MOVRO: Loosely Coupled EKF-Based Monocular Visual Radar Odometry

Vlaho-Josip Štironja, Juraj Peršić, Ivan Marković, and Ivan Petrović[1]

University of Zagreb Faculty of Electrical Engineering and Computing,
Laboratory for Autonomous Systems and Mobile Robotics (LAMOR), Croatia
{name.surname} @fer.hr

Abstract. Reliable and accurate ego-motion estimation constitute an integral part of any autonomous system. To achieve this, it is essential to integrate sensors of complementary characteristics while also minimizing costs. The most often used ego-motion estimation setup usually consists of a monocular camera and an inertial measurement unit (IMU) which can often drift due to inaccuracy of modest performance IMUs. Inlieu of enhancing such setups with a higher-performing IMU, in this paper we propose to fuse the monocular camera with a radar sensor. Specifically, we combine measurements from a 3D milimeter-wave frequency modulated continuous wave radar and a monocular camera within an error-state extended Kalman filter framework. The filter was implemented using loose coupling that allows for an independent and scalable development of sensor models. We evaluated the proposed method against state-of-the-art radar inertial, visual inertial, and radar visual inertial odometries. The results show that the proposed method is comparable with other multi-domain solutions, especially in outdoor sequences.

Keywords: radar visual odometry, error state extended Kalman filter, multisensor fusion

1 Introduction

The cornerstone of successful navigation is the ability to reliably know the state of the autonomous system in an environment. Therefore, it is essential to combine sensors with complementary characteristics while keeping the cost-effectiveness of the sensor setup. Typically, the most cost-effective setup is the one that combines a camera and an inertial measurement unit (IMU) sensor. The next step is usually to add a lidar, but at additional higher cost. However, with the development of millimeter wave radar technology, these sensor are being introduced as an alternative in multisensor setups for ego-motion estimation, especially since they are often used in vehicles for front collision warning and adaptive cruise control. They also exhibit higher robustness to environmental factors such as smoke, rain, fog, and low-visibility conditions [1]. Moreover, in contrast to lidars, radar can output the relative velocity between the target and sensor, enabling measurements of ego-velocity without scan matching [2]. Nevertheless, radars have their own challenges such as comparably low resolution, multipath reflections, larger

noise, ghost targets etc. [3]. Fusion of radars and cameras is usually performed for object detection and tracking [4], and less often for ego-motion estimation [5]. Both sensors have their advantages and are highly complementary. For example, monocular cameras, i.e., monocular odometry, gives the possibility of global consistency and accurate orientation estimate but lacks metric scale (excluding deep approaches with monocular depth estimation). On the other hand, radar enables instantaneous ego-velocity estimates but cannot provide good orientation estimation due to error-prone scan matching. Given that, combination of the two sensors presents great opportunity to explore in the context of ego-motion estimation.

Visual ego-motion estimators can be divided into feature-based or direct visual odometries (VO). Open-source packages, e.g., LibViso [6] and ORB-SLAM [7] are examples of feature based VOs. In feature-based approaches the camera pose is usually solved for via the perspective-n-point projection problem. On the other hand, direct odometries, such as [8] and [9], optimize geometry directly on image intensity values without any feature extraction. Edge-based approaches [10], [11] are more robust to lightning changes, but often more computationally expensive. Authors in [12] combine the advantages of direct and feature-based VOs to improve tracking and mapping performance. More recent advances in monocular odometry leverage convolutional neural networks (CNN) [13].

Radar inertial sensor setup has attracted a lot of attention as it is shown to be highly robust to environmental conditions [14]. Authors in [1] implemented the first radar inertial ego-velocity estimation using a sliding window approach. The solution was proven to be comparable with the visual-inertial approaches in conditions favorable to visual-inertial odometry and more accurate in visually degraded environments. Authors of [2] and [15] fused 3D radar ego-velocity estimate with an IMU and a barometer (for reducing drift) within an ES-EKF framework. Radar inertial odometry within ES-EKF with online extrinsic calibration of radar was first introduced [2] extending previous work [15]. A tightly coupled EKF-based approach to radar inertial odometry that fuses IMU with both the ego-velocity estimate and distance measurements to 3D points was first introduced in [16]. 1D radar velocity, height, optical flow, GPS, and IMU data were loosely fused within an EKF in [17]. Authors in [18] presented loosely coupled ES-EKF-based fusion concept including 2D radar velocity, stereo pair of cameras, Lidar, and IMU data. State-of-the-art radar visual inertial and radar thermal inertial odometry was presented in [19]. Authors extended Robust Visual Inertial Odometry (ROVIO) [20] using radar drift-free 3D ego-velocity and outperformed radar inertial and monocular open-source odometries.

MonoRAD is a 2D decoupled ego-motion estimation algorithm based on radar and monocular camera sensor fusion where an instantaneous ego-velocity estimate is used to scale monocular odometry [5]. Similarly, authors in [21] fuse interpolated instantaneous 2D radar ego-velocity estimates with monocular ORB-SLAM outputs between consecutive key-frames to correct the scale. An alternative approach presented in [22] fuses monocular odometry with interpolated 2D radar ego-velocity estimates which are obtained using scan matching. In addi-

tion, the method assumes radar's narrow elevation field of view while applying the two-dimensional hypothesis. While the method includes radar-estimated rotation, it's biggest disadvantage is evaluating translation using scan matching with radar which can be error-prone due to low resolution, larger noise, multi-path propagation, and often a small number of targets in a frame.

