MOVRO2: Loosely Coupled Monocular Visual Radar Odometry using Factor Graph Optimization

Vlaho-Josip Štironja[a,*], Juraj Peršić[b], Luka Petrović[a], Ivan Marković[a], Ivan Petrović[a]

[a] *University of Zagreb Faculty of Electrical Engineering and Computing*
Laboratory for Autonomous Systems and Mobile Robotics (LAMOR)
Unska 3, HR-10000, Zagreb, Croatia

[b] *Calirad d.o.o.*
Augusta Harambašića 4, HR-10000, Zagreb, Croatia

Abstract

Ego-motion estimation is an indispensable part of any autonomous system, especially in scenarios where wheel odometry or global pose measurement is unreliable or unavailable. In an environment where a global navigation satellite system is not available, conventional solutions for ego-motion estimation rely on the fusion of a LiDAR, a monocular camera and an inertial measurement unit (IMU), which is often plagued by drift. Therefore, complementary sensor solutions are being explored instead of relying on expensive and powerful IMUs. In this paper, we propose a method for estimating ego-motion, which we call MOVRO2, that utilizes the complementarity of radar and camera data. It is based on a loosely coupled monocular visual radar odometry approach within a factor graph optimization framework. The adoption of a loosely coupled approach is motivated by its scalability and the possibility to develop sensor models independently. To estimate the motion within the proposed framework, we fuse ego-velocity of the radar and scan-to-scan matches with the rotation obtained from consecutive camera frames and the unscaled velocity of the monocular odometry. We evaluate the performance of the proposed method on two open-source datasets and compare it to various mono-, dual- and three-sensor solutions, where our cost-effective method demonstrates performance comparable to state-of-the-art visual-inertial radar and LiDAR odometry solutions using high-performance 64-line LiDARs.

Keywords: monocular visual odometry, radar odometry, visual radar odometry, multisensor fusion, factor graph optimization

1. Introduction

In an effort to develop robust and versatile autonomous systems, it is essential to combine information from multiple sensors with complementary modalities while reducing costs. Depending on the specific application and operating environment of a mobile robot or autonomous vehicle, different combinations of sensors are used. For example, in outdoor applications, the Global Navigation Satellite System (GNSS) is often used as a crucial component of the sensor package for ego-motion estimation, while in indoor applications, visual inertial odometry is often used due to the unavailability and unreliability of GNSS data. Nowadays, the sensor systems of mobile robots mostly consist of monocular cameras, LiDARs and inertial measurement units (IMU). However, the integration of information from inexpensive IMUs often leads to additional drift in the estimation of the robot's ego-motion, while integration of high-performance IMUs can be cost-ineffective.

Recent advances in millimeter-wave radar technology have led to a growing number of multi-sensor position estimators that integrate radar information [1, 2, 3]. The integration of radar measurements improves the robustness of the estimation, which is particularly important in difficult weather and light conditions such as fog, smoke and heavy snowfall. In addition, radar enables the estimation of ego velocity from a single scan by measuring Doppler velocity, avoiding the often error-prone scan matching procedure. Combined with its robustness to light conditions, long range, low cost and light weight, the ability to estimate ego velocity makes radar an attractive alternative to other sensors in various robotic platforms [4]. However, radars face challenges such as relatively high noise compared to other sensors, ghost targets and multi-path reflections.

Monocular cameras are widely used sensors because they provide a wealth of information, are relatively inexpensive and have high accuracy. However, most monocular visual odometry methods suffer from scale ambiguity. Deep learning methods are applied to monocular visual odometry to solve the above problem by using stereo sequences as ground truth data in the training dataset [5], but this comes at an additional cost for data acquisition as training data is not always readily available. Therefore, cameras are commonly paired with IMUs, which mitigate the problem of scale ambiguity and improve the overall accuracy of ego-motion estimates obtained with monocular visual-inertial odometry.

Information from cameras and radar is usually combined for

*Corresponding author: Vlaho-Josip Štironja
Email addresses: vlaho-josip.stironja@fer.unizg.hr (Vlaho-Josip Štironja), juraj.persic@calirad.net (Juraj Peršić), luka.petrovic@fer.unizg.hr (Luka Petrović), ivan.markovic@fer.unizg.hr (Ivan Marković), ivan.petrovic@fer.unizg.hr (Ivan Petrović)

object detection and tracking [6, 7], using the valuable object depth information from the radar and the extensive semantic information from the camera. However, cameras and radars are rarely used together for ego-motion estimation. Monocular visual odometry can provide accurate orientation estimates and allows to achieve global consistency by closing loops, but it lacks metric scale. On the other hand, radar can accurately estimate instantaneous ego velocity but cannot provide accurate orientation estimates. This complementary nature of cameras and radars offers significant potential for the study of ego-motion estimation in a monocular visual radar system.

In this paper, we propose a novel MOnocular Visual and 3D Radar Odometry method, dubbed MOVRO2, based on Factor Graph Optimization (FGO) using a loosely coupled approach. MOVRO2 integrates ego-velocity estimates and scan-to-scan matches from radar measurements with rotation from consecutive camera frames and unscaled ego-velocity estimates obtained from monocular visual odometry. The loosely coupled approach to sensor fusion allows the scalable integration of independently obtained solutions for any underlying task, i.e., any monocular visual odometry or scan matching method can be used in our framework. MOVRO2 makes no assumptions about the environment and does not require any other sensors (e.g., a barometer) besides a monocular camera and a radar.

The MOVRO2 method improves our original approach presented in [8] by adding the scan-to-scan radar matches and by formulating the monocular visual radar odometry as FGO, motivated by the exhibited advantages of FGO-based methods for inertial-GNSS fusion compared to the Extended Kalman Filter (EKF) [9]. In addition, we include precise monocular odometry orientation estimates in MOVRO2 as the rotation between two steps. This enhances the system's robustness and effectively handles the reinitialization of monocular odometry. We compared the proposed method with different monocular and multi-sensor odometry methods, such as ROVIO [10], EKF-RIO [11], RVIO [2], 4DRVO-Net [3], and CMFlow [12], using IRS [2] and VoD [13] datasets that feature two radars with different accuracies. For radar ego velocity estimation, we use the open-source packages *reve* [14] and RAVE [15], while for the monocular odometry we use both OV2SLAM [16] and ORB-SLAM3 [17] without loop-closures. The results demonstrate how the use of the additional radar scan matching factors affects the accuracy of ego-motion estimation, and emphasize the importance of understanding the limitations of the radar in multi-sensor setups. Furthermore, we demonstrate the modularity of the loosely coupled approach by testing the proposed framework with both OV2SLAM and ORB-SLAM3 methods, showing its adaptability to future state-of-the-art methods.

The rest of the paper is organized as follows. Section 2 reviews the related work in the areas of monocular visual and radar-based odometry. Section 3 presents the proposed method, followed by the experimental results on real-world datasets in Section 4. Finally, we summarize our results, present our conclusions and discuss future work in Section 6.

2. Related work

In this section we provide an overview of related work on radar-based, monocular visual, and monocular visual radar ego-motion estimators.

2.1. Radar-based Ego-Motion Estimators

The instantaneous ego-velocity obtained from a single radar scan has been proven to be very robust and accurate in automotive scenarios [18]. Recently, the implementation of 3D radar inertial velocity estimation using sliding window approaches [19] has achieved comparable accuracy to visual inertial approaches under conditions favorable to visual inertial odometry, while also performing well in visually degraded environments where monocular odometry often fails. Doer et al. [14] fused the 3D radar's ego-velocity estimates within a single scan with the IMU, and integrated a barometer into the Error State Extended Kalman Filter (ES-EKF) to reduce drift. Their subsequent work [11] incorporates online estimation of the radar extrinsic calibration, enhancing the system accuracy without the need for tedious manual extrinsic calibration. Michalczyk et al.[20] presented a novel 3D tightly-coupled EKF-based radar inertial odometry method that uses an additional correction with range measurements to reduce odometry drift. The information about the measured distance to a point proved to be beneficial as it does not contain the point angle information, which can be noisy in low-cost Frequency Modulated Continuous Wave (FMCW) radars. In their extended work [1], the final drift was significantly reduced by tracking persistent landmarks across past robot poses using a lightweight and low-cost FMCW radar sensor. The authors also demonstrated that the inclusion of additional extrinsic parameters in the state (online calibration) improves the accuracy and consistency of the system, and presented the first closed-loop flight of an unmanned aerial vehicle (UAV) using radar inertial odometry [21]. DRIO [22] proposed a novel radar inertial odometry method improved by adopting the ground point which enables accurate localization in highly dynamic environments. The first 3D radar inertial odometry and mapping was presented in [23], using the iterative EKF. Instead of using Random Sample Consensus (RANSAC) for the outlier rejections step, the authors used a graduated non-convexity method to robustly and efficiently estimate the instantaneous ego-velocity.

The first 3D Pose Graph SLAM methods relying on radars were recently presented in [24, 25], with both works leveraging scan-matching and the ego-velocity estimation capabilities contained in the pose graph. The authors in [26] compared the error-prone radar scan-to-scan and scan-to-map registration algorithms with the integration of Doppler/IMU data and EKF sensor fusion, with sensor fusion leading to a similar or better result than 3D point matching. However, the authors do not address to consider both approaches for the odometry solution, while the fusion of scan-to-map or scan-to-scan with ego-velocity estimation could lead to better results, as shown in [1]. The state-of-the-art 3D radar visual/thermal inertial odometry is presented in [2]. By extending the visual inertial odometry method ROVIO [10] with the drift-free 3D radar ego-velocity,

the authors achieved higher accuracy compared to the dual-domain visual inertial and radar inertial estimators. The authors in [27] implemented a loosely coupled fusion that incorporates longitudinal radar velocity, height, optical flow, GPS, and IMU data into an EKF. Similarly, [28] presents a loosely coupled fusion concept based on the ES-EKF that incorporates 2D radar velocity and data from a stereo pair of cameras, LiDAR, and IMU.

Advances in deep learning have led to novel radar odometry methods, such as RaFlow [29] and CMFlow [12]. RaFlow is the first 3D scene-flow estimation method that does not depend on costly annotations, but relies on self-supervised learning. Similarly, CMFlow [12] integrates cross-modal learning to estimate odometry, segmentation, and scene flow without additional annotations, outperforming RaFlow in accuracy. CMFlow utilizes the complementary supervision signals from heterogeneous sensors (e.g., a camera, an odometer, a LiDAR) to constrain the model outputs.

2.2. Monocular Visual Ego-Motion Estimators

Visual ego-motion estimators are typically divided into feature-based and direct methods [30]. The general idea of feature-based methods is to detect and match 2D features in consecutive images. By reprojecting 2D points into 3D space, a corresponding relationship between the images is established. For example, the open-source packages ORB-SLAM3 [17] and OV2SLAM [16] are both feature-based monocular visual estimators. ORB-SLAM3 was the first visual odometry that was able to reuse all stages of the method and all prior information from highly-parallax, co-visible keyframes (even temporally distant from each other), which greatly increased accuracy [17]. The authors in [16] presented the first visual SLAM method that integrates Bag of Words, focusing on real-time performance without sacrificing accuracy and robustness. On the other hand, direct visual methods [31, 32]) developed from optical flow estimate motion directly from the spatial localization of pixels. This implies the minimization of the photometric error, i.e., the reprojection error of feature points in the feature point method, without extracting the feature or computing the feature description [30]. The SVO algorithm [33] combines the advantages of feature-based and direct visual estimators to increase the accuracy of tracking and mapping. More recent advances in monocular visual odometry leverage deep learning methods, e.g., by incorporating the monocular depth estimation of the scene obtained from deep models trained in a self-supervised manner with stereo image pairs [5, 34].

