robot orientation, $\mathbf{p} \in \mathbb{R}^2$ is the robot position on a plane. Given a set of keyframes $\mathcal{K}$, which observe $\{\mathbf{l}_\mathrm{j}\}$ landmark from $\mathcal{L} \doteq [\mathbf{l}_0 \ldots \mathbf{l}_\mathrm{m}]$, and keyframe states $\mathcal{X} \doteq [\mathbf{x}_0 \ldots \mathbf{x}_\mathrm{n}]$, we iteratively update robot state vector $\mathbf{x}$, based on the following optimization problem, which is stated as maximum a-posteriori estimation (MAP):

$$\min_{\mathbf{x_i}} \Bigg(\sum_{i=1}^{n} \|\mathbf{r}_{\mathcal{O}\, i,i+1}\|^2_{\mathbf{\Sigma}^{-1}_{\mathcal{O}_{i,i+1}}} + \sum_{j=0}^{m-1} \sum_{i \in \mathcal{K}} \rho_a\Big(\|\mathbf{r}_{i,j}\|^2_{\mathbf{\Sigma}^{-1}_{i,j}}\Big)\Bigg) \tag{2}$$

where $\mathbf{r}_{i,j}$ is a visual residual (SubSection IV-B.1); $\mathbf{r}_{\mathcal{O}\, i,i+1}$ is a wheel odometry residual (SubSection IV-B.2); $\rho_a(\cdot)$ is the adaptive kernel (section IV-A). $\mathbf{\Sigma}_{\mathcal{O}_{i,i+1}}$ and $\mathbf{\Sigma_{i,j}}$ are covariance matrices representing noise density for wheel odometry and visual data respectively. MAP problem can be formulated as a factor graph optimization [20] on a sliding window of keyframes observing m landmarks. These observations formulate the re-projection error, while odometric edge is formulated between sequential keyframes. The optimization graph is shown in Fig. 3.

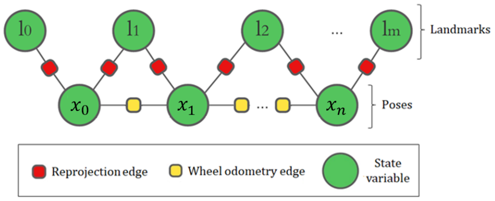

Fig. 3. Visual-Wheel fusion in the form of factor graph

A. Visual Odometry for Dynamic Environment Based on the Adaptive M-estimator

It is well-known that a negative impact of outliers and moving objects on optimization quality can be decreased by different outlier rejection techniques including M-estimators [21]. In the SLAM field, the standard choice for outlier rejection is usage of Huber Loss, which down-weights outliers impact, but still sensitive to large errors, which is the case for moving objects presence. Hence, we introduce the adaptive M-estimator, which is based on Barron Loss function [22]. For j-th landmark observed by i-th frame, the adaptive M-estimator can be derived as follows:

$$\rho_a(e_{ij}, \alpha, c) = \frac{|\alpha - 2|}{\alpha} \cdot \Bigg(\Big(\frac{(\frac{e_{ij}}{c})^2}{\alpha - 2} + 1\Big)^{\frac{\alpha}{2}} - 1\Bigg), \tag{3}$$

where $e_{ij} = \|\mathbf{r}_{i,j}\|^2_{\mathbf{\Sigma}^{-1}_{i,j}}$ is the re-projection error (see 2); α is a parameter, establishing a *shape* of the function (see Fig. 4b); c is a parameter, which determines the width of the loss function. In our implementation we set $c = 1$.

A useful property of the function (3) is that it generates any standard loss, depending on α parameter. This is shown in Fig. 4b. Generating of different losses down-weights outlier impact in an adaptive manner in comparison with Huber Loss. A practical advantage of the adaptive loss is that there is no need to manually set a loss function based on prior assumption about an outlier distribution, which may variate depending on the scene. Instead, the loss can be adjusted adaptively in run-time.

As demonstrated in [23], this approach is able to significantly improve the localization accuracy in the presence of moving objects in the scene. However, if there is no additional sources of prior knowledge about the moving objects, then the α parameter has to be obtained as a result of the alternating optimization, which have such disadvantages as slow convergence and sensitivity to an initial guess.

Our approach is to directly configure the adaptive M-estimator by adjusting the *shape* parameter. We introduce a landmark reliability model, which utilizes both semantic prior information about a visual landmark and re-projection error for a corresponding pixel on an image plane. Thus, the

α parameter is directly defined based on a reliability of a landmark during the tracking process. Here, by the landmark reliability we mean its probability to be static.