In this paper, we propose a novel Monocular Visual Radar Odometry (MOVRO) framework for ego-motion estimation that combines complementary characteristics of a monocular camera and a radar. To the best of our knowledge, this is the first method that fuses instantaneous 3D radar ego-velocity estimates and monocular visual odometry poses. The proposed approach is formulated in a loosely coupled fashion within an Error State Extended Kalman Filter (ES-EKF), thus enabling usage of any monocular visual odometry. The ES-EKF was chosen as orientation error state is minimal thus avoiding issues related to over-parameterization and risks of singularity [23]. As the error state is always small, all second-order products are negligible which makes Jacobians easy and fast to compute [24]. Unlike the radar inertial odometry state estimator (EKF-RIO) [2], where the fusion of barometric height measurements within the filter was implemented in order to reduce the drift along the height, in this sensor setup, we propose correction using scaled monocular ego-velocity. We compared MOVRO to EKF-RIO without the barometer [2], ROVIO [20], as well with radar monocular visual inertial odometry (RVIO)[19]. For radar ego-velocity estimation, we use *reve* [15], while for monocular odometry ORB-SLAM [7] was used without loop closures. The results show that the proposed approach can properly scale motion estimated with a monocular camera and compensate for radar's inaccurate rotation estimation, in effect yielding comparable results with state-of-the-art dual-domain sensor solutions and even outperforming them at outdoor sequences.

2 Proposed Method

The following notation is used in the paper. Scalar is written in lower case, non-bold letters, e.g, m. Constants are written as non-bold upper case letters, e.g., M, and matrices as bold upper case letters, e.g., $\mathbf{M}$. Vectors are printed as bold lower case letters, e.g., $\mathbf{m}$. Rotation and vector writing form is taken from the [25]. Expression ${}_{\mathrm{A}}\mathbf{t}_{\mathrm{BC}}$ represents vector from frame $\mathcal{F}_{\mathrm{B}}$ to $\mathcal{F}_{\mathrm{C}}$ expressed in $\mathcal{F}_{\mathrm{A}}$. Rotation matrix $\mathbf{R}_{\mathrm{AB}}$ is a rotation of frame $\mathcal{F}_{\mathrm{B}}$ with respect to the $\mathcal{F}_{\mathrm{A}}$, i.e., $\mathbf{R}_{\mathrm{AB}}$ transforms a vector ${}_{\mathrm{B}}\mathbf{v}$ from $\mathcal{F}_{\mathrm{B}}$ into $\mathcal{F}_{\mathrm{A}}$ (${}_{\mathrm{A}}\mathbf{v}$).

MOVRO estimates the state using a loosely coupled ES-EKF. The state is propagated using the constant velocity model and corrected with ego-velocity measurements from monocular odometry and radar as well with orientation from the monocular odometry. Therefore, extrinsic calibration between radar and camera is required prior to running the proposed method.

2.1 Modelling the system – prediction step

In ES-EKF terminology we differentiate the nominal, error, and true state. The nominal state, at time instance k, $\mathbf{x}_k$ can be treated as a large quantity, while the error state $\delta\mathbf{x}_k$ can be viewed as a small quantity. Thus, the error state

is linearly integrable which enables linear-Gaussian filtering [24]. The nominal state does not take into account an inadequate process model and noise terms, so it accumulates error over time which is estimated with ES-EKF. The correction step provides error state estimate which is later injected into a nominal state, while the error state is reset to zero. The composition of error state and nominal state is called the true state $\hat{\mathbf{x}}_k$. Note that the true state estimate is the same as nominal state before correction as error state is not observed yet. A more detailed explanation is given in [24].

In our case the system state at step k is defined as:

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

where $\mathbf{q}_{\mathrm{WC},k}$, ${}_{\mathrm{C}}\boldsymbol{\omega}_k$, ${}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k}$, ${}_{\mathrm{W}}\mathbf{v}_k$, and r_k stand for orientation presented as a quaternion, angular velocity, position, camera velocity, and scale factor. Frames $\mathcal{F}_{\mathrm{C}}$ and $\mathcal{F}_{\mathrm{W}}$ stand for the camera and world frame, respectively. World frame is initiated at the initial step of the filter. The nominal state kinematics are given as follows:

$$\mathbf{q}_{\mathrm{WC},k} = \mathbf{q}_{\mathrm{WC},k-1} \otimes \mathbf{q}\{{}_{\mathrm{C}}\boldsymbol{\omega}_{k-1}\Delta t\}, \tag{2}$$

$${}_{\mathrm{C}}\boldsymbol{\omega}_k = {}_{\mathrm{C}}\boldsymbol{\omega}_{k-1}, \tag{3}$$

$${}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k} = {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k-1} + {}_{\mathrm{W}}\mathbf{v}_{k-1}\Delta t, \tag{4}$$