2.3. Monocular Visual Radar Ego-Motion Estimators

MonoRAD is a decoupled 2D ego-motion estimation method that relies on the Doppler measurement of a single radar for translation estimation and on the monocular camera image for rotation estimation [35]. The authors pointed out the advantage of using visual radar estimators in a loosely coupled manner, as this provides the opportunity to easily integrate new state-of-the-art techniques into the pipeline. Following a similar idea, the authors in [36] integrate instantaneous 2D radar ego-velocity estimates into ORB-SLAM key-frames to improve scale estimation. However, the drawback of this method is that it uses velocity estimates to scale the monocular visual odometry only after the loop is closed, which is not possible in online applications. In [37], the authors assume that the elevation filed of view (FOV) of the radar is narrow enough to obtain the 2D radar ego-velocity estimates using scan matching, which are then fused with the monocular visual odometry. The main drawback of the method is the lack of integration of instantaneous ego-velocity estimates, as it relies solely on radar scan matching, which can be prone to errors due to low resolution, multipath reflections, relatively high noise, and often small number of targets per image. MOVRO is the first 3D monocular visual radar odometry implemented within the ES-EKF framework in a loosely coupled manner [8]. MOVRO takes into account the unscaled velocity of the monocular odometry in addition to the estimates of the ego velocity of the radar and the monocular odometry orientation, which reduces the translational drift due to the scaling factor. However, MOVRO does not address monocular odometry failures and reinitialization, as it only corrects the state using orientation data. The first deep learning approach for visual radar odometry was presented in [3], in which a multi-sensor feature extraction network called Radar-PointNet++ was proposed. The authors integrate two sensing modalities using multiscale cross-modal feature interaction and adaptive multi-sensor feature fusion and introduce velocity-driven point confidence estimation, which achieves high accuracy comparable to LiDAR odometry with a lower-cost system. However, the method has been trained on the specific training data and the ability to generalize to other environments has not been demonstrated. To mitigate the risk of poor performance in unknown environments due to model overfitting, the method would need to be retrained with manually collected data, which can be time-consuming. We believe that our approach, utilizing a loosely coupled system fusion, is scalable and adaptable to various datasets. This method does not require model training and enhances the robustness of MOVRO by incorporating scan-to-scan matches from radar and treating rotation estimates as transitions between states rather than standalone orientation measurements accounting for monocular odometry failures.

3. MOVRO2 method

The proposed method, dubbed MOVRO2, estimates odometry by a loosely coupled radar-camera sensor fusion and factor graph optimization based approach. There are no restrictions regarding environmental conditions (e.g. presence of a ground plane) or the fact that the sensors must have the same timestamps. The state is estimated using the ego-velocity and the transformation matrix of the radar calculated with the scan registration method, as well as the rotation and the unscaled ego-velocity from the monocular visual odometry. Following our original MOVRO method [8], we take advantage of the scaling factor to reduce translation errors. Original MOVRO estimates the state with a loosely coupled Error state Extended Kalman Filter. The state is propagated using the constant velocity model

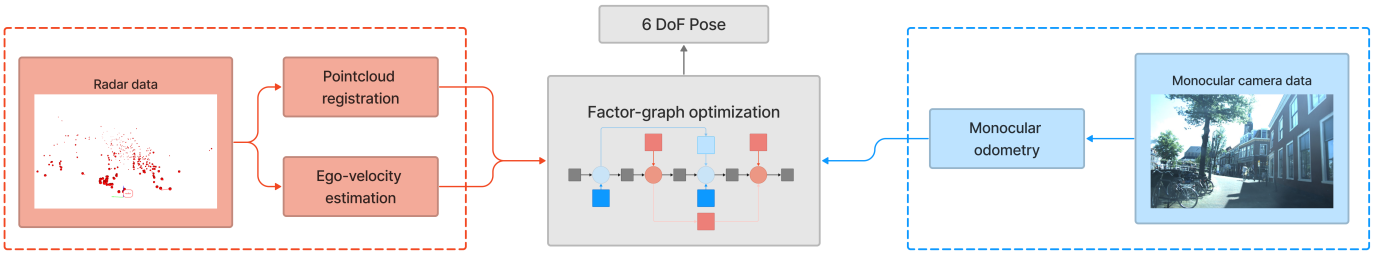


Figure 1: The architecture of the MOVRO2 method. The proposed factor graph takes as inputs the monocular odometry data, the radar ego-velocity estimates and the result of radar pointcloud registration, all of which can be obtained with any existing method from raw radar and monocular camera data. We combine the rotation and unscaled ego-velocity of monocular visual odometry with the transformation matrix of radar using scan matching and ego-velocity estimation. The 6 DoF odometry estimate is obtained by optimizing the described factor graph.

and corrected with ego-velocity measurements from the monocular odometry and radar as well with the orientation from the monocular odometry. To improve robustness and reliability, MOVRO2 implements the FGO framework for sensor fusion, instead of the EKF framework used in MOVRO, and utilizes the radar's scan registration capabilities, and treats the precise rotation estimates of the monocular odometry as transitions between states rather than standalone orientation measurements, thereby accounting for monocular odometry failures.

The architecture of the proposed MOVRO2 method is shown in Fig. 1. The proposed factor graph takes as inputs the monocular odometry, the radar ego-velocity estimates and the result of radar scan registration. The key feature of MOVRO2 is modularity, meaning that each input can be obtained with any existing method from raw radar and monocular camera measurements. We combine the rotation and unscaled ego-velocity of monocular visual odometry with the transformation matrix of radar using scan matching and ego-velocity estimation. The 6 DoF odometry estimate is obtained by optimizing the described factor graph. The MOVRO2 method is derived step by step below.

3.1. Notation, state definition and assumptions

The following notation is used in this paper. Scalars are written in non-bold lowercase letters, e.g., m. Constants are written in non-bold capital letters, e.g., M, and matrices in bold capital letters, e.g., $\mathbf{M}$. Vectors are written in bold lower case letters, e.g. $\mathbf{m}$. The spelling of rotation and vector is taken from [38]. The expression ${}_{\mathrm{A}}\mathbf{t}_{\mathrm{BC}}$ stands for the vector from frame $\mathcal{F}_{\mathrm{B}}$ to $\mathcal{F}_{\mathrm{C}}$, expressed in $\mathcal{F}_{\mathrm{A}}$. The rotation matrix $\mathbf{R}_{\mathrm{AB}}$ is a rotation of the frame $\mathcal{F}_{\mathrm{B}}$ with respect to $\mathcal{F}_{\mathrm{A}}$, i.e., $\mathbf{R}_{\mathrm{AB}}$ transforms a vector ${}_{\mathrm{B}}\mathbf{v}$ from $\mathcal{F}_{\mathrm{B}}$ to $\mathcal{F}_{\mathrm{A}}$ (${}_{\mathrm{A}}\mathbf{v}$).

The system state at step k is defined as:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{R}_{\mathrm{WC},k} & {}_{\mathrm{C}}\boldsymbol{\omega}_k & {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k} & {}_{\mathrm{W}}\mathbf{v}_k & r_k \end{bmatrix}, \tag{1}$$

where $\mathbf{R}_{\mathrm{WC},k}$, ${}_{\mathrm{C}}\boldsymbol{\omega}_k$, ${}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k}$, ${}_{\mathrm{W}}\mathbf{v}_k$, and r_k stand for orientation, angular velocity, position, camera velocity, and scale factor, respectively. With each measurement arrival, an additional system step is added to the system. Extrinsic parameters, i.e. the camera to radar transformation, consist of the translation segment ${}_{\mathrm{C}}\mathbf{t}_{\mathrm{RC}}$, and the orientation segment $\mathbf{R}_{\mathrm{RC}}$ which are assumed to be apriori known. The frames $\mathcal{F}_{\mathrm{C}}$ and $\mathcal{F}_{\mathrm{W}}$ stand for the camera frame and world frame, respectively. The world frame is initialized in the first step of the proposed method. The orientations are represented as rotation matrices $\mathbf{R} \in \mathrm{SO}(3)$ [39].

We assume that the mobile robot is stationary at startup, therefore position, angular, and translational velocity are initialized to zero, while the scaling factor is set to an initial value of 1. The uncertainty of the initial state is retained as a tuning parameter. Since some sequences start with motion, the velocity noise is increased. The extrinsic calibration between camera and radar is assumed to be known.

3.2. Factor graph model

Typically, multi-sensor integration, which can be formulated in a factor graph, aims to estimate the optimal posterior state using the maximum a posteriori (MAP) problem [39]. A factor graph is an undirected graph that contains two types of nodes: variable nodes and factor nodes [40]. The variable represents the state that is the target of the estimation, while the factors contain probabilistic information about the state, derived from the measurements or prior knowledge [41]. The factor graph can be modeled as a MAP estimation problem, assuming that all measurements conditionally are independent:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \prod_{k,i} p(\mathbf{z}_k|\mathbf{x}_k) \prod_{k} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k), \tag{2}$$

where $\mathbf{x}_k$, $\mathbf{z}_k$, and $\mathbf{u}_k$ represent the state, measurements and control inputs at time k, while $\hat{\mathbf{x}}$ represents the optimal system state set [39]. Since a factor node ϕ represents a probability density function (e.g. $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$), Eq. (2) can be rewritten as follows:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \prod_{k} \phi_k(\mathbf{x}_k). \tag{3}$$

Furthermore, we define the measurement as $\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k$, where $\mathbf{h}(*)$, and $\mathbf{v}_k$ stand for the measurement function and the measurement noise. Using the Gaussian distribution with the zero mean covariance matrix $\sum_k$ results in the following:

$$\phi_k(\mathbf{x}_k) \propto \exp\left\{-\frac{1}{2}\|\mathbf{z}_k - \mathbf{h}_k(\mathbf{x}_k)\|^2_{\sum_i}\right\}. \tag{4}$$

Assuming that all nodes are independent and the sensor noise corresponds to a Gaussian distribution with zero mean, we can rewrite Eq. (3) as follows ([42]):

$$\hat{\mathbf{x}} = \underset{\mathrm{x}}{\operatorname{argmin}} \sum_{k} \|\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}_i)\|^2_{\sum_i} \tag{5}$$

In contrast to recursive Bayesian filters, which continuously calculate estimates based on the previous state, control inputs and observation matrix, while FGO estimates states in a batch fashion. The FGO converts Eq. (2) into a nonlinear least squares minimization problem (Eq. (5)) that optimizes the optimal system state and minimizes the sum of error functions. Eq. (5) can be effectively solved by applying a linearized formulation of the underlying problem. Alternatively, a direct solution can be obtained using the Levenberg–Marquardt or Gauss–Newton iteration algorithm via Eq. (4). To solve the factor graph optimization problem, we use the Incremental smoothing and mapping (iSAM2) algorithm [43], which can be used for both batch optimization and real-time estimation. iSAM2 solves the graphical model directly without converting the factor graph into a sparse matrix and then applying linear algebra methods [43] by transforming it into a Bayesian network. Under the assumption of Gaussian noise, the algorithm incrementally estimates a set of variables (states of the mobile robot or the position of landmarks in the environment) and specifies a set of nonlinear factors, with both sets growing with time. The method incorporates just-in-time or fluid linearization, which allows them to keep track of the validity of the linearization points for each variable and linearize only when necessary [43], reducing the overall runtime. The factor graph we propose consists of the motion model factor f_{MM}, the scan matching factor of the radar f_{SM}, the ego-velocity factor of the radar f_R, the ego-velocity factor of the monocular odometry f_{MV}, and the rotation factor of the monocular odometry f_{MR}. Each factor is described in details in the following Sections 3.3-3.7. The ego-velocity factor of the radar and the monocular odometry are *unary* factors that constrain the states in which the radar measurements and monocular odometry estimates were obtained, respectively. The motion model factor establishes constraints between each consecutive state in the graph, while the radar scan matching factor and rotation factor of the monocular odometry set constraints between the two consecutive states in which the each sensor measurements were obtained. The described factor graph is illustrated in Fig. 2.