The reliability model can be described by the following equation:

$$P_r = 1 - (\omega_s P_s + \omega_o P_o), \tag{4}$$

where P_r is the landmark reliability; P_s is the landmark movement probability, which is based on semantic information and updated in the tracking process; P_o is the outlier probability based on re-projection error; ω_i are the weights on each component.

The landmark movement probability is calculated by the following equation:

$$P_s = \eta P_{prior}(\lambda_1 \bar{P}_s + \lambda_2(1 - \bar{P}_s)), \tag{5}$$

where $\bar{P}_s$ is a landmark movement probability calculated from the last frame observed the landmark; P_{prior} is a prior movement probability based on semantic label corresponding to the landmark; λ_1, λ_2 are coefficients determining the probability propagation rate; η is a normalization term. P_{prior} is pre-defined for each semantic label, which is provided by the semantic segmentation neural network, as we described in III-D. For example, we use $P_{prior} = 0.95$ for a "person" class.

The outlier probability is calculated by the following derivation:

$$P_o = \begin{cases} \frac{d_{ij}}{d_{max}} & \textit{if} \quad d_{ij} \leq \quad d_{max} \\ 1 & \textit{otherwise} \end{cases} \tag{6}$$

$$d_{ij} = |e_{ij} - e_{th}|, \quad d_{max} = |e_{max} - e_{th}| = const,$$

where e_{ij} is an actual re-projection error for j-th landmark observed by i-th frame; e_{th} is a re-projection error threshold; e_{max} is a maximum allowable re-projection error. For choosing e_{max} we analysed re-projection error distribution generated by ORB-SLAM3 visual odometry on eight *fr3* sequences of TUM RGB-D [4] dataset (category Dynamic Objects).

Finally, the *shape* parameter α dependency on the reliability of a landmark is quadratic and designed as follows:

$$\alpha = \begin{cases} a\bar{P}_r^2 + b\bar{P}_r + c & \textit{if} \quad \bar{P}_r \leq \quad P_h \\ -\infty & \textit{otherwise} \end{cases}, \tag{7}$$

where $\bar{P}_r = 1 - P_r$ is a landmark "unreliability"; P_h is a truncation threshold; coefficients determine the form of the functional dependency. The threshold is set to sharply down-weight highly unreliable landmarks. The set of parameters is chosen such that when a landmark is considered to be reliable with $P_r = 1$, a standard Huber Loss ($\alpha = 1$) is used in the state estimation process. When a landmark is considered as not reliable, α parameter gets infinitely small and Tukey Loss is used, strictly rejecting the unreliable visual landmark, that eliminates its impact on the state estimation. In our implementation we use the following setting: $P_h = 0.9; a = -8; b = -2; c = 1$.

B. Tightly-Coupled Visual-Wheel Fusion Based on the Planar Motion Assumption

We perform visual-wheel fusion by exploiting the planar motion assumption, applied to wheeled robots. Hence, robot state vector as well as all the constraints in our non-linear optimization are parameterized in $SE(2)$ space. Here, we describe the constraints utilized to formulate equation 2 in the back-end optimization process in RVWO.

1) The visual re-projection constraint parameterized on $SE(2)$: This constraint represents the visual residual between an observed 2D feature u and 2D projected point $\mathbf{u}\left(\ell_{\mathbf{j}}\right)$ from the corresponding landmark ℓ in 3D world frame w.r.t robot body frame $[\mathbf{R}_i|\mathbf{p}_i]$ and then w.r.t image plane C:

$$\mathbf{r}_{i,j} = \mathbf{u}\left(\ell_{\mathbf{j}}\right) = \Pi(\mathbf{R}_{CB}\mathbf{R}_i^T\left(\ell_{\mathbf{j}} - \mathbf{p}_i\right) + \mathbf{p}_{CB}) + \eta_u, \tag{8}$$

where $\eta_u \sim \mathcal{N}\left(\mathbf{0}, \sigma_u^2 \mathbf{I}_2\right)$ is a re-projection uncertainty (σ_u is the covariance from ORB pyramid level from which the feature is extracted) and $[\mathbf{R}_{CB}|\mathbf{p}_{CB}]$ is calculated from extrinsic calibration of the camera w.r.t. the body frame. $\Pi()$ is a projection function, which depends on a camera model.

The feature-based $SE(2) - XYZ$ constraint [7] benefits from encapsulating the $out{-}of{-}SE(2)$ motion perturbation and directly parameterizing the robot's poses on $SE(2)$.