$${}_{\mathrm{W}}\mathbf{v}_k = {}_{\mathrm{W}}\mathbf{v}_{k-1}, \tag{5}$$

$$r_k = r_{k-1}, \tag{6}$$

where Δt represents the time difference and $\mathbf{q}\{\mathbf{v}\}$ is the quaternion associated to the rotation vector $\mathbf{v}$. Error state at step k has the following notation:

$$\delta\mathbf{x}_k = \begin{bmatrix} \delta\boldsymbol{\theta}_{\mathrm{WC},k} & \delta_{\mathrm{C}}\boldsymbol{\omega}_k & \delta_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k} & \delta_{\mathrm{W}}\mathbf{v}_k & \delta r_k \end{bmatrix}. \tag{7}$$

The error state angle-axis notation is used due to its minimal rotation representation. This avoids the risk of parameter singularities such as gimbal-lock, due to the small magnitude of the error state [24]. The angular error $\delta\boldsymbol{\theta}_{\mathrm{WC}}$ is specified *locally* with respect to the nominal orientation in same way as in [24].

The error state dynamic system is given as:

$$\delta\boldsymbol{\theta}_{\mathrm{WC},k} = \mathbf{R}^{\mathrm{T}}\{{}_{\mathrm{C}}\boldsymbol{\omega}_{k-1}\Delta t\}\delta\boldsymbol{\theta}_{\mathrm{WC},k-1} + \delta_{\mathrm{C}}\boldsymbol{\omega}_{k-1}\Delta t + \mathbf{w}_{\alpha,k-1}\frac{\Delta t^2}{2}, \tag{8}$$

$$\delta_{\mathrm{C}}\boldsymbol{\omega}_k = \delta_{\mathrm{C}}\boldsymbol{\omega}_{k-1} + \mathbf{w}_{\alpha,k-1}\Delta t, \tag{9}$$

$$\delta_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k} = \delta_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k-1} + \delta_{\mathrm{W}}\mathbf{v}_{k-1}\Delta t + \mathbf{w}_{a,k-1}\frac{\Delta t^2}{2}, \tag{10}$$

$$\delta_{\mathrm{W}}\mathbf{v}_k = \delta_{\mathrm{W}}\mathbf{v}_{k-1} + \mathbf{w}_{a,k-1}\Delta t, \tag{11}$$

$$\delta r_k = \delta r_{k-1} + \mathrm{w}_{r,k-1}\Delta t. \tag{12}$$

The model of the error state is as follows:

$$\delta\mathbf{x}_k = f(\mathbf{x}_{k-1}, \delta\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) = \mathbf{F}_{k-1}(\mathbf{x}_{k-1})\delta\mathbf{x}_{k-1} + \mathbf{G}_{k-1}\mathbf{w}_{k-1}, \tag{13}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^{\mathrm{T}} + \mathbf{Q}_{k-1}, \tag{14}$$

while process noise $\mathbf{w}_{k-1}$ is defined as

$$\mathbf{w}_{k-1} = \begin{bmatrix}\mathbf{w}_{\alpha,k-1} & \mathbf{w}_{a,k-1} & \mathrm{w}_{r,k-1}\end{bmatrix}, \tag{15}$$

where $\mathbf{w}_{\alpha,k-1}$, $\mathbf{w}_{a,k-1}$, and $\mathrm{w}_{r,k-1}$ stand for constant white angular acceleration, translational acceleration and scale factor described by zero-mean Gaussian white noise. The process noise acts here as relaxation via piecewise constant translational and rotational acceleration [26].

The system model Jacobian and process noise matrix are given as:

$$\mathbf{F}_{k-1} = \frac{\partial f}{\partial \delta x} = \begin{bmatrix} \mathbf{R}^{\mathrm{T}}\{{}_{\mathrm{C}}\boldsymbol{\omega}_{k-1}\Delta t\} & \mathrm{I}_{3\times3}\Delta t & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & \mathrm{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & \mathrm{I}_{3\times3} & \mathrm{I}_{3\times3}\Delta t & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathrm{I}_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 1 \end{bmatrix}, \tag{16}$$

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \boldsymbol{\sigma}_{\alpha}^2 \frac{\Delta t^4}{4}\mathrm{I}_{3\times3} & \boldsymbol{\sigma}_{\alpha}^2 \frac{\Delta t^3}{2}\mathrm{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ \boldsymbol{\sigma}_{\alpha}^2 \frac{\Delta t^3}{2}\mathrm{I}_{3\times3} & \boldsymbol{\sigma}_{\alpha}^2 \Delta t^2\mathrm{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & \boldsymbol{\sigma}_{a}^2 \frac{\Delta t^4}{4}\mathrm{I}_{3\times3} & \boldsymbol{\sigma}_{a}^2 \frac{\Delta t^3}{2}\mathrm{I}_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & \boldsymbol{\sigma}_{a}^2 \frac{\Delta t^3}{2}\mathrm{I}_{3\times3} & \boldsymbol{\sigma}_{a}^2 \Delta t^2\mathrm{I}_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & \sigma_r^2 \Delta t^2 \end{bmatrix}, \tag{17}$$

where $\boldsymbol{\sigma}_{\alpha}^2$, $\boldsymbol{\sigma}_{\mathrm{a}}^2$, and σ_r^2 stand for angular acceleration, translational acceleration and scale factor variances respectively. The aforementioned expressions were derived in a similar manner as in [26].