3.3. Motion model factor

To constrain the consecutive nodes within the factor graph, we use the constant velocity model as the motion model factor f_{MM}. The measurement function h^{MM} of this factor is expressed as follows:

$$h^{MM}(\mathbf{x}_{k-1}) = \begin{bmatrix} \mathbf{R}_{\mathrm{WC},k-1}\mathrm{Exp}[({}_{\mathrm{C}}\boldsymbol{\omega}_{k-1}\Delta t)^{\wedge})] \\ {}_{\mathrm{C}}\boldsymbol{\omega}_{k-1} \\ {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k-1} + {}_{\mathrm{W}}\mathbf{v}_{k-1}\Delta t \\ {}_{\mathrm{W}}\mathbf{v}_{k-1} \\ r_{k-1} \end{bmatrix} \tag{6}$$

where *Exp* and *hat* represent mappings from Euclidean space and the corresponding Lie algebra $\mathfrak{so}(3)$ to the Lie group SO(3) [39]. Therefore, the state in step k can be described as follows:

$$\mathbf{x}_k = h^{MM}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}^{MM}, \tag{7}$$

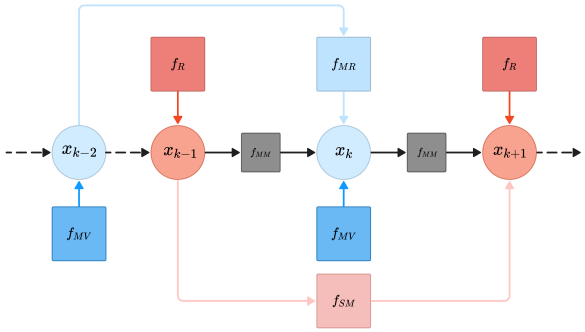


Figure 2: Illustration of the MOVRO2 factor graph structure. The factor graph consists of the motion model factor (f_{MM}), the scan matching factor (f_{SM}), the radar ego-velocity factor (f_R), the unscaled ego-velocity factor of the monocular odometry (f_{MV}), and the rotation factor of the monocular odometry (f_{MR}). The motion model factor establishes constraints between each consecutive state in the graph, while the radar scan matching factor and the rotation factor of the monocular odometry set constraints between the two consecutive states in which the each sensor measurements were obtained.

where $\mathbf{v}_{k-1}^{MM}$ stands for Gaussian noise with zero mean and covariance $\sum_k^{MM}$. The covariance of the motion model is calculated in the same way as in [44], using piecewise constant white angular and translational acceleration noise, which serves as a relaxation stage of the constant velocity assumption.

We can now form the error function ($\mathbf{e}_k^{MM}$)):

$$\|\mathbf{e}_k^{MM}\|^2_{\sum_k^{MM}} = \|\mathbf{x}_k - h^{MM}(\mathbf{x}_{k-1})\|^2_{\sum_k^{MM}} \tag{8}$$

3.4. Scan matching factor of the radar

The extraction of plane or edge features based on radar data is not easy due to radar noise. Therefore, we decided to integrate a pure scan matching factor into our method. We found that Generalized Iterative Closest Point (GICP) [45] provides acceptable results in contrast to the standard Iterative Closest Point (ICP) [46] or Normal Distribution Transform (NDT) [47], similar to those reported in [24].

GICP, a generalization of ICP, considers the local planar structure of two scans in a probabilistic model [45]. The key difference between standard ICP and GICP is the treatment of association problems in a probabilistic way.

The standard ICP minimization problem, which involves the alignment of point clouds A and B, is expressed as follows:

$$T \leftarrow \underset{\mathrm{x}}{\mathrm{argmax}}\left\{\sum_i w_i \|T \cdot b_i - m_i\|^2\right\}. \tag{9}$$

Here, w_i, m_i, T, and b_i denote the weight, the outcomes of identifying the nearest points in the point cloud A, the sought transformation and the data of the point cloud B, respectively. The weight w_i is assigned the value of 1 for inliers, where $m_i - T \cdot b_i \leq d_{\max}$, and the value of 0 for that characterize points beyond the specified threshold $d_{\max}$.

GICP introduces a probabilistic model in the minimization step, while the rest of the ICP algorithms remains unchanged to reduce the complexity and maintain the running time of the algorithm, making it well suited for integration into online odometry algorithms. In the probabilistic model, the authors assume

the existence of an underlying set of points, denoted as $\hat{A} = \{\hat{a}_i\}$ and $\hat{B} = \{\hat{b}_i\}$, generating A and B according to the distributions $a_i \sim \mathcal{N}(\hat{a}_i, C_i^A)$ and $b_i \sim \mathcal{N}(\hat{b}_i, C_i^B)$, where C_i^A and C_i^B represent covariance matrices [45]. Assuming perfect correspondence, i.e., $\hat{b} = T\hat{a}$, and using maximum likelihood estimators, the transformation can be solved iteratively by

$$T \leftarrow \underset{\mathrm{x}}{\operatorname{argmax}} \Big\{ \sum_i w_i \log(p(b_i - T a_i)) \Big\}. \tag{10}$$

The matching threshold d_{max} represents a trade-off between convergence and accuracy, with a low value leading to poor convergence and a large number of false matches. Further details can be found in [45].

We use the GICP transformation estimation from the previous to the current radar image and thus obtain the rotation $\mathbf{R}_{\mathrm{SM}}$ and the translation $\mathbf{t}_{\mathrm{SM}}$. A transformation that is expressed in the camera image frame and uses extrinsic parameters by multiplying the transformation matrices, is the following:

$$\begin{bmatrix} \mathbf{R}_{\mathrm{SM,C}} & {}_{\mathrm{C}}\mathbf{t}_{\mathrm{SM,C}} \\ 0_{3x1} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{\mathrm{RC}} & {}_{\mathrm{C}}\mathbf{t}_{\mathrm{RC}} \\ 0_{3x1} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathrm{SM}} & \mathbf{t}_{\mathrm{SM}} \\ 0_{3x1} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{\mathrm{RC}}^{\mathrm{T}} & -\mathbf{R}_{\mathrm{RC}}^{\mathrm{T}}\, {}_{\mathrm{C}}\mathbf{t}_{\mathrm{RC}} \\ 0_{3x1} & 1 \end{bmatrix}, \tag{11}$$

where $\mathbf{R}_{\mathrm{SM,C}}$, and ${}_{\mathrm{C}}\mathbf{t}_{\mathrm{SM,C}}$ stand for the GICP estimates for rotation and translation in the camera frame. We can now form the measurement function h^{SM} of the scan matching factor f_{SM} of the radar as follows:

$$h^{SM}(\mathbf{x}_{k-1}) = \begin{bmatrix} \mathbf{R}_{\mathrm{SM,C}}^{\mathrm{T}}\, \mathbf{R}_{\mathrm{WC},k-1} \\ {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k-1} + \mathbf{R}_{\mathrm{WC},k-1}\, {}_{\mathrm{C}}\mathbf{t}_{\mathrm{SM,C}} \end{bmatrix}. \tag{12}$$

The state evolution is described as:

$$\mathbf{x}_k = h^{SM}(\mathbf{x}_{k-1}) + \mathbf{v}_{k-1}^{SM} \tag{13}$$

where $\mathbf{v}_{k-1}^{SM}$ stands for Gaussian noise with zero mean and covariance $\sum_k^{SM}$. The GICP covariance is treated as a tuning parameter. Therefore, the error function is as follows:

$$\|\mathbf{e}_k^{SM}\|_{\sum_k^{SM}}^2 = \|\mathbf{x}_k - h^{SM}(\mathbf{x}_{k-1})\|_{\sum_k^{SM}}^2. \tag{14}$$

3.5. The ego-velocity factor of the radar

In addition to the 3D spatial information and often also the signal-to-noise ratio, radar systems provide the radial relative velocity of each detection. This velocity is the projection of the relative velocity between the detection and the radar in the direction of the detection, which is estimated using the Doppler effect.

To test the ease of integration of various methods of MOVRO2, we used both the open-source package *reve* [11], as in [8], and RAVE [15] for the ego-velocity measurements of the radar. Both implementations assume that most of the radar detections $\mathrm{D}_{\mathrm{i},k}$ ($i \in (1, M)$) within a single scan at a time instance k are static and contain the Doppler velocity measurement $v_{\mathrm{D_{i,k}}}$ and the 3D object position ${}_{\mathrm{R}}\mathbf{p}_{\mathrm{RD_{i},}k}$ expressed in $\mathcal{F}_{\mathrm{R}}$.

Each valid static radar detection measurement fulfills the following conditions:

$$-v_{\mathrm{D_{i},}k} = \frac{{}_{\mathrm{R}}\mathbf{p}_{\mathrm{RD_{i},}k}}{\|{}_{\mathrm{R}}\mathbf{p}_{\mathrm{RD_{i},}k}\|} \cdot {}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}, \tag{15}$$

where ${}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}$ is a 3D translational radar velocity at time k. Therefore, it is possible to determine a 3D translational velocity ${}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}$ with at least three non-coplanar radar measurements. Since radar measurements often face problems such as ghost targets and multipath effects, *reve* uses a three-point RANSAC approach [48] to exclude outliers, and the final refinement is performed with orthogonal distance regression as proposed in [49]. While RAVE offers multiple inlier and outlier rejection methods, we opted for the Cauchy robust loss kernel based on the results presented in the paper. Additionally, we used RAVE's proposed sliding-window filter to reject unlikely velocity estimates.