The $out - of - SE(2)$ motion [7] includes two parts: the translation perturbation along Z axis as $\eta_z \sim \mathcal{N}\left(\mathbf{0}, \sigma_z^2\right)$ and the rotation perturbation in xy plane as $\eta_{xy} \sim \mathcal{N}\left(\mathbf{0}_{2\times1}, \Sigma_{\theta_{xy}}\right)$.

Therefore, the pose after applying perturbation on $SE(2)$ can be written as $[\tilde{\mathbf{R}}_i|\tilde{\mathbf{p}}_i]$ where:

$$\tilde{\mathbf{R}}_i = \mathbf{R}_i\, \mathrm{e}^{(\eta_\theta)}, \quad \tilde{\mathbf{p}}_i = \mathbf{p}_i + \eta_z \tag{9}$$

Then the projection equation (8) becomes

$$\begin{aligned} \mathbf{u}\left(\ell_{\mathbf{j}}\right) &= \Pi\left(\mathbf{R}_{CB}\tilde{\mathbf{R}}_i^T\left(\ell - \tilde{\mathbf{p}}_i\right) + \mathbf{p}_{CB}\right) + \eta_u \\ &\approx \Pi\left(\ell_{C_i}\right) + \mathbf{J}_{\eta_\theta}\mathbf{u}_\theta\eta_\theta + \mathbf{J}_{\eta_z}^{\mathbf{u}}\eta_z + \eta_u \\ &= \Pi\left(\ell_{C_i}\right) + \delta\eta_u, \end{aligned} \tag{10}$$

where $\delta\eta_u$ is a synthetic zero-mean noise. The noises η_θ, η_z and η_u are independent. Here we use first-order approximation to linearize the noise terms.

2) The wheel odometry constraint parameterized on $SE(2)$: Inspired by [7] we perform the preintegration of wheel encoder measurements on $SE(2)$. From the motion model for the wheel encoder, we get the relative robot body poses between consecutive odometry readings. Hence, the preintegrated odometry measurement is formulated as follows:

$$\begin{aligned} \phi_i &:= \tilde{\phi}_i - \delta(\phi_i), \\ \mathbf{p}_i &:= \tilde{\mathbf{p}}_i - \delta(\mathbf{p}_i), \end{aligned} \tag{11}$$

where ϕ_i is a yaw angle; p_i is a 2D translation vector; $\tilde{(\cdot)}$ denotes raw measured quantity; $\delta(\cdot)$ are corresponding noises. The propagation of the wheel odometry measurement

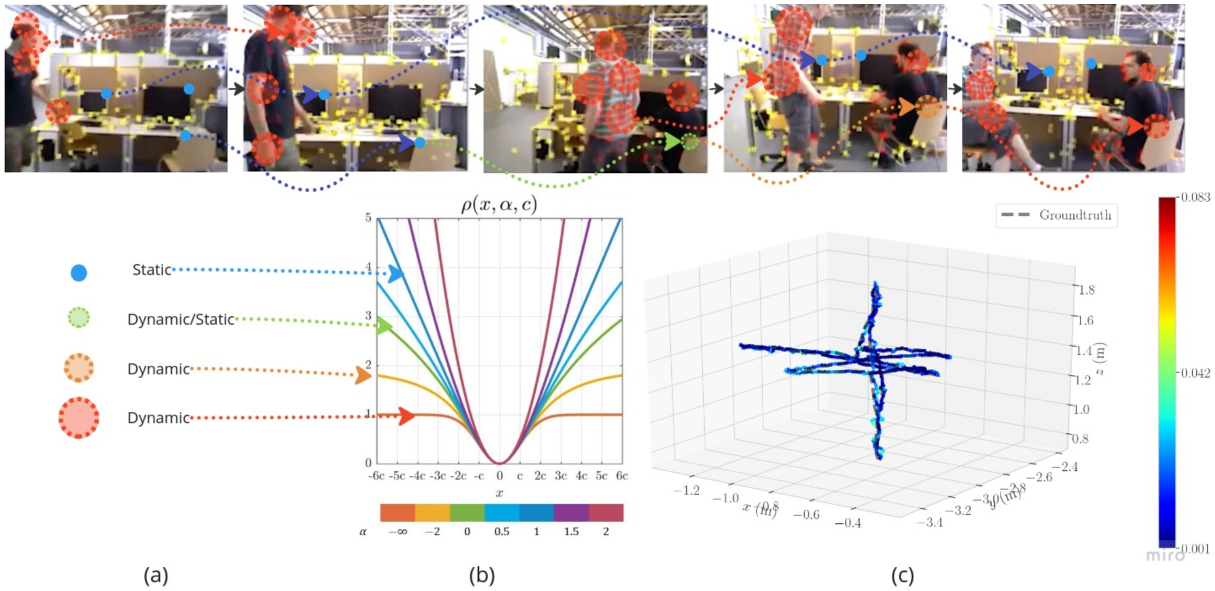

Fig. 4. (a) Color indicate the *alpha* paramter chosen for a landmark, (b) Set of standard losses derived from the Barron Loss [22], (c) The trajectory generated by RVWO on TUM RGB-D *fr3/w/xyz* sequence compared to groundtruth