2.2 Modelling the measurements – correction step

In this section we present the correction step of our framework, where we use the baseline Kalman filter equations as follows:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^{\mathrm{T}} + \mathbf{V}_k)^{-1}, \tag{18}$$

$$\delta\hat{\mathbf{x}}_k = \mathbf{K}_k(\mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})), \tag{19}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k) + \mathbf{K}_k\mathbf{V}_k\mathbf{K}_{\mathrm{k}}^{\mathrm{T}}, \tag{20}$$

where $\mathbf{y}_k$ is sensor measurement at step k, matrix $\mathbf{V}_k$ is covariance of white Gaussian measurement noise, $\delta\hat{\mathbf{x}}_k$ is the corrected error state estimate at step k, and $\hat{\mathbf{x}}_{k|k-1}$ is the true state estimate before correction, while measurement equation is given as follows:

$$h(\hat{\mathbf{x}}_{k|k-1}) = \begin{cases} h_{\mathrm{C}}(\hat{\mathbf{x}}_{k|k-1}) & \text{camera model} \\ h_{\mathrm{R}}(\hat{\mathbf{x}}_{k|k-1}) & \text{radar model.} \end{cases} \tag{21}$$

Note that true state estimate is the same as nominal state before correction as error state is not observed yet.

After the correction, the true state $\hat{\mathbf{x}}_k$ is the current estimate at step k of the state:

$$\hat{\mathbf{x}}_k = \begin{bmatrix} \hat{\mathbf{q}}_{\mathrm{WC},k} & {}_{\mathrm{C}}\hat{\boldsymbol{\omega}}_k & {}_{\mathrm{W}}\hat{\mathbf{p}}_{\mathrm{WC},k} & {}_{\mathrm{W}}\hat{\mathbf{v}}_k & \hat{r}_k \end{bmatrix}. \tag{22}$$

The true state is determined as a composition of the nominal state and the error state by means of the following equations:

$$\hat{\mathbf{q}}_{\mathrm{WC},k} = \mathbf{q}_{\mathrm{WC},k} \cdot \mathbf{q}\{\delta\hat{\boldsymbol{\theta}}_{\mathrm{WC},k}\}, \tag{23}$$

$${}_{\mathrm{C}}\hat{\boldsymbol{\omega}}_k = {}_{\mathrm{C}}\boldsymbol{\omega}_k + \delta_{\mathrm{C}}\hat{\boldsymbol{\omega}}_k, \tag{24}$$

$${}_{\mathrm{W}}\hat{\mathbf{p}}_{\mathrm{WC},k} = {}_{\mathrm{W}}\mathbf{p}_{\mathrm{WC},k} + \delta_{\mathrm{W}}\hat{\mathbf{p}}_{\mathrm{WC},k}, \tag{25}$$

$${}_{\mathrm{W}}\hat{\mathbf{v}}_k = {}_{\mathrm{W}}\mathbf{v}_k + \delta_{\mathrm{W}}\hat{\mathbf{v}}_k, \tag{26}$$

$$\hat{r}_k = r_k + \delta\hat{r}_k. \tag{27}$$

In order to derive the Jacobian matrices H_k, we use the chain rule:

$$\mathbf{H}_k = \left.\frac{\partial h}{\partial \delta \mathbf{x}_k}\right|_{\mathbf{x}_k} = \left.\frac{\partial h}{\partial \hat{\mathbf{x}}_{k|k-1}}\right|_{\mathbf{x}_k} \left.\frac{\partial \hat{\mathbf{x}}_{k|k-1}}{\partial \delta \mathbf{x}_k}\right|_{\mathbf{x}_k}. \tag{28}$$

The first part of the Jacobian $\frac{\partial h}{\partial \hat{\mathbf{x}}_{k|k-1}}$ depends on the measurement function of the sensor with respect to the true state before correction at the true state estimate which is identical to the nominal state, since the mean value of the error state before correction is zero. The second part depends on the true state with respect to the error state derived in [24] as:

$$\left.\frac{\partial \hat{\mathbf{x}}_{k|k-1}}{\partial \delta \mathbf{x}_k}\right|_{\mathbf{x}_k} = \begin{bmatrix} \left.\frac{\partial \mathbf{q}_k \otimes \delta\mathbf{q}(\delta\boldsymbol{\theta}_k)}{\partial \delta\boldsymbol{\theta}_k}\right|_{\mathbf{x}_k} & 0_{4\times3} & 0_{4\times3} & 0_{4\times3} & 0_{4\times1} \\ 0_{3\times3} & \mathrm{I}_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & \mathrm{I}_{3\times3} & 0_{3\times3} & 0_{3\times1} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & \mathrm{I}_{3\times3} & 0_{3\times1} \\ 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 0_{1\times3} & 1 \end{bmatrix}, \tag{29}$$

$$\left.\frac{\partial \mathbf{q}_k \otimes \delta\mathbf{q}(\delta\boldsymbol{\theta}_k)}{\partial \delta\boldsymbol{\theta}_k}\right|_{\mathbf{x}_k} = \frac{1}{2}\begin{bmatrix} -q_{x,k} & -q_{y,k} & -q_{z,k} \\ q_{w,k} & -q_{z,k} & q_{y,k} \\ q_{z,k} & q_{w,k} & -q_{x,k} \\ -q_{y,k} & q_{x,k} & q_{w,k} \end{bmatrix}. \tag{30}$$