We can now define the measurement function h^R for the the ego-velocity factor of the radar f_R as follows:

$$h^R(\mathbf{x}_k) = {}_{\mathrm{R}}\mathbf{v}_k = \mathbf{R}_{\mathrm{RC}}(\mathbf{R}_{\mathrm{CW},k}\, {}_{\mathrm{W}}\mathbf{v}_k + {}_{\mathrm{C}}\boldsymbol{\omega}_k \times {}_{\mathrm{C}}\mathbf{t}_{\mathrm{RC}}). \tag{16}$$

The transformation between $\mathcal{F}_{\mathrm{W}}$ and $\mathcal{F}_{\mathrm{R}}$ assumes that the extrinsic calibration is known. The measurement model of the radar can be described as follows:

$$\mathbf{z}_k = h^R(\mathbf{x}_k) + \mathbf{v}_k^R \tag{17}$$

where $\mathbf{v}_k^R$ stands for Gaussian noise with zero mean and covariance $\sum_k^R$. The ego-velocity covariance of the radar is calculated by orthogonal distance regression. Moreover, we can define the error factor function as follows:

$$\|\mathbf{e}_k^R\|_{\sum_k^R}^2 = \|\mathbf{z}_k - h^R(\mathbf{x}_k)\|_{\sum_k^R}^2 \tag{18}$$

3.6. The ego-velocity factor of the monocular odometry

When the monocular odometry data arrives, we add a unary factor to the graph, i.e., the unscaled ego-velocity. In our prior work [8], we demonstrated that performing additional correction with the unscaled monocular ego-velocity by using a scale factor in the state can improve the overall system accuracy. Following this idea, we modified the scaling factor as follows:

$$r_k = \frac{\mathrm{norm}({}_{\mathrm{C}}\mathbf{v}_{\mathrm{M},k})}{\mathrm{norm}({}_{\mathrm{C}}\mathbf{v}_k)} = \frac{\mathrm{norm}({}_{\mathrm{C}}\mathbf{v}_{\mathrm{M},k})}{\mathrm{norm}(\mathbf{R}_{\mathrm{WC},k}^{\mathrm{T}}\, {}_{\mathrm{W}}\mathbf{v}_k)}, \tag{19}$$

where ${}_{\mathrm{C}}\mathbf{v}_{\mathrm{M},k}$ and ${}_{\mathrm{C}}\mathbf{v}_k$ denote the unscaled monocular odometry velocity and the system state velocity expressed in the camera image frame at time k, respectively. Keeping the scaling factor in the state allows correction with the monocular unscaled ego-velocity estimate to reduce odometry drift. Thus, the monocular odometry measurement function h^{MV} of monocular odometry factor f_{MV} is defined as

$$h^{MV}(\mathbf{x}_k) = {}_{\mathrm{C}}\mathbf{v}_{\mathrm{M},k} = r_k\, \mathbf{R}_{\mathrm{WC},k}^{\mathrm{T}}\, {}_{\mathrm{W}}\mathbf{v}_k. \tag{20}$$

The measurement model is given by:

$$\mathbf{z}_k = h^{MV}(\mathbf{x}_k) + \mathbf{v}_k^{MV}, \tag{21}$$

where $\mathbf{v}_k^{MV}$ stands for Gaussian noise with zero mean and covariance $\sum_k^{MV}$. According to Eq. (20), the error function of the factor is expressed as follows:

$$\|\mathbf{e}_k^{MV}\|_{\sum_k^{MV}}^2 = \|\mathbf{z}_k - h^{MV}(\mathbf{x}_k)\|_{\sum_k^{MV}}^2. \tag{22}$$

3.7. The rotation factor of the monocular odometry

We assume that rotation estimation provided by the monocular odometry is reliably accurate, thus we utilize the rotation estimation between frames as a binary factor in our framework. Using two consecutive orientation measurements, we compute the rotation matrix, denoted as $\mathbf{R}_{\mathrm{MR,C}}$, which represents the transformation from the previous state to the current state. Consequently, we can now define the measurement function h^{MR} for the scan matching factor f_{MR} of the radar as follows:

$$h^{MR}(\mathbf{x}_{k-1}) = \mathbf{R}^{\mathrm{T}}_{\mathrm{MR,C}}\,\mathbf{R}_{\mathrm{WC},k-1}. \tag{23}$$

The state evolution is described as:

$$\mathbf{x}_k = h^{MR}(\mathbf{x}_{k-1}) + \mathbf{v}^{MR}_{k-1} \tag{24}$$

where $\mathbf{v}^{MR}_{k-1}$ stands for Gaussian noise with zero mean and covariance $\sum^{MR}_k$. The covariance is treated as a tuning parameter. Therefore, the error function is as follows:

$$\|\mathbf{e}^{MR}_k\|^2_{\sum^{MR}_k} = \|\mathbf{x}_k - h^{MR}(\mathbf{x}_{k-1})\|^2_{\sum^{MR}_k}. \tag{25}$$

3.8. Factor graph integration

After defining the error function of each proposed factor, we can formulate the problem of the whole factor graph optimization, where the factor graph error function is usually calculated as a sum of the error functions of each factor for each state in the graph. Due to high noise in radar-generated point clouds, both *reve* or RAVE and GICP implementations can lead to outlier estimates in scan matching and velocity measurements. To counteract this problem, we introduce a robustness measure by including a Huber loss kernel in the solution of the the system, which is inspired by the work of Huang et al. [50]. We can now obtain the optimal $\hat{\mathbf{x}}$ by solving the factor graph optimization problem formulated as follows:

$$\hat{\mathbf{x}} = \underset{\mathrm{x}}{\operatorname{argmin}} \sum_k \Big(\|\mathbf{e}^{MM}_k\|^2_{\sum^{MM}_k} + \|\mathbf{e}^{MR}_k\|^2_{\sum^{MR}_k} + \|\mathbf{e}^{MV}_k\|^2_{\sum^{MV}_k} + \rho(\|\mathbf{e}^{R}_k\|^2_{\sum^{R}_k}) + \rho(\|\mathbf{e}^{SM}_k\|^2_{\sum^{SM}_k}) \Big), \tag{26}$$

where the Huber norm [51] is defined as follows [42]:

$$\rho(s) = \begin{cases} \frac{1}{2}\mathrm{s}^2 & \text{if } |\mathrm{s}| < 1 \\ \frac{1}{2}1^2 + l|\mathrm{s} - 1| & \text{otherwise.} \end{cases} \tag{27}$$

During optimization, fixed-delay smoothing with iSAM2 [43] is used to solve Eq. (26) with the GTSAM [42] library. We set a 10-second optimization window within which variables passed through will be marginalized as prior factors. The MOVRO2 factor graph construction and optimization is summarized in Algorithm 1.

Algorithm 1 MOVRO2

Inputs: Measurements $\mathbf{z}_{1:K}$
Outputs: Odometry estimates $\hat{\mathbf{x}}$

for $k = 1$ to K **do**
  $h^{MM} \leftarrow$ Motion model (Eq. 6)
  $\|\mathbf{e}^{MM}_k\|^2_{\sum^{MM}_k} = \|\mathbf{x}_k - h^{MM}(\mathbf{x}_{k-1})\|^2_{\sum^{MM}_k}$
  $h^{SM} \leftarrow$ GICP (Eq. (12))
  $\|\mathbf{e}^{SM}_k\|^2_{\sum^{SM}_k} = \|\mathbf{x}_k - h^{SM}(\mathbf{x}_{k-1})\|^2_{\sum^{SM}_k}$
  $h^{R} \leftarrow$ *reve* (Eq. (16))
  $\|\mathbf{e}^{R}_k\|^2_{\sum^{R}_k} = \|\mathbf{z}_k - h^{R}(\mathbf{x}_k)\|^2_{\sum^{R}_k}$
  $h^{MV} \leftarrow$ Monocular odometry (Eq. (20))
  $\|\mathbf{e}^{MV}_k\|^2_{\sum^{MV}_k} = \|\mathbf{z}_k - h^{MV}(\mathbf{x}_k)\|^2_{\sum^{MV}_k}$
  $h^{MR} \leftarrow$ Monocular odometry (Eq. (23))
  $\|\mathbf{e}^{MR}_k\|^2_{\sum^{MR}_k} = \|\mathbf{x}_k - h^{MR}(\mathbf{x}_{k-1})\|^2_{\sum^{MR}_k}$
end for
$\mathcal{L} = \sum_k \Big(\|\mathbf{e}^{MM}_k\|^2_{\sum^{MM}_k} + \|\mathbf{e}^{MR}_k\|^2_{\sum^{MR}_k} + \|\mathbf{e}^{MV}_k\|^2_{\sum^{MV}_k} + \rho(\|\mathbf{e}^{R}_k\|^2_{\sum^{R}_k}) + \rho(\|\mathbf{e}^{SM}_k\|^2_{\sum^{SM}_k}) \Big)$
$\hat{\mathbf{x}} = \underset{\mathrm{x}}{\operatorname{argmin}}\ \mathcal{L}$

4. Experimental results

In this section, we present the results of the experimental validation of MOVRO2 and its comparison with state-of-the-art monocular and multisensor odometry methods on two open datasets, namely the IRS dataset [2], a large indoor dataset covering a wide range of scenes, and the View-of-Delft (VoD) dataset [13], a large automotive multiclass object detection dataset. We compared MOVRO2 with ROVIO [10], EKF-RIO [11], RVIO [2], 4DRVO-Net [3] and CMFlow [12], among others. We also performed ablation studies with both datasets to investigate the impact of each proposed factor on MOVRO2 performance. To demonstrate the robustness of MOVRO2, we also tested and analyzed its performance in certain scenarios where monocular camera or radar measurements are not available at times.

The implementation of MOVRO2 in both datasets operates in real-time under the assumption that other sub-modules, namely scan-to-scan, velocity estimation, and monocular odometry, are performing in real-time. It is imperative to emphasize that radar and camera sensors with reasonable costs were used in both datasets we tested on in this study, making this sensor setup cost-effective. To ensure a fair comparison, we did not take advantage of closing the loop when evaluating monocular visual odometry.

For the evaluation, we decided to compare the relative pose error (RPE), i.e. the relative translation error (t_{rel}) and rotation error (r_{rel}), as both datasets have accurate and reliable ground truth measurements and RPE is better tailored to odometry than the absolute trajectory error (ATE) metric used in [8], which gives no indication of where the pose estimation takes place, and tiny errors in the orientation estimates accumulate to large ATE values as the trajectory progresses, as discussed in [52]. We used the open-source package evo [53], which implements the calculation of the relative pose error.

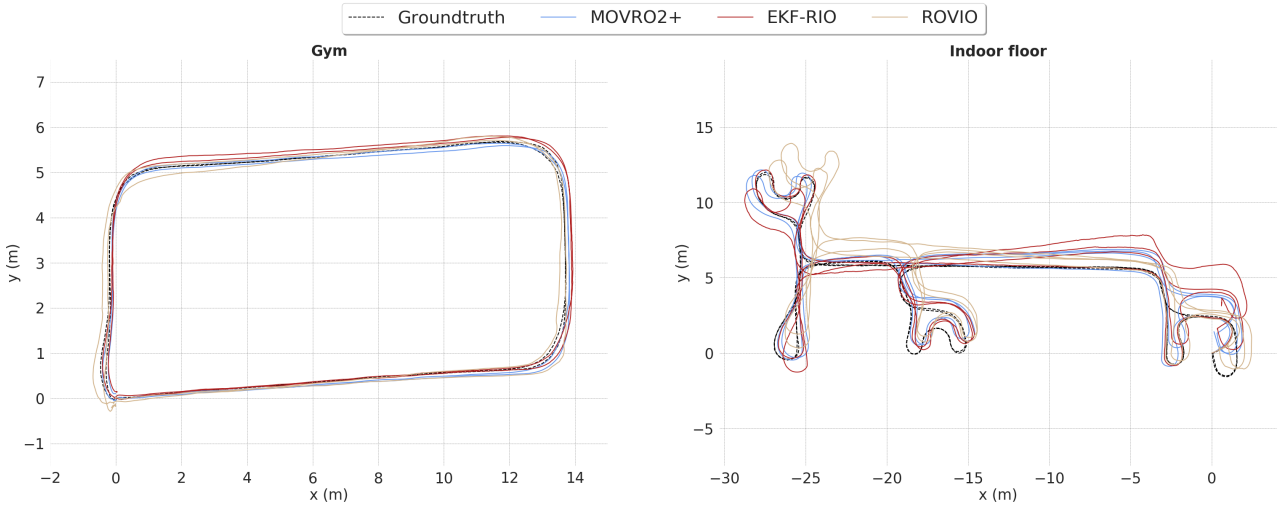


Figure 3: Visualization of trajectories for Gym and Indoor_floor sequences of the IRS dataset.