can be written in a compact form:

$$\begin{bmatrix} \delta(\mathbf{p}_{i+1}) \\ \delta(\phi_{i+1}) \end{bmatrix} := \delta(\nu_{i+1}) = \mathbf{A}_i \delta(\nu_i) + \mathbf{B}_i \eta_{\nu i},$$

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{I}_2 & \Phi\left(\tilde{\phi}_i\right) 1^{\times} \tilde{\mathbf{p}}_i \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{B}_i = \begin{bmatrix} \Phi\left(\tilde{\phi}_i\right) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{12}$$

where $\mathbf{A}_i$, $\mathbf{B}_i$ are Jacobians of the motion model by state vector and noise vector, respectively.

The covariance $\Sigma_{\delta(\nu_{i+1})}$ of a wheel odometry measurement $\delta(\nu_i)$ is propagated as follows:

$$\Sigma_{\delta(\nu_{i+1})} = \mathbf{A}_i \Sigma_{\delta(\nu_i)} \mathbf{A}_i^T + \mathbf{B}_i \mathbf{\Sigma}_{\nu i} \mathbf{B}_i^T, \tag{13}$$

where $\mathbf{\Sigma}_{\nu i}$ represents covariance matrix of the wheel odometry noise.

Finally, the wheel odometry constraint representing an error function of a preintegrated wheel odometry measurement can be written as follows:

$$\mathbf{r}_{\mathcal{O}i,i+1} = \begin{bmatrix} \Phi\left(-\phi_i\right)\left(\mathbf{p}_{i+1} - \mathbf{p}_i\right) \\ \phi_{i+1} - \phi_i \end{bmatrix} - \begin{bmatrix} \tilde{\mathbf{p}}_{i+1} \\ \tilde{\phi}_{i+1} \end{bmatrix} \tag{14}$$

We incorporate the resulting error function (14) as an $SE(2)$ wheel odometry edge in our factor graph.

Therefore, we formulate equation 2, from both equations 8 and 14.

V. Evaluation

In order to ensure reliable odometry and localization for wheeled robots, we conducted a thorough evaluation across multiple stages:

A. First, we evaluated the impact of the adaptive M-estimator, without wheel fusion, by using camera measurements only. We integrated our proposed adaptive M-estimator into the visual ORB-SLAM3 system and tested it on the TUM RGB-D dataset [4], which contains dynamic objects.

B. Second, in Fig. 2 and 3 in [26], the authors demonstrated that many recent SLAM systems are not yet suitable for use on wheeled service robots. To validate our system on a publicly available benchmark, we evaluated RVWO on the open-source OpenLoris dataset.

C. Third, we conducted further evaluation of RVWO on our own mobile robot, illustrated in Fig. 2. We aim to assess the robustness of our system in various challenging conditions, such as lack of optical features, immediate change in illumination, existence of moving objects. We also hold the robot against its moving direction to induce wheel slippage.

To align all the trajectories w.r.t. groundtruth we used SE(3) Umeyama alignment method. In all the experiments we utilized the Absolute Trajectory Error (ATE), which provides a measure of the global consistency of the estimated trajectory. We also employed the Root Mean Square Error (RMSE), derived from the ATE, as an accuracy metric.

A. Evaluation of the Adaptive M-estimator on TUM RGB-D Dataset

To validate the efficiency of the proposed adaptive M-estimator, we selected sequences from the TUM RGB-D dataset [4]. This dataset consists of recorded data from a Kinect RGB-D sensor. For our evaluation, we used four sequences containing scenes with moving objects (marked *fr3/w*) and two static sequences (marked *fr3/s*). The *fr3/w/rpy* and *fr3/w(s)/xyz* sequences involve rotational and translational movements of the camera, respectively, in three degrees of freedom. The *fr3/w(s)/half* sequence involves both rotational and translational movements of the camera. The *fr3/w/static* sequence involves a stationary camera.

TABLE I

RESULTS OF THE ADAPTIVE M-ESTIMATOR EVALUATION ON TUM RGB-D. **BEST RESULT** IS HIGHLIGHTED IN BOLD, WHILE SECOND-BEST RESULT IS UNDERLINED.