In the following section we show the derivation of the innovation term $\mathbf{e}_k = \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1})$ and first part $\frac{\partial h}{\partial \hat{\mathbf{x}}_{k|k-1}}$ of the Jacobian $\mathbf{H}_k$ for both sensor corrections.

2.3 Radar-based filter correction

Radar's ego-velocity is estimated using the open-source package *reve* [2]. It assumes that all radar detections $\mathrm{D}_{\mathrm{i},k}$ ($i \in (1, M)$) within a single scan at time instance k are static and contain doppler-based range-rate measurement $v_{\mathrm{D_{i,k}}}$ and 3D object position ${}_{\mathrm{R}}\mathbf{p}_{\mathrm{RDi},k}$ expressed in $\mathcal{F}_{\mathrm{R}}$. Each valid static raw radar measurement abides to the following expression:

$$-v_{\mathrm{D}_{\mathrm{i},k}} = \frac{{}_{\mathrm{R}}\mathbf{p}_{\mathrm{RDi},k}}{||{}_{\mathrm{R}}\mathbf{p}_{\mathrm{RDi},k}||} \cdot {}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}, \tag{31}$$

where ${}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}$ is a 3D translational radar velocity at time instance k. With at least three non-coplanar raw radar measurements, it is possible to recover a full 3D translational radar velocity ${}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}$. Inliers can be determined with the use of the three-point RANSAC approach [27], while the final refinement is done using orthogonal distance regression as proposed in [28]. We use ${}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}$ as the measurement in the proposed filter.

To derive the innovation term, we need to transform the true state velocity from $\mathcal{F}_{\mathrm{W}}$ to $\mathcal{F}_{\mathrm{R}}$.

$$ {}_{\mathrm{R}}\hat{\mathbf{v}}_{k|k-1} = \mathbf{R}_{\mathrm{RC}}(\hat{\mathbf{R}}_{\mathrm{CW},k|k-1}\,{}_{\mathrm{W}}\hat{\mathbf{v}}_{k|k-1} + {}_{\mathrm{C}}\hat{\boldsymbol{\omega}}_{k|k-1} \times {}_{\mathrm{C}}\mathbf{l}_{\mathrm{RC}}). \tag{32}$$

Transformation between the $\mathcal{F}_{\mathrm{W}}$ and $\mathcal{F}_{\mathrm{R}}$ assumes apriori known, constant extrinsic calibration as in [2]. The radar innovation term at step k ($\mathbf{e}_{R,k}$) evaluates to:

$$\mathbf{e}_{R,k} = y_{R,k} - {}_{\mathrm{R}}\hat{\mathbf{v}}_{k|k-1}, \tag{33}$$

where the radar measurement $y_{R,k}$ is defined as $y_{R,k} = {}_{\mathrm{R}}\mathbf{v}_{\mathrm{R},k}$. The Jacobian matrix $\mathbf{H}_{R,k}$ is evaluated as:

$$\mathbf{H}_{R,k} = \left.\frac{\partial h_R}{\partial \hat{\mathbf{x}}_{k|k-1}}\right|_{\mathbf{x}_k} \left.\frac{\partial \hat{\mathbf{x}}_{k|k-1}}{\partial \delta \mathbf{x}_k}\right|_{\mathbf{x}_k}, \tag{34}$$

where first part of the Jacobian is derived as follows:

$$\frac{\partial h_R}{\partial \hat{\mathbf{x}}_{k|k-1}} = \begin{bmatrix} 0_{3\times 4} & \mathbf{R}_{\mathrm{RC}}[{}_{\mathrm{C}}\mathbf{l}_{\mathrm{RC}}]_\times & 0_{3\times 3} & \mathbf{R}_{\mathrm{RC}}\mathbf{R}_{\mathrm{CW},k} & 0 \end{bmatrix}, \tag{35}$$

where skew-symmetric operator is defined as

$$[\mathbf{x}]_\times = \begin{bmatrix} 0 & -\mathrm{x}_3 & -\mathrm{x}_2 \\ -\mathrm{x}_3 & 0 & -\mathrm{x}_1 \\ -\mathrm{x}_2 & \mathrm{x}_1 & 0 \end{bmatrix}. \tag{36}$$