4.1. IRS dataset

The IRS Radar Thermal Visual Inertial Dataset is an open-source dataset for ego-motion estimation tasks [2]. It includes over 17 minutes of recordings, with cumulative trajectory lengths exceeding 1000 meters with containing ground truth motion.

4.1.1. Experimental setup

The sensor platform of the IRS dataset is equipped with a high-performance IMU (Analog Devices ADIS16448), a Frequency Modulated Continous Wave Radar (Texas Instruments IWR6843AOP) and a monocular grayscale camera (IDS UI-3241). The data frequency of the sensors is as follows: IMU (409 Hz), a visual camera (20 Hz) and radar (10 Hz). The IWR6843AOP is a single-chip 60 GHz radar with an integrated antenna array (3 Tx and 4 Rx) with 120° FOV in azimuth and elevation [2]. The monocular camera images have a resolution of 640 × 512 px. All sensors are synchronized by implementing a hardware trigger and sync signals via a microcontroller board. However, it is important to note that the timestamps of the radar and camera data do not contain exactly the same values and therefore the temporal correlation between the measurements is preserved. The mocap sequences were recorded in a laboratory environment with the Vicon motion capture system as the ground truth reference, while the remaining sequences were recorded in different environments with different trajectory lengths of the scene using VINS [54] (with loop closure) as the ground truth [2].

4.1.2. Results

To emphasize the versatility and advantages of a loosely coupled fusion approach, ORB-SLAM3 [17] was used offline without loop closures and additionally OV2SLAM [16] in online mode to obtain monocular odometry estimates. For radar ego-velocity estimation, we used the open-source packages *reve* [14] and RAVE [15], and GICP [45] for radar scan-to-scan matching. We performed the evaluation on six dataset sequences (Gym, Indoor_floor, Mocap_easy, Mocap_medium, Outdoor_campus and Outdoor_street) out of a total of nine dataset sequences, as ORB-SLAM3 failed due to fast movements (Mocap_difficult) or poor illumination (Mocap_dark, Mocap_dark_fast), similar to what was reported in [55]. We compared MOVRO2 with seven other methods: (i) our previous EKF-based monocular visual radar odometry MOVRO [8], (ii) the monocular visual inertial odometry ROVIO [10] using 25 standard features in the filter condition, (iii) the radar inertial odometry EKF-RIO [11], (iv) the visual radar inertial odometry RVIO [2], (v) the monocular visual odometry ORB-SLAM3, (vi) the monocular visual odometry OV2SLAM and (vii) the generalized iterative closest point GICP [45].

Table 1 shows the relative pose errors for six IRS dataset sequences and Table 2 their mean values. MOVRO2* stands for an implementation that integrates ORB-SLAM3, while MOVRO2 and MOVRO+ integrate OV2SLAM with *reve* and RAVE respectively into the factor graph framework. EKF-RIO was evaluated without the barometer to ensure a fair comparison. The parameters of ROVIO, EKF-RIO and RVIO were taken from [2], those of MOVRO and ORB-SLAM3 from [8]. The results are divided into mono-, dual- and three-sensor solutions. Due to the sparse radar data, GICP cannot provide interpretable results. Monocular odometry methods have no absolute scale, but provide accurate orientation estimates. Both MOVRO and MOVRO2* utilize the same data and use the scale factor. Given the limited performance of GICP in tracking noisy radar data, they have similar relative rotational and translational errors. However, since MOVRO2* incorporates more historical data through the FGO framework, it shows a lower relative rotation and translation error. MOVRO and MOVRO2 instances achieve the smallest relative translation errors, highlighting the advantages of combining radar and monocular camera sensors while maintaining or reducing the relative rotation errors of monocular odometry. The results from MOVRO2 and MOVRO2+, which integrate OV2SLAM over ORB-SLAM3, demonstrate the advantages of sensor integration in a loosely

Method	M	R	I	Gym		Indoor_floor		Mocap_easy		Mocap_medium		Outdoor_campus		Outdoor_street	
				t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
ORB-SLAM3	✓			0.285	1.333	0.252	0.199	0.031	0.626	0.034	0.108	0.370	**0.088**	0.525	**0.074**
OV2SLAM	✓			0.571	**0.080**	2.362	**0.152**	0.013	**0.057**	0.102	**0.068**	0.571	**0.082**	9.060	**0.060**
GICP		✓		2.757	14.902	1.284	29.272	0.362	9.179	0.370	10.473	1.988	10.913	4.898	17.009
EKF-RIO		✓	✓	0.024	0.274	0.030	0.351	0.008	0.257	0.011	0.458	0.075	0.377	0.047	0.244
ROVIO	✓		✓	0.024	**0.081**	**0.016**	**0.080**	**0.002**	**0.025**	**0.003**	**0.035**	0.090	0.105	0.148	0.110
MOVRO	✓	✓		**0.016**	1.333	**0.019**	0.199	0.004	0.631	0.007	0.125	**0.024**	**0.088**	**0.019**	**0.074**
MOVRO2*	✓	✓		**0.015**	1.333	0.020	0.199	**0.003**	0.630	**0.006**	0.109	**0.022**	**0.088**	**0.020**	**0.074**
MOVRO2	✓	✓		**0.015**	**0.080**	**0.019**	**0.152**	**0.003**	**0.057**	**0.006**	**0.068**	**0.022**	**0.082**	**0.020**	**0.060**
MOVRO2+	✓	✓		**0.015**	**0.080**	**0.019**	**0.152**	**0.003**	**0.057**	**0.006**	**0.068**	**0.022**	**0.082**	**0.020**	**0.060**
RVIO	✓	✓	✓	0.011	0.082	0.012	0.071	0.002	0.024	0.003	0.074	0.021	0.086	0.022	0.082

Table 1: Relative translation and relative rotation error (t_{rel} (m), r_{rel} (deg)) for six IRS dataset sequences, rounded to three decimal places. For each sequence, the best result obtained with a monocular or monocular radar odometry method is highlighted in bold and the second best in blue. The symbols M, R and I stand for the monocular camera, the radar and the IMU sensor respectively.

coupled fashion, highlighting its scalability and ease of integration, reducing the overall relative rotation error, while relative translation error stays the same as in MOVRO2*. Additionally, there is no significant difference between the results of MOVRO2 and MOVRO2+. Our solution is comparable to the state-of-the-art of visual-inertial-radar odometry (RVIO) and even outperforms it in the outdoor street sequence in both metrics. The visualization of the trajectories is shown in Fig. 3.

Method	M	R	I	Mean	
				t_{rel}	r_{rel}
ORB-SLAM3	✓			0.250	0.405
OV2SLAM	✓			2.113	**0.083**
GICP		✓		1.943	15.291
EKF-RIO		✓	✓	0.033	0.327
ROVIO	✓		✓	0.047	**0.073**
MOVRO	✓	✓		**0.015**	0.408
MOVRO2*	✓	✓		**0.014**	0.406
MOVRO2	✓	✓		**0.014**	**0.083**
MOVRO2+	✓	✓		**0.014**	**0.083**
RVIO	✓	✓	✓	0.012	0.070

Table 2: Average relative translation and relative rotation error (t_{rel} (m), r_{rel} (deg)) for the evaluated methods in Table 1. The two best results for each dataset sequence are highlighted in bold.

Version of MOVRO2+	Gym t_{rel} (m)	Mocap_easy t_{rel} (m)	Outdoor_campus t_{rel} (m)
Plain	0.01524	0.00307	**0.02198**
Scale	0.01520	0.00307	0.02199
GICP	0.01520	0.00307	0.02190
Full	**0.01517**	0.00307	0.02200

Table 3: Ablation study of the MOVRO2 method on Gym, Mocap_easy, and Outdoor_street sequences of the IRS dataset. The best result for each dataset sequence is highlighted in bold.

To determine the effects of each factor on the relative translational error t_{rel}, we performed an ablation study of the MOVRO2+ method with three sequences, as shown in Table 3. The "Plain" version refers to the MOVRO2+ framework without the obtained scaling factor within the state and the exclusion of the scan matching factor. The "Scale" version refers to MOVRO2+ with scaling factor but without the scan matching factor. In contrast, the "GICP" version refers to our method with the scan matching factor, but without the scaling factor. Finally, the "Full" version refers to the implementation with both the scaling factor within the state and the scan matching factor. Given the significant dependence of relative rotation error (r_{rel}) on monocular odometry due to twice the data frequency (20 Hz) compared to radar ego-velocity estimates (10 Hz), our evaluation focused solely on the t_{rel} metric. The addition of the scaling factor and scan-matching factor led to a reduction of translation error by 0.07% and 0.30%, respectively. The ablation study confirms that the integration of the scaling factor as well as the scan matching factor does not degrade the performance. On the contrary, these additions increase the robustness of the system and thus make the fusion more reliable.

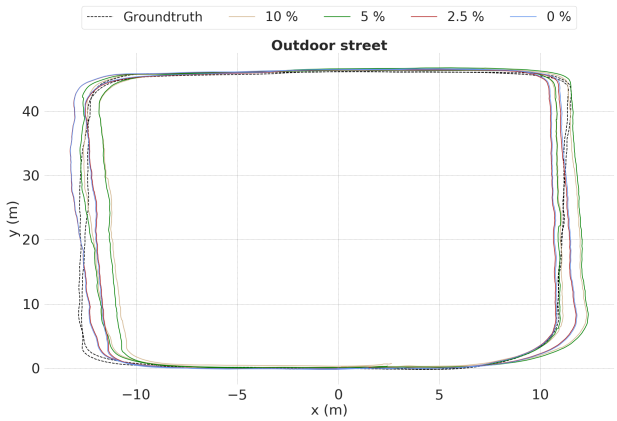


Figure 4: The visualization of trajectories in all four robustness test scenarios on Outdoor_street sequence of the IRS dataset.

To demonstrate the system's robustness against large measurement errors, we evaluated our system with occasionally

random ego-velocity measurements of the radar. For 2.5%, 5%, and 10% of random ego-velocity measurements from the radar, we estimated the odometry using MOVRO2+. We used the highest value of the estimated velocity in the sequence for the maximum allowed random value. The results are provided in Table (4), and the visualization of the simulation on the Outdoor_street sequence is shown in Fig. 4. The random values do not affect the relative rotation error but, as expected, degrade the relative translation error. However, our method achieves competitive performance compared to other dual-sensor solutions, even with 10% of the radar ego-velocity measurement being completely random. This robustness is attributed to the Huber norm integration, which successfully mitigates the large impact of spurious measurements.