Sequence	RMSE, m				
	ORB-SLAM2	Detect-SLAM	DS-SLAM	Dyna-SLAM	Ours
fr3/s/half	0.019	0.023	\	0.019	**0.017**
fr3/s/xyz	**0.009**	0.023	\	0.013	**0.009**
fr3/w/half	0.467	0.052	**0.030**	**0.030**	0.043
fr3/w/rpy	0.784	0.078	0.444	**0.035**	0.047
fr3/w/static	0.387	0.010	0.008	**0.007**	0.010
fr3/w/xyz	0.721	0.022	0.025	**0.016**	0.021

In Fig. 4a, we depict an illustration of several landmark probability by tracking them and visualizing the *shape* parameter used. It is seen that most landmarks located on persons are already outliers because of their semantic prior. We also note that α parameter value for such landmarks increases when the person stop moving (as noted by the radius of the red circles in Fig. 4a). For another static landmark, e.g. the landmark located on the chair upper edge to the bottom right of the images, it is clear that moving this chair affects the α parameter value for this particular landmark, as the movement causes an increase of a reprojection error.

An example of the generated trajectory is illustrated in Fig. 4c. We compared our approach with other state-of-the-art systems that were specifically designed to operate in dynamic environments. Mainly, we compare with DS-SLAM [14], Detect-SLAM [16] and Dyna-SLAM [11]. The evaluation results are presented in Table I. We marked by $\{\backslash\}$ the cells indicating no testing results available in public domain. As demonstrated in the table, our proposed solution exhibits robustness in scenes with moving objects, and it also achieves competitive results when compared to the aforementioned state-of-the-art approaches designed for operating in such environments. However, unlike Dyna-SLAM, our approach does not depend on the type of camera used and our solution can be applied to cheaper visual sensor setups. At the same time, we show best results in static sequences, which is achievable due to our adaptive outlier rejection technique. This allows to down-weight an impact of outliers caused not only by moving objects, but also by occlusions and motion blur.

B. Evaluation of RVWO on OpenLoris Benchmark

In [26], authors addressed scene changes caused by human activities, day-night shifts, and other factors. To capture these changes, multiple sequences were collected for each scene using several cameras, IMU, encoders, and LiDAR on a wheeled robot moving at human walking speed or slower in various environments. Our evaluation was performed on several sequences using the stereo fisheye RealSense T265 camera.

TABLE II

RESULTS OF EVALUATION ON OPENLORIS DATASET. **BEST RESULT** IS HIGHLIGHTED IN BOLD, WHILE {-} INDICATES FAILURE.

Sequence	RMSE, m			
	Wheel Odometry	ORB-SLAM3 (S-I)	VINS-Fusion (S-W)	RVWO (S-W)
office1-1	0.040	0.109	0.104	**0.020**
office1-3	0.043	–	0.037	**0.022**
office1-5	0.090	0.235	0.323	**0.081**
office1-7	0.072	0.069	0.064	**0.022**
home1-1	0.230	0.406	0.570	**0.221**
home1-2	0.314	0.363	0.470	**0.299**
home1-5	**0.066**	0.318	0.374	0.069
cafe1-1	0.236	**0.116**	0.324	0.213
cafe1-2	0.424	0.164	0.401	0.422
corridor1-3	0.222	0.990	0.525	**0.159**
corridor1-4	0.345	0.819	0.716	**0.235**
corridor1-5	0.640	1.131	0.504	**0.420**

We performed testing of the open-source systems ORB-SLAM3 in Stereo-Inertial (S-I) mode and VINS-Fusion[3] in Stereo-Wheel (S-W) mode.

Table II demonstrates that wheel encoders provide accurate odometry measurements. At the same time, S-I ORB-SLAM3 suffers from bigger drift and fails in some cases (as discussed in Section V-C).

From S-W VINS-Fusion[4] test results it can be seen that although this system performs fusion of visual and wheel odometry measurements, it still drifts more than pure wheel odometry in some cases. This indicates the negative impact of visual challenges in OpenLoris dataset on the fusion accuracy. In contrast, RVWO demonstrates robust and reliable performance in different scenes, showing that our proposed solution adapts to cases when visual tracking is unstable and when pure wheel odometry drifts.