2.4 Visual odometry-based filter correction

In this case, when the monocular odometry update arrives, double correction is conducted using unscaled velocity and orientation. Preserved scale factor in the state enables correction with monocular unscaled velocity to reduce drift. Monocular odometry innovation term is:

$$\mathbf{e}_{C,k} = y_{C,k} - h_C(\hat{\mathbf{x}}_{k|k-1}), \tag{37}$$

where the monocular odometry measurement $y_{C,k}$ is defined as

$$y_{C,k} = \begin{bmatrix} \boldsymbol{\theta}_{\mathrm{WC},k} \\ {}_{\mathrm{W}}\mathbf{v}_{M,k} \end{bmatrix} \tag{38}$$

and $\boldsymbol{\theta}_{\mathrm{WC},k}$ and ${}_{\mathrm{W}}\mathbf{v}_{M,k}$ stand for monocular orientation and unscaled ego-velocity estimates. Given that, the innovation term is as follows:

$$\mathbf{e}_C = \begin{bmatrix} [\log(\mathrm{R}^{\mathrm{T}}\{\hat{\mathbf{q}}_{\mathrm{WC},k|k-1}\}\mathrm{R}\{\boldsymbol{\theta}_{\mathrm{WC},k}\})]^\vee \\ {}_{\mathrm{W}}\mathbf{v}_{M,k} - \frac{1}{\hat{r}_{k|k-1}}\,{}_{\mathrm{W}}\hat{\mathbf{v}}_{k|k-1} \end{bmatrix}, \tag{39}$$

where *log* and *vee* maps represent mappings from the Lie group SO(3) to Lie algebra $\mathfrak{so}(3)$, and the pertaining Euclidean space [29]. We can now form the Jacobian component, similarly as in [24]:

$$\mathbf{H}_{C,k} = \left.\frac{\partial h_C}{\partial \hat{\mathbf{x}}_{k|k-1}}\right|_{\mathbf{x}_k} \left.\frac{\partial \hat{\mathbf{x}}_{k|k-1}}{\partial \delta \mathbf{x}_k}\right|_{\mathbf{x}_k}, \tag{40}$$

where the first part of the Jacobian is derived as

$$\frac{\partial h_C}{\partial \hat{\mathbf{x}}_{k|k-1}} = \begin{bmatrix} 2\begin{bmatrix} -q_{x,k} & q_{w,k} & q_{z,k} & -q_{y,k} \\ -q_{y,k} & -q_{z,k} & q_{w,k} & q_{x,k} \\ -q_{z,k} & q_{y,k} & -q_{x,k} & q_{w,k} \end{bmatrix} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 1} \\ 0_{3\times 4} & & 0_{3\times 3} & 0_{3\times 3} & \frac{1}{r_k}\mathrm{I}_{3\times 3} & -\frac{{}^{\mathrm{W}}\mathbf{v}_k}{r_k^2} \end{bmatrix}, \tag{41}$$

while the second part of the Jacobian is the same as in (29).

2.5 ES-EKF reset

After injecting the error state vector into the nominal state, the error state $\delta\hat{\mathbf{x}}$ gets reset which is especially important for the orientation error that is represented locally [24]. We can define the reset function $\mathcal{R}$ as:

$$\delta\mathbf{x} = \mathcal{R}(\delta\mathbf{x}) = \delta\mathbf{x} \ominus \delta\hat{\mathbf{x}}, \tag{42}$$

where operator $\ominus$ stands for decomposition, i.e., composition inverse of $\oplus$. Therefore, we write covariance reset as:

$$\mathbf{P}_{k|k} = \mathbf{R}_k \mathbf{P}_{k|k} \mathbf{R}_k^{\mathbf{T}}, \tag{43}$$

where matrix $\mathbf{R}_k$ at step k is defined as:

$$\mathbf{R}_k = \left.\frac{\partial \mathcal{R}}{\partial \delta \mathbf{x}_k}\right|_{\delta\hat{\mathbf{x}}}, \tag{44}$$

$$\mathbf{R}_k = \begin{bmatrix} \mathrm{I}_{3\times 3} - [\frac{1}{2}\delta\hat{\boldsymbol{\theta}}_{\mathrm{WC},k}]_\times & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 1} \\ 0_{3\times 3} & \mathrm{I}_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 1} \\ 0_{3\times 3} & 0_{3\times 3} & \mathrm{I}_{3\times 3} & 0_{3\times 3} & 0_{3\times 1} \\ 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & \mathrm{I}_{3\times 3} & 0_{3\times 1} \\ 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 0_{1\times 3} & 1 \end{bmatrix}. \tag{45}$$

The complete derivation of matrix $\mathbf{G}_k$ can be found in [24].

3 Experimental Results

The proposed method was evaluated using the IRS dataset [19] and compared to several competing approaches. The dataset contains measurements from a high-performance IMU (Analog Devices ADIS16448), an FCMW radar (Texas Instruments IWR6843AOP), and monocular grayscale camera (IDS UI-3241). All the sensors were synchronized using a hardware trigger and sync signals via

microcontroller board. ORB-SLAM3 [7] without loop closures in offline mode was used to obtain monocular odometry estimates that we used within MOVRO. For radar ego-velocity estimates the open-source package *reve* [15] was used. The implementation of MOVRO operates in a real-time, under the assumption that the messages obtained from the monocular visual odometry and the radar ego-velocity estimates are delivered in real time. MOVRO is evaluated on six from a total of nine dataset sequences. The remaining three sequences caused ORB-SLAM3 to fail due to rapid motions (Mocap_difficult) or poor lightning (Mocap_dark, Mocap_dark_fast), as reported in [30].