Robustness test	Mocap_medium		Outdoor_street	
	t_{rel} (m)	r_{rel} (deg)	t_{rel} (m)	r_{rel} (deg)
10 %	0.010	0.068	0.046	0.060
5 %	0.008	0.068	0.031	0.060
2.5 %	0.007	0.068	0.025	0.060
MOVRO2+	0.006	0.068	0.020	0.060

Table 4: Robustness tests for the MOVRO2+ method were conducted in three different simulated scenarios using test sequences Mocap_medium and Outdoor_street from the IRS Dataset. For each sequence, the percentage of random ego-velocity estimates is varied to test the system's robustness to large sensor errors.

4.1.3. Discussion

The presented results on the IRS dataset demonstrate the accurate, modular, and robust performance of the MOVRO2 instances. All three instances of MOVRO2 achieved the smallest t_{rel}, while MOVRO2 and MOVRO2+ incorporating OV2SLAM monocular odometry achieved the second-best relative rotation error results among mono- and dual-sensor solutions. Our proposed method even outperformed a state-of-the-art three-sensor solution in one of the sequences. We demonstrate the modularity of method by using OV2SLAM over ORB-SLAM3, and RAVE over *reve* to illustrate the seamless adaptability to future state-of-the-art radar ego-velocity, scan matching, and monocular visual estimators. The ablation study of factor contribution shows that the integration of additional factors does not degrade performance. The scan-matching and scale factor integration did not significantly improve accuracy, as GICP did not perform well on low-cost radar data and the scale factor did not increase accuracy because the ego-velocity measurements are already quite accurate as MOVRO2 method achieve smaller t_{rel} among dual-sensor estimation methods. In further experiments, we tested our system against large errors in ego-velocity estimates of the radar, where our system demonstrated extreme robustness due to the integration of the Huber norm.

4.2. VoD dataset

The View-of-Delft (VoD) dataset was recorded with a vehicle, favoring scenarios with vulnerable road users, i.e., cyclists and pedestrians [13].

4.2.1. Experimental setup

The dataset contains recorded data from a ZF FR-Gen21 3+1D radar (13 Hz), a stereo camera pair (1936 × 1216 px, 30 Hz), a Velodyne HDL-64 S3 lidar scanner (10 Hz) and the odometry of the ego-vehicle estimated with RTK-GPS, IMU and wheel odometry. In addition, the dataset contains "synchronized frames" consisting of a rectified monocular image, a radar point cloud, a LiDAR point cloud and odometry described as a transformation matrix. The authors chose LiDAR as the leading sensor and selected the closest available camera and radar information with a maximum tolerated time of 0.05 seconds, which corresponds to the LiDAR frequency of 10 Hz [13]. The images are arranged sequentially at 10 Hz with an average length of about 40 seconds. This type of synchronization benefits tasks such as multi-class 3D object detection, for which this dataset is intended, but not ego-motion estimation, as the temporal correlation between the sensor data is not maintained. In addition, the dataset also provides ego-motion compensated Doppler measurements, but in our implementation we use raw Doppler data.

4.2.2. Results

Table 5 shows the relative pose errors of our method and other evaluated odometry methods for seven test sequences of the VoD dataset [12, 3] and their mean values. We have evaluated MOVRO2 and MOVRO2+ methods that integrate OV2SLAM into the FGO, since ORB-SLAM3 often could not be initialized due to motion at the beginning of the sequences, and compared it with eight other methods: three monocular visual odometry methods (DeepVO [56], TartanVo [57], and OV2SLAM [16]), three radar-based odometry methods (RaFlow [29], GICP [45], and CMFlow [12]), one monocular visual radar odometry method (4DRVO-Net [3]), and one LiDAR-based odometry method (A-LOAM [58]). For the estimation of radar ego velocity for MOVRO2 and MOVRO2+, we employed *reve* and RAVE, respectively. The A-LOAM method is evaluated with a high performance lidar and is the only method that has "true" time stamps, so that a direct comparison with other methods is not possible. Similar to the IRS dataset, the monocular visual estimator tends to provide a more accurate orientation estimate but no scaling, while the radar-based estimator provides a smaller translation error but a large relative error estimate. Therefore, it should not be surprising that visual radar odometries have better relative translation and relative rotation errors with comparable accuracy to lidar odometries. In contrast to the results in Table 1 for the IRS dataset, we can see that GICP provides competitive results for the VoD dataset, even compared to CMFlow and RaFlow, since the dataset contains next-generation radar. Both instances of MOVRO2 method reduced the relative rotation error of the best monocular odometry (OV2SLAM) by 6.25%, while MOVRO2+ achieves the same relative translation error as 4DRVO-Net, but without the need for a training dataset and the training process. MOVRO2+ even outperforms LiDAR odometry in sequence 19, which contains a large number of moving objects such as cyclists. However, the proposed method shows the overall worst performance in sequence 03 as

	Method	03		04		09		17		19		22		24		Mean	
		t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Camera	DeepVO	1.34	2.75	1.87	1.46	1.56	2.01	0.82	0.48	5.14	4.04	1.91	1.98	1.41	2.22	2.00	2.13
	TartanVO	0.78	0.25	0.19	0.12	0.37	0.11	0.55	**0.04**	1.01	0.64	0.64	**0.18**	0.48	0.19	0.57	0.22
	OV2SLAM	0.18	0.24	0.21	0.22	0.25	0.16	0.64	0.06	**0.05**	**0.12**	0.20	0.23	0.26	**0.09**	0.26	0.16
Radar	RaFlow	0.87	2.09	0.07	0.44	0.11	**0.09**	0.13	**0.03**	1.22	4.09	0.72	1.34	0.25	1.14	0.48	1.32
	GICP	0.09	0.41	0.09	0.35	0.10	0.27	0.19	0.22	0.07	0.25	0.09	0.52	**0.09**	0.20	0.10	0.32
	CMFlow	**0.06**	**0.10**	**0.05**	**0.09**	**0.09**	0.14	**0.06**	**0.03**	0.28	0.94	0.14	0.29	0.12	0.58	0.11	0.31
Camera-radar fusion	4DRVO-Net	**0.02**	**0.02**	**0.01**	**0.02**	**0.03**	**0.05**	**0.11**	**0.03**	0.26	**0.08**	**0.06**	**0.09**	**0.09**	0.18	**0.08**	**0.07**
	MOVRO2	0.09	0.23	0.09	0.20	0.10	0.14	0.14	0.05	**0.03**	**0.12**	0.08	0.23	**0.08**	**0.10**	**0.09**	**0.15**
	MOVRO2+	0.08	0.23	0.08	0.20	0.10	0.14	0.14	0.05	**0.03**	**0.12**	**0.07**	0.22	**0.08**	**0.10**	**0.08**	**0.15**
LiDAR	A-LOAM	0.06	0.10	0.03	0.12	0.06	0.06	0.03	0.02	0.14	0.19	0.06	0.11	0.06	0.08	0.06	0.10

Table 5: Relative translation and relative rotation error (t_{rel} (m), r_{rel} (deg)) for seven VoD dataset sequences, rounded to two decimal places. For each sequence, the best result obtained with a monocular, a radar or a monocular radar odometry method is highlighted in bold and the second best in blue.

OV2SLAM exhibits largest relative rotation error in the following sequence. MOVRO2+,which integrates ego-velocity estimation from RAVE liberary exhibits smaller t_{rel} compared to MOVRO2, indicating that using more accurate ego velocity estimation methods can further enhance the accuracy of our system's results. The visualized trajectories in Fig. 5 illustrate the small deviation of our method.

Version of MOVRO2+	03		22	
	t_{rel} (m)	r_{rel} (deg)	t_{rel} (m)	r_{rel} (deg)
Plain	**0.084022**	0.234338	**0.065183**	0.221638
Scale	0.084998	0.234341	0.065201	0.221642
GICP	0.084992	**0.234329**	**0.065183**	**0.221626**
Full	0.084998	0.234331	0.065206	0.221631

Table 6: Ablation study of the MOVRO2+ method on sequences 03 and 22 of the VoD dataset. The best result for each dataset sequence is highlighted in bold.

We performed the same ablation study for the MOVRO2+ method as for the IRS dataset. From the results shown in Table 6, it is clear that additional factors do not significantly improve or degrade the overall accuracy of the method, as OV2SLAM and RAVE provide accurate estimates. However, the introduction of additional factors was motivated to increase the robustness of the system.

Same as for IRS dataset, we demonstrate the system's robustness against large measurement errors by evaluating it with occasionally random ego-velocity measurements from the radar on VoD dataset where the highest value of the estimated velocity in the sequence for the maximum allowed random value We estimated the odometry using MOVRO2+ with 2.5%, 5%, and 10% of random ego-velocity measurements from the radar. The results are provided in Table (7), and the visualization of the simulation on the 09 and 24 sequence of the VoD dataset is shown in Fig. 6. Same as in IRS dataset, we can conclude that random values do not affect the relative rotation error but do degrade the relative translation error. MOVRO2+ with 2.5% random radar ego-velocity measurements achieves similar results to MOVRO2+ without outliers, due to the Huber norm, which mitigates the impact of the outliers.

Robustness test	09		24	
	t_{rel} (m)	r_{rel} (deg)	t_{rel} (m)	r_{rel} (deg)
10 %	0.14	0.14	0.18	0.10
5 %	0.13	0.14	0.12	0.10
2.5 %	0.12	0.14	0.09	0.10
MOVRO2+	0.10	0.14	0.08	0.10

Table 7: Robustness tests for the MOVRO2+ method were conducted in three different simulated scenarios using test sequences 09 and 24 from the VoD Dataset. For each sequence, we varied the percentage of random ego-velocity estimates to test the system's robustness to large sensor errors.