C. Evaluation of RVWO on Custom Hardware Setup in Challenging Scenarios

In this section, we provide an analysis of experiments conducted on our custom sensor setup. Our setup includes stereo camera Zed2 with an internal IMU mounted on top of a wheeled mobile robot (see Fig. 2). The robot is equipped with internal wheel encoders that provide odometry measurements with a frequency of 30 *Hz*. For both intrinsic and extrinsic camera parameters calibration we used Kalibr toolbox [25]. The IMU noise density and bias random walk obtained by using Allan Variance, which is a statistical technique used to characterize the random noise in the IMU measurements. Intrinsic parameters of wheel encoders are calculated using the online calibration technique from [27]. The transformation matrix from the wheel odometer

[3] This implementation is the closest to VINS-on-Wheels [3]. It is built upon VINS-Fusion system [27]. see https://github.com/TouchDeeper/VIW-Fusion

[4] We manually inflated the wheel encoder noise values by factors of 0.1, 1, 10, 100 to find the best configuration.

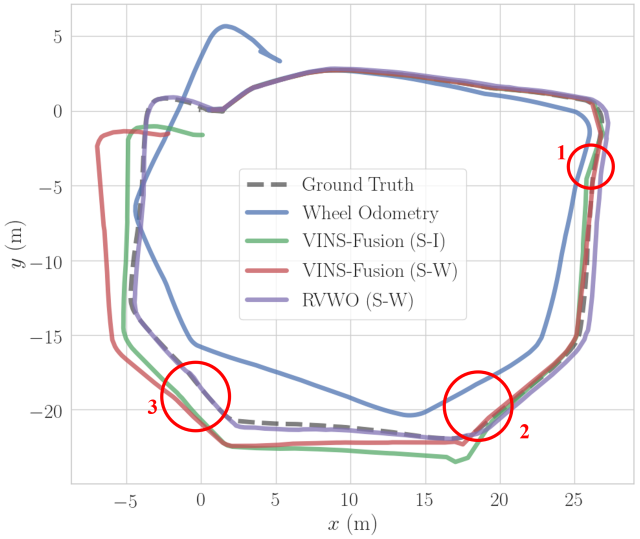

Fig. 5. Trajectories generated on *courier_4* sequence compared to groundtruth

TABLE III

RESULTS OF EVALUATION ON REAL-WORLD SEQUENCES RECORDED ON OUR SETUP, **BEST RESULT** IS HIGHLIGHTED IN BOLD, WHILE $\{-\}$ INDICATES FAILURE.

Sequence	RMSE, m				
	Wheel Odometry (W)	ORB-SLAM3 (S-I)	VINS-Fusion (S-I)	VINS-Fusion (S-W)	RVWO (ours) (S-W)
courier_1	2.899	0.383	**0.262**	0.369	0.421
courier_2	0.509	–	0.338	0.289	**0.140**
courier_3	0.415	–	0.180	0.248	**0.096**
courier_4	1.790	–	0.736	1.272	**0.278**

coordinate system to the camera coordinate system is computed by using the same approach[4]. Check our open source repository[2] for further details on calibration and sensor setup. We recorded four test sequences. In the first sequence, named *courier_1*, the robot navigates inside an office space. This sequence primarily involves straight movements with a constant acceleration. The second and third sequences, named *courier_2* and *courier_3*, present challenging conditions for both visual and wheel odometry. These conditions include moving persons, featureless surfaces, and wheel slippage cases. The fourth and most challenging sequence, named *courier_4*, includes scenes with moving objects, wheel slippage and closed camera situations. (see Fig. 7). These sequences are available in RVWO repository[2].

For each sequence, we used Cartographer [24] algorithm to generate groundtruth trajectory. This algorithm utilizes wheel odometry and 2D LiDAR data into an offline global optimization problem. It is important to note as seen in Table. III, that pure wheel odometry provided by our mobile robot is not as reliable as in [26] in general (see Fig. 5).

We evaluate the robustness of RVWO and other systems in specific challenging cases as follows:

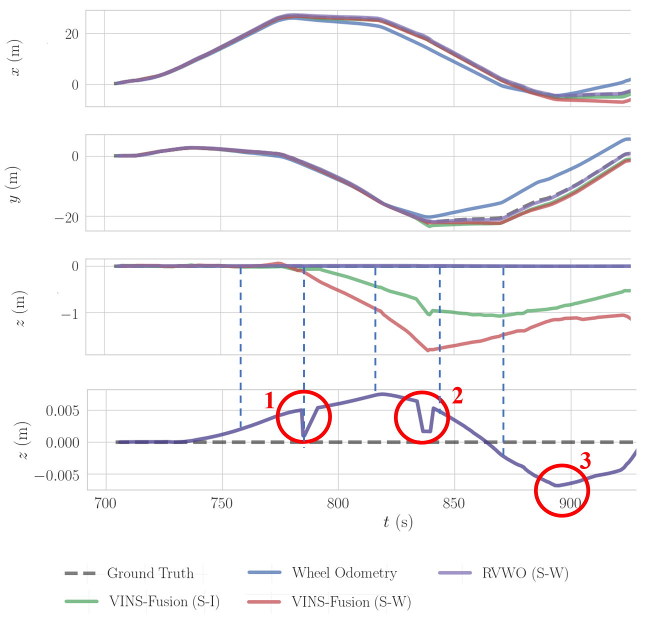

Fig. 6. Trajectories of RVWO, and other systems, generated on Courier_4 sequence compared to groundtruth. From Above, X, Y and Z respectively, then zoomed trajectory along Z-axis generated by RVWO.