3.1 Validation

For the odometry evaluation metric, we concentrated on the absolute translational error (ATE) as our implementation focuses on scaling monocular odometry. We compared MOVRO to (i) monocular visual inertial odometry ROVIO [20] using default 25 features in the filter state, (ii) radar inertial odometry EKF-RIO [2] and (iii) radar visual inertial odometry RVIO [19]. EKF-RIO was evaluated without the barometer in order to ensure fair comparison. ROVIO, EKF-RIO, and RVIO parameters were taken from [19]. The results are shown in Table 3.1 using the evaluation tool presented in [31], where it can be seen that MOVRO significantly outperforms other dual-domain solutions in outdoor sequences (Outdoor_campus and Outdoor_street) and has similar accuracy as RVIO that is state-of-the-art three-domain solution. Figure 2 show translational and rotational error comparison between odometry results using MOVRO with and without additional correction ((38) without ${}_{\mathrm{W}}\mathbf{v}_{M,k}$) with monocular ego-velocity where MOVRO without additional correction is labeled as MOVRO-. Respectively, MOVRO- innovation term in monocular odometry-based filter correction does not contain monocular unscaled ego-velocity residual vector. We can see that velocity correction additionally reduces translational drift while rotation error stays the same.

Based on the results, the following key insights are stated. Loosely coupled EKF-based monocular camera radar system fusion using an adequate motion model can produce similar results, and in some cases outperform state-of-the-art radar-inertial and visual-inertial solutions without the use of an expensive

	M	R	I	Gym	Indoor_floor	Mo_easy	Mo_med	Out_cam	Out_street
EKF-RIO		✓	✓	**0.10**	0.86	0.12	0.13	0.48	1.59
ROVIO	✓		✓	0.23	0.48	**0.08**	**0.10**	1.39	4.70
MOVRO	✓	✓		0.11	0.73	0.11	0.20	**0.26**	0.631
RVIO	✓	✓	✓	0.11	**0.31**	0.08	0.11	0.31	**0.44**

Table 1. Absolute translational errors (ATE) [m] [31]. Evaluation has been done aligning estimated trajectories using *se3* alignment with groundtruth. The best results of each dataset sequence are marked in bold and all values are rounded to two decimal points. M, R, and I symbols stand for the monocular camera, radar, and IMU sensor. Complete names of evaluated sequences are the following: Gym, Indoor_floor, Mocap_easy, Mocap_medium, Outdoor_campus, and Outdoor_street.

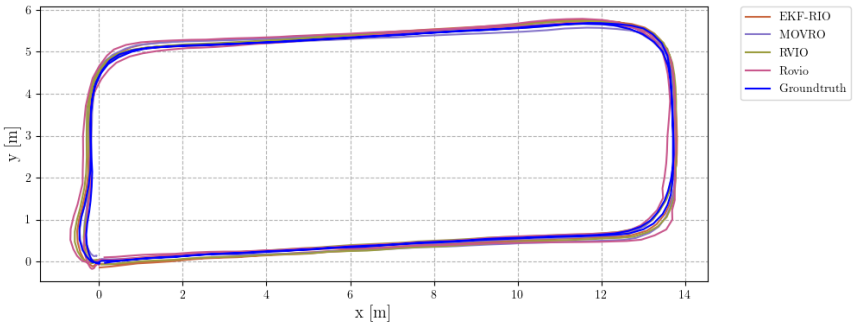


Fig. 1. Odometry comparison on the Gym dataset sequence. The visualization is made using *rpg_trajectory_evaluation* [31] with *se3* trajectory alignment.

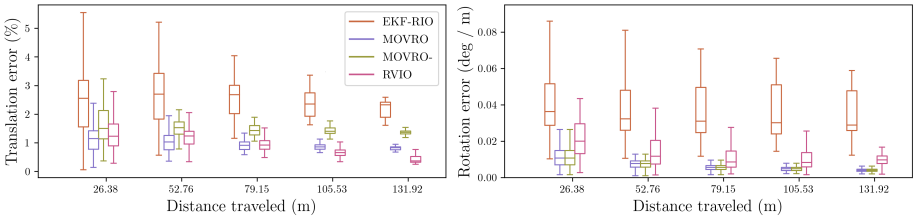


Fig. 2. Translational and rotational error comparison on Outdoor_street dataset sequence. The visualization is made using *rpg_trajectory_evaluation* [31] with *se3* trajectory alignment.

high-performance IMU. Having a camera instead of a high-performance IMU for orientation estimate can add the possibility to achieve global consistency, i.e. perform loop closure. MOVRO's loosely coupled approach enables the incorporation of any monocular state solution as well as any sensor that can estimate ego-velocity and independent development of each sensor model. The same principle can thus be used for any sensor setup consisting of a monocular camera and sensor that can estimate ego-velocity, e.g. Doppler Velocity Logger (DVL).

4 Conclusion

In this paper, we have presented a loosely coupled EKF-based monocular visual radar ego-motion estimator that we dubbed MOVRO. The novelty of our approach is using 3D radar and camera ego-velocity estimate to improve the scale of monocular visual odometry by including scale factor in the ES-EKF state. The proposed approach was compared against state-of-the-art radar inertial, visual inertial, and radar visual inertial odometry using an open-source dataset. In outdoor sequences, MOVRO outperforms other dual-domain solutions, while using a more economic sensor setup. Our solution is general and allows integration with other sensor setups that provide ego-velocity and orientation measurements,

such as a DVL-camera system fusion. Additionally, MOVRO's loosely coupled approach enables an independent and scalable framework.

Future work could include additional orientation correction using radar data as well as enabling radar online extrinsic calibration.

Acknowledgments

This research has been supported by the European Regional Development Fund under the grant KK.01.2.1.02.0119 (A-UNIT).

References

1. A. Kramer, C. Stahoviak, A. Santamaria-Navarro, A. A. Agha-Mohammadi, and C. Heckman, "Radar-inertial ego-velocity estimation for visually degraded environments," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5739–5746, 2020.
2. C. Doer and G. F. Trommer, "Radar inertial odometry with online calibration," *European Navigation Conference (ENC)*, 2020.
3. M. Adams, J. Mullane, J. Ebi, and B.-N. Vo, *Robotic Navigation and Mapping with Radar*. Artech House, 2012.
4. R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," *IEEE Winter Conference on Applications of Computer Vision, (WACV)*, 2021.
5. C. D. Monaco, "Ego-motion estimation from doppler and spatial data in sonar, radar, or camera images," Ph.D. dissertation, 2019.
6. A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," *IEEE Intelligent Vehicles Symposium (IV)*, pp. 963–968, 2011.
7. C. Campos, R. Elvira, J. J. Rodriguez, J. M. Montiel, and J. D. Tardos, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, pp. 1874–1890, 2021.
8. R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," *IEEE International Conference on Computer Vision (ICCV)*, pp. 2320–2327, 2011.
9. R. Mur-Artal and J. D. Tardós, "Probabilistic semi-dense mapping from highly accurate feature-based monocular slam," *Robotics: Science and Systems (RSS)*, vol. 11, 2015.
10. G. Klein and D. Murray, "Improving the agility of keyframe-based slam," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5303 LNCS, 2008.
11. E. Eade and T. Drummond, "Edge landmarks in monocular slam," *Image and Vision Computing (IMAVIS)*, vol. 27, 2009.
12. C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22, 2014.
13. K. Koledić, I. Cvišić, I. Marković, and I. Petrović, "Moft: Monocular odometry based on deep depth and careful feature selection and tracking," *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
14. J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H. L. Bloecher, K. Werber, and A. Sailer, "Automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding," *IEEE Radar Conference, (RadarConf)*, 2016.

15. C. Doer and G. F. Trommer, "An ekf based approach to radar inertial odometry," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 152–159, 2020.
16. J. Michalczyk, R. Jung, and S. Weiss, "Tightly-coupled ekf-based radar-inertial odometry," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 12 336–12 343, 2022.
17. M. Mostafa, S. Zahran, A. Moussa, N. El-Sheimy, and A. Sesay, "Radar and visual odometry integrated system aided navigation for uavs in gnss denied environment," *Sensors*, vol. 18, 2018.
18. Y. Liang, S. Muller, D. Schwendner, D. Rolle, D. Ganesch, and 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)*, pp. 1661–1668, 2020.
19. C. Doer and 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)*, pp. 331–338, 2021.
20. M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research (IJRR)*, vol. 36, pp. 1053–1072, 2017.
21. Y.-E. Lu, S. Tsai, M.-L. Tsai, and 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.
22. A. F. Scannapieco, A. Renga, M. D. Graziano, and G. Fasano, "Preliminary performance assessment of radar-aided monocular visual odometry for small aerial platforms," *IEEE International Workshop on Metrology for AeroSpace, (MetroAeroSpace)*, pp. 278–283, 2019.
23. V. K. Madyastha, V. C. Ravindray, S. Mallikarjunan, and A. Goyal, "Extended kalman filter vs. error state kalman filter for aircraft attitude estimation," *AIAA Guidance, Navigation, and Control Conference*, 2011.
24. J. Solà, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.
25. H. Bremer, "Elastic multibody dynamics," *Intelligent Systems, Control and Automation: Science and Engineering (ISCA)*, vol. 13, pp. 85–90, 2008.
26. N. L. Baisa, "Derivation of a constant velocity motion model for visual tracking," *arXiv preprint arXiv:2005.00844*, 2020.
27. M. A. Fischler and R. C. Bolles, "Random sample consensus," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.
28. D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using doppler radar," *IEEE Conference on Intelligent Transportation Systems, (ITSC)*, pp. 869–874, 2013.
29. T. D. Barfoot, *State estimation for robotics*. Cambridge University Press, 2017.
30. E. Wise, Q. Cheng, and J. Kelly, "Spatiotemporal calibration of 3d mm-wavelength radar-camera pairs," *arXiv preprint arXiv:2211.01871*, 2022.
31. Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7244–7251, 2018.