4.2.3. Robustness to failure modes

To demonstrate the robustness of the MOVRO2+ method, we ran four different scenarios with test sequences 04 and 17 of the VoD dataset. The results are shown in Table 8. In the first scenario (MO complete failure), we completely omitted the monocular odometry data to simulate the situation in which the monocular camera pipeline or the monocular visual odometry do not provide any data. Consequently, the method relies solely on radar data. MOVRO2+ without monocular camera data showed a similar relative translation error t_{rel} as MOVRO2+ with all available data. In addition, it has a higher overall accuracy compared to GICP. In the second scenario (MO partial failure), we simulated the failure of OV2SLAM odometry by removing 31% and 42% of the data in sequences 04 and 17, respectively. This scenario was designed to show how MOVRO2+ would react if the data frequency of the camera was significantly reduced. It proved to be robust in this scenario and provided the relative translation error in sequence 17 equal to the third best in Table 5. The advantage of integrating monocular odometry data is evident in both metrics in contrast to the first scenario. In contrast to the previous scenario, the third scenario tested the robustness of the MOVRO2+ method in the absence of32% and 38% radar data in sequences 04 and

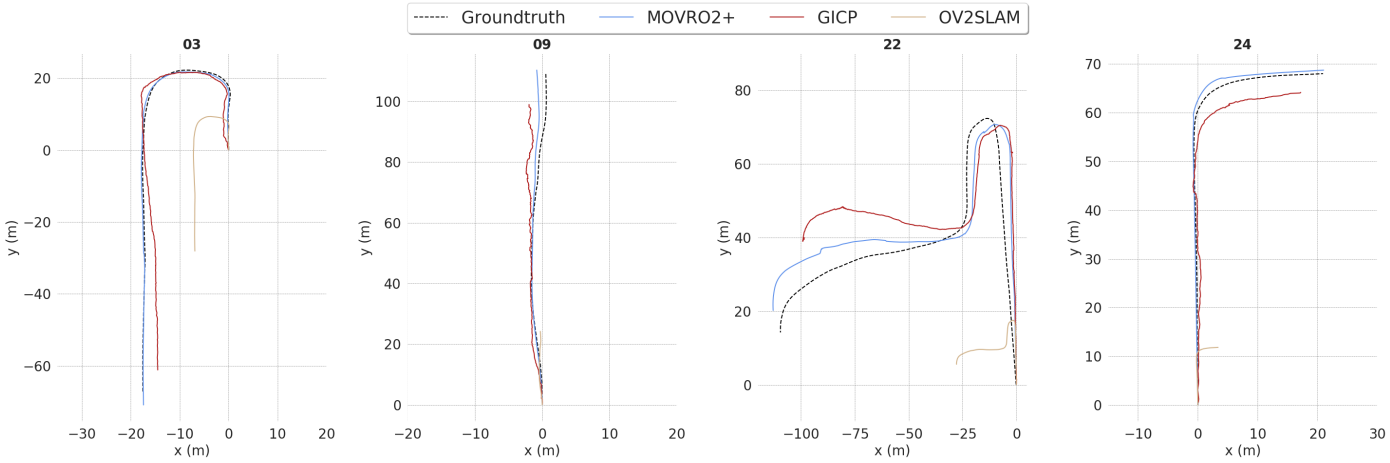


Figure 5: Visualization of the trajectories for sequences 03, 09, 22 and 24 of the VoD dataset.

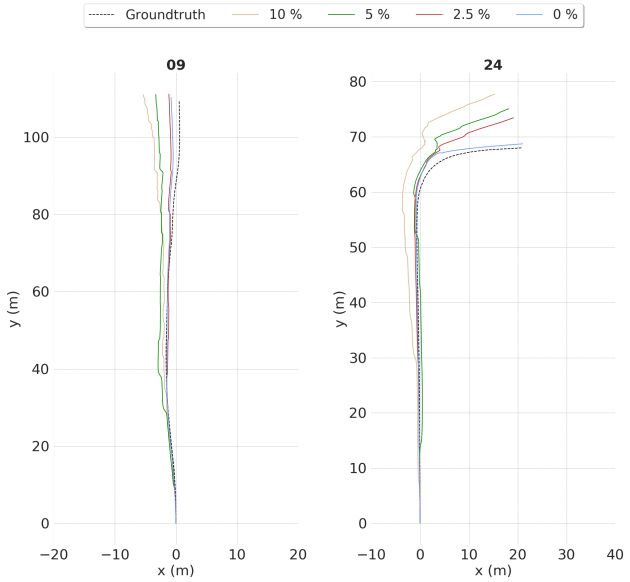


Figure 6: The visualization of trajectories in all four robustness test scenarios on 09 and 24 sequence of the VoD dataset.

17, respectively. The third scenario provided more frequent and accurate rotation estimates due to additional monocular odometry data, but achieved the same relative translation error due to the obtained within-state scaling factor as well as the motion model factor. The third scenario shows similar performance to the standard MOVRO2+ scenario with all data and factors. Finally, in the fourth scenario, we simulated same partial sensor failures of both the radar and monocular odometry data. In this way, we tested the robustness of our method against the asynchronous arrival of measurements and failures of simulated estimation method or sensor communication. It is important to note that radar and camera data contain temporal inaccuracies due to the synchronization of cameras and radar data with LiDAR. Despite this significant loss of data, our method did not fail and achieved a performance almost comparable to the standard MOVRO2+ using the full sensor data. Fig. 7 shows the details of the arrival of the measurements in all four scenarios in sequences 04 and 17, while Fig. 8 shows that the trajectories in all these scenarios exhibit minor deviations, inaccuracies or errors.

Failure modes	04		17	
	t_{rel}	r_{rel}	t_{rel}	r_{rel}
MO - complete failure	**0.08**	0.30	**0.14**	0.06
MO - partial failure	**0.08**	0.25	**0.14**	0.06
R - partial failure	**0.08**	**0.20**	0.15	**0.05**
MO & R - partial failure	0.09	0.29	**0.14**	0.07
MOVRO2+	**0.08**	**0.20**	**0.14**	**0.05**

Table 8: Relative translation and relative rotation error (t_{rel} (m), r_{rel} (deg)) for two VoD dataset sequences, rounded to two decimal places.Robustness test of the MOVRO2+ method, performed in four different simulated scenarios with test sequences 04 and 17 of the VoD dataset. The best results of each sequence are highlighted in bold. MO and R symbols represent the monocular visual odometry and radar data, respectively.

4.2.4. *Discussion*

The MOVRO2 experimental results, consistent with those on the IRS dataset, achieved competitive performance, resulting in smallest t_{rel} alongside the state-of-the-art 4DRVO-Net [3]. Ablation study showed that integration of additional factors did not yield more accurate results but did enhance the overall performance scope of our method. Robustness tests against large errors in the radar ego-velocity estimation demonstrated our system's resilience to unlikely ego-velocity estimates. In further experiments, we investigated the performance of MOVRO2 in different case scenarios, without monocular visual odometry, without a significant amount of monocular visual odometry and radar data. In the final robustness test, we simulated both radar and monocular odometry data failures. In all cases, the MOVRO2 method proved robust to various data dropouts. Even in the absence of monocular odometry data, MOVRO2 achieved comparable results to other mono- and dual-sensor solutions. After these experiments, we can conclude that the additional scan matching factor in MOVRO2 has not signifi-

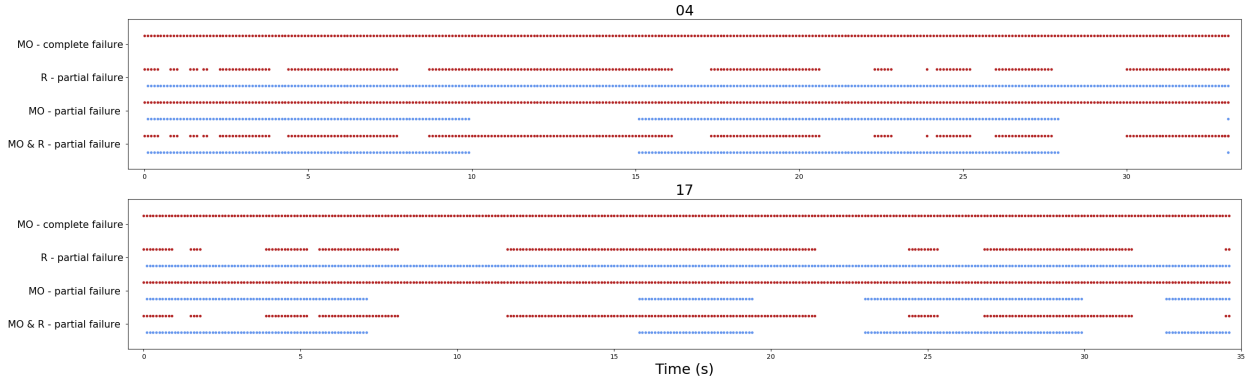


Figure 7: The details of the arrival of the measurements in all four robustness test scenarios in sequences 04 and 17 of the VoD dataset.

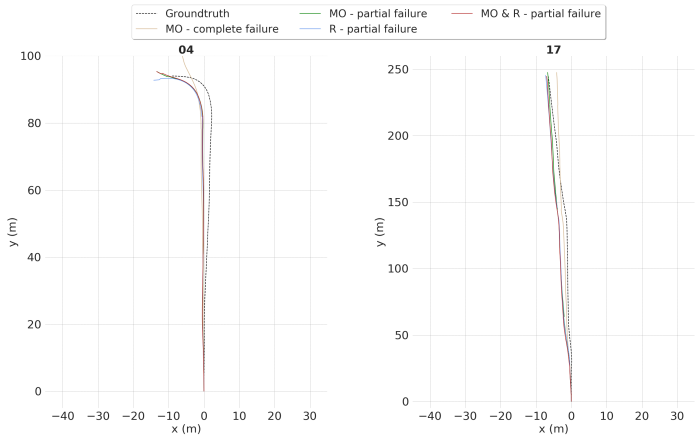


Figure 8: The trajectories in all four robustness test scenarios in sequences 04 and 17 of the VoD dataset.

cantly increased its overall accuracy, but has significantly improved its versatility and robustness. If the scan matching factor were missing, MOVRO2 would not be able to rely on the radar alone, as it would try to solve a problem with too few constraints. A tight integration of the scan matching factor with the ego-velocity estimation would improve the overall accuracy of the method, but the method would be more susceptible to sensor modality failures. The consistency of the estimation is evident from the results, with the MOVRO2 estimation results being among the four most accurate estimators in both metrics for each sequence in both datasets. In contrast, the 4DRVO-Net method, which integrates deep learning techniques, exhibited a significantly large relative translation error in sequence 19 of the VoD dataset compared to other sequences. Moreover, unlike 4DRVO-Net, MOVRO2 does not require additional data for training, which is often time-consuming to collect, and its tuning parameters are interpretable and can be easily adapted to different system configurations.

5. Applicable scope and limitations of the method

The presented experimental results illustrate the ability of MOVRO2 to reliably, accurately and robustly estimate odometry. For state estimation we used the iSAM2 algorithm making our implementation suitable for real-time operation (assuming that other submodules operate in real-time). MOVRO2 is not subject to constraints on environmental conditions (e.g., planes or edges) or message arrival, which distinguishes it from the state-of-the-art 3D visual radar odometry 4DRVO-Net [3] in terms of robustness. In addition, the proposed method is not limited to sensor frequency, but also allows the easy integration of additional factors, e.g. the addition of a GNSS position correction.

The proposed sensor fusion approach for odometry estimation offers a promising method to advance pose and velocity estimation using a more cost-effective solution compared to radar inertial or monocular visual-inertial systems. While a low-cost IMU may seem appealing, it often proves impractical due to issues with bias calibration and the gravity alignment, which can degrade bias estimates. However, our system, equipped with modern radar technology capable of scan-matching, can operate in radar-only mode even under visually degraded conditions, unlike conventional VIO systems. Additionally, our system sets itself apart from traditional radar-camera setups by prioritizing odometry estimation over common ADAS functions. While our experimental results demonstrate promising performance, there is room for future research and development. By tightly coupling images or implementing loop closure within the FGO framework, our system could achieve a higher level of robustness. Additionally, enabling extrinsic online calibration of the radar within the FGO framework or addressing issues related to point cloud registration degeneracy using a low-cost radar could also improve the accuracy and robustness of our method.