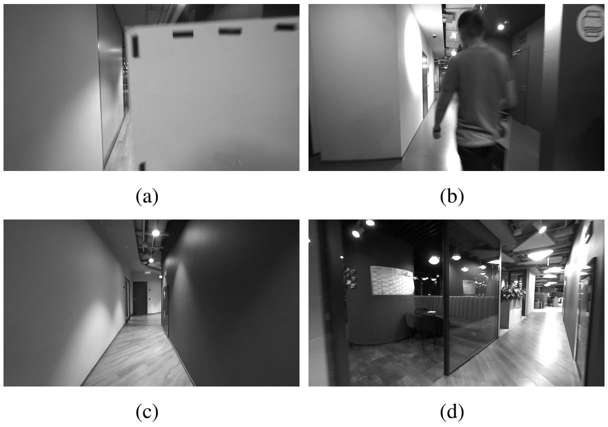
(a)
(b)
(c)
(d)

Fig. 7. Visual challenging conditions: (a) Closing the camera, (b) Moving objects, (c) Low visual texture, (d) Light reflection and high exposure

1) Visual challenging scenarios: Our sequences mostly consist of visual challenging situations, such as featureless surfaces (see Fig. 7-c), light reflection (see Fig. 7-d), or artificially closing the camera as in Fig. 7-a. Red circles marked as *1*, *2* in Fig. 5 represents moments when the robot encounters the full absence of visual features as in Fig. 7-a.

2) Wheel slippage: Red circle marked as *3* in Fig. 5 represents the case when we artificially stop the robot while moving (motors still rotating), causing a slippage and inducing wrong wheel odometry data to test the quality of visual-wheel fusion in such scenario.

Similarly to Section V-B, we test visual-inertial systems on our sequences to validate a performance of VI SLAM systems on service robots. We evaluate both VINS-Fusion and ORB-SLAM3 in Stereo-Inertial mode (see Table III). S-I

ORB-SLAM3 shows a poor performance on most sequences except for *courier_1*, while S-I VINS-Fusion shows better performance even in difficult sequences where there are challenging visual cases. In general, the performance of both systems here is questionable, because the convergence of inertial parameters to optimal values cannot be guaranteed in the case of planar motion [3]. Additionally, we show in Fig. 5 and 6 that S-I VINS-Fusion is drifting in the moments marked as *1* and *2*. This leads to accumulation of the drift along the overall trajectory.

We show in Table III, that poor wheel odometry affects visual-wheel fusion in S-W VINS-Fusion^3. This results in better S-I VINS-Fusion performance compared to S-W VINS-Fusion^3, despite problematic performance of visual-inertial setup in planar motion. Moreover, in *courier_4*, wheel slippage as in *3* in Fig. 5, and visual challenging cases as in *1*, *2*, result in large drift of S-W VINS-Fusion^3. It is important to note that although this system uses planar motion assumption, it drifts noticeably along Z-axis as depicted in Fig. 6 when facing these challenges.

In contrast, RVWO outperforms aforementioned systems (see Table III) with an average RMSE error around 25 cm on 70 m-long sequences with challenging visual and wheel odometry conditions. Moreover, we note in Fig. 6 that in visual challenging and wheel slippage situations, RVWO disturbance along Z-axis is minimal compared to other systems, as it is fluctuating around the zero value with a maximum magnitude of 5 mm, thus validating the beneficial impact of the planar motion assumption.

VI. Conclusion

This study introduces a novel robust visual-wheel SLAM system (RVWO) that leverages the planar motion assumption and adaptively filters outliers to improve pose estimation accuracy and robustness for service wheeled robots. Furthermore, prior semantic information about the environment is incorporated to enhance performance. The system's effectiveness is demonstrated through validation on open-source datasets and a series of experiments, which include a comparison with other fusion techniques using a basic sensor setup. Results indicate that RVWO provides state-of-the-art performance on a real service wheeled robot.

References

[1] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, & Juan D. Tardos (2021). ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. IEEE Transactions on Robotics, 37(6), 1874–1890.