6. Conclusion

In this paper, we have proposed a loosely coupled monocular visual radar odometry method in which we fuse the ego-velocity of radar, scan-to-scan matching with the rotation obtained from consecutive steps and the unscaled velocity of monocular odometry. The proposed method, called MOVRO2, was evaluated with two open-source datasets. Compared to various mono- and multi-sensor solutions, MOVRO2 ranks in the top two results and even outperforms three-sensor odometry (visual, inertial snd radar) and LiDAR odometry with a pow-

erful 64-line LiDAR in certain sequences. Furthermore, we have demonstrated the versatility, modularity and robustness of the proposed method by simulating errors in the incoming measurement data.

ACKNOWLEDGMENT

This research has been supported by the European Regional Development Fund under the grant PK.1.1.02.0008 (DATACROSS).

References

[1] J. Michalczyk, R. Jung, C. Brommer, S. Weiss, Multi-state tightly-coupled ekf-based radar-inertial odometry with persistent landmarks, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 4011–4017.

[2] C. Doer, G. F. Trommer, Radar visual inertial odometry and radar thermal inertial odometry: Robust navigation even in challenging visual conditions, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2021) 331–338.

[3] G. Zhuoins, S. Lu, L. Xiong, H. Zhouins, L. Zheng, M. Zhou, 4drvo-net: Deep 4d radar–visual odometry using multi-modal and multi-scale adaptive fusion, IEEE Transactions on Intelligent Vehicles (2023).

[4] Y. Cheng, J. Su, M. Jiang, Y. Liu, A novel radar point cloud generation method for robot environment perception, IEEE Transactions on Robotics 38 (6) (2022) 3754–3773.

[5] K. Koledić, I. Cvišić, I. Marković, I. Petrović, Moft: Monocular odometry based on deep depth and careful feature selection and tracking, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 6175–6181.

[6] R. Nabati, H. Qi, Centerfusion: Center-based radar and camera fusion for 3d object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1527–1536.

[7] R. Nabati, H. Qi, Radar-camera sensor fusion for joint object detection and distance estimation in autonomous vehicles, arXiv preprint arXiv:2009.08428 (2020).

[8] V.-J. Štironja, J. Peršić, I. Marković, I. Petrović, Movro: Loosely coupled ekf-based monocular visual radar odometry, International Conference on Intelligent Autonomous Systems (IAS) (2023).

[9] W. Wen, T. Pfeifer, X. Bai, L.-T. Hsu, Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter, NAVIGATION: Journal of the Institute of Navigation 68 (2) (2021) 315–331.

[10] M. Bloesch, M. Burri, S. Omari, M. Hutter, R. Siegwart, Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback, The International Journal of Robotics Research (IJRR) 36 (2017) 1053–1072.

[11] C. Doer, G. F. Trommer, Radar inertial odometry with online calibration, European Navigation Conference (ENC) (2020).

[12] F. Ding, A. Palffy, D. M. Gavrila, C. X. Lu, Hidden gems: 4d radar scene flow learning using cross-modal supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 9340–9349.

[13] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, D. M. Gavrila, Multi-class road user detection with 3+ 1d radar in the view-of-delft dataset, IEEE Robotics and Automation Letters (RA-L) 7 (2) (2022) 4961–4968.

[14] C. Doer, G. F. Trommer, An ekf based approach to radar inertial odometry, IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI) (2020) 152–159.

[15] V.-J. Štironja, L. Petrović, J. Peršić, I. Marković, I. Petrović, Rave: A framework for radar ego-velocity estimation, arXiv preprint arXiv:2406.18850 (2024).

[16] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, G. Le Besnerais, Ov2 slam: A fully online and versatile visual slam for real-time applications, IEEE Robotics and Automation Letters (RA-L) 6 (2) (2021) 1399–1406.

[17] C. Campos, R. Elvira, J. J. Rodriguez, J. M. Montiel, J. D. Tardos, Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam, IEEE Transactions on Robotics 37 (2021) 1874–1890.

[18] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, K. Dietmayer, Instantaneous ego-motion estimation using doppler radar, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), IEEE, 2013, pp. 869–874.

[19] A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A.-A. Aghamohammadi, C. Heckman, Radar-inertial ego-velocity estimation for visually degraded environments, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2020, pp. 5739–5746.

[20] J. Michalczyk, R. Jung, S. Weiss, Tightly-coupled ekf-based radar-inertial odometry, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 12336–12343.

[21] J. Michalczyk, M. Scheiber, R. Jung, S. Weiss, Radar-inertial odometry for closed-loop control of resource-constrained aerial platforms (2023).

[22] H. Chen, Y. Liu, Y. Cheng, Drio: Robust radar-inertial odometry in dynamic environments, IEEE Robotics and Automation Letters (RA-L) (2023).

[23] Y. Zhuang, B. Wang, J. Huai, M. Li, 4d iriom: 4d imaging radar inertial odometry and mapping, IEEE Robotics and Automation Letters (RA-L) (2023).

[24] J. Zhang, H. Zhuge, Z. Wu, G. Peng, M. Wen, Y. Liu, D. Wang, 4dradarslam: A 4d imaging radar slam system for large-scale environments based on pose graph optimization, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 8333–8340.

[25] X. Li, H. Zhang, W. Chen, 4d radar-based pose graph slam with ego-velocity pre-integration factor, IEEE Robotics and Automation Letters (RA-L) (2023).

[26] V. Kubelka, E. Fritz, M. Magnusson, Do we need scan-matching in radar odometry?, arXiv preprint arXiv:2310.18117 (2023).

[27] M. Mostafa, S. Zahran, A. Moussa, N. El-Sheimy, A. Sesay, Radar and visual odometry integrated system aided navigation for uavs in gnss denied environment, Sensors 18 (2018).

[28] Y. Liang, S. Muller, D. Schwendner, D. Rolle, D. Ganesch, I. Schaffer, A scalable framework for robust vehicle state estimation with a fusion of a low-cost imu, the gnss, radar, a camera and lidar, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2020) 1661–1668.

[29] F. Ding, Z. Pan, Y. Deng, J. Deng, C. X. Lu, Self-supervised scene flow estimation with 4-d automotive radar, IEEE Robotics and Automation Letters (RA-L) 7 (3) (2022) 8233–8240.

[30] M. He, C. Zhu, Q. Huang, B. Ren, J. Liu, A review of monocular visual odometry, The Visual Computer 36 (5) (2020) 1053–1065.

[31] R. A. Newcombe, S. J. Lovegrove, A. J. Davison, Dtam: Dense tracking and mapping in real-time, IEEE International Conference on Computer Vision (ICCV) (2011) 2320–2327.

[32] R. Mur-Artal, J. D. Tardós, Probabilistic semi-dense mapping from highly accurate feature-based monocular slam, Robotics: Science and Systems (RSS) 11 (2015).

[33] C. Forster, M. Pizzoli, D. Scaramuzza, Svo: Fast semi-direct monocular visual odometry, IEEE International Conference on Robotics and Automation (ICRA) (2014) 15–22.

[34] R. Li, S. Wang, Z. Long, D. Gu, Undeepvo: Monocular visual odometry through unsupervised deep learning, in: 2018 IEEE international conference on robotics and automation (ICRA), IEEE, 2018, pp. 7286–7291.

[35] C. D. Monaco, Ego-motion estimation from doppler and spatial data in sonar, radar, or camera images, Ph.D. thesis (2019).

[36] Y.-E. Lu, S. Tsai, M.-L. Tsai, K.-W. Chiang, A low-cost visual radar-based odometry framework with mmwave radar and monocular camera, International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences (ISPRS) (2022).

[37] A. F. Scannapieco, A. Renga, M. D. Graziano, G. Fasano, Preliminary performance assessment of radar-aided monocular visual odometry for small aerial platforms, IEEE International Workshop on Metrology for AeroSpace, (MetroAeroSpace) (2019) 278–283.

[38] H. Bremer, Elastic multibody dynamics, Intelligent Systems, Control and Automation: Science and Engineering (ISCA) 13 (2008) 85–90.

[39] T. D. Barfoot, State estimation for robotics, Cambridge University Press, 2017.

[40] D. Koller, N. Friedman, Probabilistic graphical models: principles and techniques, MIT press, 2009.

[41] F. Dellaert, Factor graphs and gtsam: A hands-on introduction, Georgia Institute of Technology, Tech. Rep 2 (2012) 4.

[42] F. Dellaert, M. Kaess, et al., Factor graphs for robot perception, Foundations and Trends® in Robotics 6 (1-2) (2017) 1–139.

[43] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, F. Dellaert, isam2: Incremental smoothing and mapping using the bayes tree, The International Journal of Robotics Research 31 (2) (2012) 216–235.
[44] N. L. Baisa, Derivation of a constant velocity motion model for visual tracking, arXiv preprint arXiv:2005.00844 (2020).
[45] A. Segal, D. Haehnel, S. Thrun, Generalized-icp., in: Robotics: science and systems, Vol. 2, Seattle, WA, 2009, p. 435.
[46] P. J. Besl, N. D. McKay, Method for registration of 3-d shapes, in: Sensor fusion IV: control paradigms and data structures, Vol. 1611, Spie, 1992, pp. 586–606.
[47] P. Biber, W. Straßer, The normal distributions transform: A new approach to laser scan matching, in: Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(Cat. No. 03CH37453), Vol. 3, IEEE, 2003, pp. 2743–2748.
[48] M. A. Fischler, R. C. Bolles, Random sample consensus, Communications of the ACM 24 (1981) 381–395.
[49] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, K. Dietmayer, Instantaneous ego-motion estimation using doppler radar, IEEE Conference on Intelligent Transportation Systems, (ITSC) (2013) 869–874.
[50] J.-T. Huang, R. Xu, A. Hinduja, M. Kaess, Multi-radar inertial odometry for 3d state estimation using mmwave imaging radar, arXiv preprint arXiv:2311.08608 (2023).
[51] P. J. Huber, Robust estimation of a location parameter, in: Breakthroughs in statistics: Methodology and distribution, Springer, 1992, pp. 492–518.
[52] N. J. Abu-Alrub, N. A. Rawashdeh, Radar odometry for autonomous ground vehicles: A survey of methods and datasets, arXiv preprint arXiv:2307.07861 (2023).
[53] M. Grupp, evo: Python package for the evaluation of odometry and slam., https://github.com/MichaelGrupp/evo (2017).
[54] T. Qin, P. Li, S. Shen, Vins-mono: A robust and versatile monocular visual-inertial state estimator, IEEE Transactions on Robotics 34 (4) (2018) 1004–1020.
[55] E. Wise, Q. Cheng, J. Kelly, Spatiotemporal calibration of 3-d millimetre-wavelength radar-camera pairs, IEEE Transactions on Robotics 39 (6) (2023) 4552–4566.
[56] S. Wang, R. Clark, H. Wen, N. Trigoni, Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks, in: 2017 IEEE international conference on robotics and automation (ICRA), IEEE, 2017, pp. 2043–2050.
[57] W. Wang, Y. Hu, S. Scherer, Tartanvo: A generalizable learning-based vo, in: Conference on Robot Learning, PMLR, 2021, pp. 1761–1772.
[58] J. Zhang, S. Singh, Low-drift and real-time lidar odometry and mapping, Autonomous Robots 41 (2017) 401–416.