[2] Qin, T., Li, P., & Shen, S. (2018). VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Transactions on Robotics, 34(4), 1004-1020.

[3] Wu, K., Guo, C., Georgiou, G., & Roumeliotis, S. (2017). VINS on wheels. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5155-5162).

[4] J. Sturm, N. Engelhard, F. Endres, W. Burgard, & D. Cremers (2012). A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proc. of the International Conference on Intelligent Robot Systems (IROS).

[5] Yang, D., Bi, S., Wang, W., Yuan, C., Wang, W., Qi, X., & Cai, Y. (2019). DRE-SLAM: Dynamic RGB-D Encoder SLAM for a Differential-Drive Robot. Remote Sensing, 11(4).

[6] Fan Zheng, Hengbo Tang, & Yun-Hui Liu (2019). Odometry-Vision-Based Ground Vehicle Motion Estimation With SE(2)-Constrained SE(3) Poses. IEEE Trans. Cybernetics, 49(7).

[7] Fan Zheng, & Yun-Hui Liu (2019). Visual-Odometric Localization and Mapping for Ground Vehicles Using SE(2)-XYZ Constraints. In Proc. IEEE Int. Conf. Robot. Autom (ICRA).

[8] Lee, W., Eckenhoff, K., Yang, Y., Geneva, P., & Huang, G. (2020). Visual-Inertial-Wheel Odometry with Online Calibration. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 4559-4566).

[9] Xingxing Zuo, Mingming Zhang, Yiming Chen, Yong Liu, Guoquan Huang, & Mingyang Li (2019). Visual-Inertial Localization for Skid-Steering Robots with Kinematic Constraints. CoRR, abs/1911.05787.

[10] Dang, Z., Wang, T., & Pang, F. (2018). Tightly-coupled Data Fusion of VINS and Odometer Based on Wheel Slip Estimation. In 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1613-1619).

[11] Bescos, B., Facil, J., Civera, J., & Neira, J. (2018). DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. IEEE Robotics and Automation Letters, 3(4), 4076–4083.

[12] L. Cui and C. Ma, ”SOF-SLAM: A Semantic Visual SLAM for Dynamic Environments,” in IEEE Access, vol. 7, pp. 166528-166539, 2019, doi: 10.1109/ACCESS.2019.2952161.

[13] Mur-Artal, R., & Tardos, J. (2017). ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics, 33(5), 1255–1262.

[14] Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q., & Fei, Q. (2018). DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[15] Zhao, X. (2021). OFM-SLAM: A Visual Semantic SLAM for Dynamic Indoor Environments. Mathematical Problems in Engineering, 2021, 5538840.

[16] Zhong, F., Wang, S., Zhang, Z., Chen, C., & Wang, Y. (2018). Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1001-1010).

[17] Forster, C., Carlone, L., Dellaert, F., & Scaramuzza, D. (2017). On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. IEEE Transactions on Robotics, 33(1), 1-21.

[18] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, & Ping Luo (2021). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. CoRR, abs/2105.15203.

[19] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, & Antonio Torralba (2016). Semantic Understanding of Scenes through the ADE20K Dataset. CoRR, abs/1608.05442.

[20] Dellaert, F., & Kaess, M. (2017). Factor Graphs for Robot Perception. Now Publishers.

[21] M-Estimators. (2008). In Introduction to Empirical Processes and Semiparametric Inference (pp. 263–282). Springer New York. https://doi.org/10.1007/978-0-387-74978-5_14

[22] Jonathan T. Barron (2017). A More General Robust Loss Function. CoRR, abs/1701.03077.

[23] Nived Chebrolu, Thomas Läbe, Olga Vysotska, Jens Behley, & C. Stachniss (2020). Adaptive Robust Kernels for Non-Linear Least Squares Problems. IEEE Robotics and Automation Letters, 6, 2240-2247.

[24] Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016). Real-time loop closure in 2D LIDAR SLAM. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1271-1278).

[25] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann and R. Siegwart, ”Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes,” 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 4304-4311, doi: 10.1109/ICRA.2016.7487628.

[26] Shi, X., Li, D., Zhao, P., Tian, Q., Tian, Y., Long, Q., Zhu, C., Song, J., Qiao, F., Song, L., Guo, Y., Wang, Z., Zhang, Y., Qin, B., Yang, W., Wang, F., Chan, R., & She, Q.. (2019). Are We Ready for Service Robots? The OpenLORIS-Scene Datasets for Lifelong SLAM.

[27] Qin, Tong Pan, Jie Cao, Shaozu Shen, Shaojie. (2019). A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors.