Gupta *et al.*
EURASIP Journal on Advances in Signal Processing (2023) 2023:100
https://doi.org/10.1186/s13634-023-01062-7

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Urban localization using robust filtering at multiple linearization points

Shubh Gupta[1], Adyasha Mohanty[2] and Grace Gao[2*]

*Correspondence: gracegao@stanford.edu

[1] Department of Electrical Engineering, Stanford University, Stanford, CA, USA
[2] Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, USA

Abstract

We propose a robust Bayesian filtering framework for state and multi-modal uncertainty estimation in urban settings by fusing diverse sensor measurements. Our framework addresses multi-modal uncertainty from various error sources by tracking a separate probability distribution for linearization points corresponding to dynamics, measurements, and cost functions. Multiple parallel robust Extended Kalman filters (R-EKF) leverage these linearization points to characterize the state probability distribution. Employing Rao–Blackwellization, we combine the linearization point distribution with the state distribution, resulting in a unified, efficient, and outlier-resistant Bayesian filter that captures multi-modal uncertainty. Furthermore, we introduce a gradient descent-based optimization method to refine the filter parameters using available data. Evaluating our filter on real-world data from a multi-sensor setup comprising camera, Global Navigation Satellite System (GNSS), and Attitude and Heading Reference System (AHRS) demonstrates improved performance in bounding position errors based on uncertainty, while maintaining competitive accuracy and comparable computation to existing methods. Our results suggest that our framework is a promising direction for safe and reliable localization in urban environments.

Keywords: GNSS, Camera, Multi-sensor, Multi-modal uncertainty, Bayesian filtering, Robust estimation, Particle filter, Kalman filter, Rao–Blackwellization, Safety-critical localization

1 Introduction

In complex urban environments, accurate estimation of both the state and its uncertainty is essential for ensuring the safe navigation of vehicles. The dynamic and unpredictable nature of these environments creates numerous sources of errors that can affect measurements from different sensors. For instance, Global Navigation Satellite System (GNSS) measurements are susceptible to bias error from multipath and non-line-of-sight signals in environments with tall structures [1]. Similarly, visual odometry measurements are influenced by dynamic objects and changes in lighting conditions [2]. Failing to account for these errors can lead to inaccurate or even misleading estimates of the state and uncertainty, which can have serious consequences for safety.

To address these challenges inherent in urban navigation, many systems rely on multi-sensor integration [3, 4]. These systems fuse data from a diverse array of complementary

sensors, thereby mitigating individual sensor limitations and leading to a more reliable estimate of the state. However, integrating multi-sensor data presents its own set of challenges, such as the need to mitigate large errors in sensor measurements and accommodate diverse noise sources [5, 6]. These diverse noise sources contribute to multiple modes of uncertainty that may yield multiple plausible solutions for the state. Hence, algorithms for multi-sensor state estimation must have the dual capability of being robust to large errors while accurately capturing the underlying multi-modal uncertainty.

Various techniques have been proposed to account for large errors and incorporate measurements from multiple sensors, such as optimization-based methods and robust Bayesian filters. Optimization-based methods, such as factor graphs with robust cost functions [7–10], are capable of handling nonlinearities and uncertainties by optimizing over a large state space consisting of multiple states and uncertainty parameters. However, these approaches are computationally demanding due to the size of the optimization space, and require significant parameter tuning to achieve good estimation performance [11]. Moreover, state uncertainty is often not assessed in optimization-based approaches due to the increased computational demands, which limits their utility in scenarios where reliable localization is crucial.

On the other hand, Bayesian filters—such as Extended Kalman filters (EKFs) and Unscented Kalman filters (UKFs)—estimate the probability distribution of the state based on the history of measurements [12]. These filters utilize Bayesian statistics to track both the state estimate and its associated uncertainty by combining prior state information, vehicle dynamics, and sensor measurements across time. For example, EKFs have been used to provide localization estimates by integrating stereo camera measurements and Real-Time Kinematic GPS (RTK-GPS) [13]. Another work [14] proposed a similar method by fusing GPS, INS, a monovision camera, and a 3D cartographical model. An adaptive EKF framework in [15] used Google Street View images with GPS for state estimation. Outlier-robust versions of these filters use strategies such as robust cost functions, statistical tests, minimax optimization, or heavy-tailed priors to handle outliers and model complex noise distributions. For example, robust UKFs were explored in [16] for tight integration of multiple GPS receivers with a monocular camera and an IMU, and in [17] for vehicle geo-localization with a GPS receiver, a video camera, and a 3D city model. While these filters are computationally efficient and easy to implement, they require careful parameter tuning based on the application to achieve good state and uncertainty estimation performance. This can become challenging in complex systems that involve multiple sensor measurements and their associated parameters [18]. Moreover, these filters rely on a Gaussian distribution to model the state probability distribution, which restricts their capacity to adequately account for the multi-modal uncertainty that can arise due to sensor measurements in complex urban environments.

Particle filter, another type of Bayesian filter, captures multi-modal uncertainty over the state by tracking the distribution as a weighted collection of points in the state space [19]. In a recent work [20], a particle filtering framework was proposed for fusing GNSS with camera images and for characterizing the uncertainty in localization from sensor fusion. Similarly, in [21], images from a monocular camera were fused with low-cost GPS sensors and a map to provide high-accuracy localization. Variants

of particle filters have been explored for multi-modal sensor fusion such as decentralized filters [22] and differentiable filters [23]. However, particle filters are often challenged by the "curse of dimensionality"—or exponentially increasing computational complexity with higher-dimensional state spaces—leading to prohibitive computational costs [24].

Recent literature has proposed two other promising approaches to efficiently capture multi-modal uncertainty over the state: employing a combination of filters, also known as Interacting Multiple Model(IMM) filters [25], and modeling the distribution as a mixture of Gaussian distributions [26]. For example, IMM filters have been explored for providing positioning by integrating low-cost GPS and in-vehicle sensors while adapting the vehicle model to various driving conditions [27]. An IMM filter was proposed in [28] to provide fault-tolerant positioning by integrating IMU with wheel encoders in GPS-degraded environments for mobile robots. Self-adaptive Gaussian mixture models were used in [29] to model the effects of non-Gaussian GNSS outliers. These approaches have shown an improved ability in capturing real-world uncertainty in a variety of robotics and navigation applications. However, these techniques also have limitations, such as model complexity and difficulty in parameter selection.

In addition to conventional approaches, various studies have introduced data-driven adaptations of filtering algorithms by incorporating machine learning and deep learning techniques. To enhance localization accuracy during GNSS outages, an adaptive Kalman filter that incorporates neural network-derived position and velocity measurements was proposed in [30]. A fully differentiable pipeline to train an Extended Kalman Filter (EKF) for visual-inertial odometry was established in [31]. Similarly, other researchers have explored neural network-based methodologies for modeling diverse parameters within filtering techniques [23, 32–34]. These methods use existing data to automatically infer good parameters for improving the accuracy of localization. However, the lack of transparency within these methods poses a challenge, as relying on parameters fine-tuned for accuracy on a limited dataset may not adequately account for uncertainty in the general case.

The limitations of existing robust state estimation methods in effectively capturing uncertainty highlight the need for developing techniques to address these challenges. In this work, we propose a novel robust Bayesian filtering framework that uses Rao–Blackwellization to effectively capture multi-modal uncertainty while maintaining the accuracy and error resilience of existing robust Bayesian filtering approaches. Our approach enhances the robust EKF by tracking multiple points where the EKF and its robust cost function are linearized. By employing a particle filter to effectively track these linearization points, our framework can overcome the limitations of the linearization and Gaussian assumptions in the EKF, capturing the multi-modal uncertainty in a computationally efficient manner.

We use Rao–Blackwellization [35] to integrate the robust EKF and the particle filter into a single robust Bayesian filter that estimates a multi-modal probability distribution of the state. This facilitates the reuse of the standard filtering components in the EKF for efficient implementation [36]. We then apply this filter to a multi-sensor setup comprising of camera, GNSS, and attitude and heading reference system (AHRS), where the sensor modules are developed based on our previous work [37]. However, tuning the

parameters for our complex multi-sensor setup is challenging. To address this, we devise a gradient descent-based optimization strategy for efficient parameter tuning.

Next, we use the estimated multi-modal uncertainty to develop an approach for computing position error bounds along the lateral, longitudinal, and vertical directions of motion. These bounds enable us to assess the positioning performance for safe operation in urban environments [38]. Our approach is implemented in PyTorch [39] to leverage parallelization for fast execution and automatic differentiation capabilities.

Our strategy for using Rao–Blackwellization to improve the estimation of state and uncertainty shares similarities with [40, 41] and distinguishes itself from the conventional Rao-Blackwellized particle filter in several key aspects. In traditional applications of Rao–Blackwellization, the overall state space of interest is partitioned into multiple substates, where each substate is tracked with a different filter [26, 42, 43]. Our approach, on the other hand, introduces an additional term—the linearization point—and employs Rao–Blackwellization to factor it separately from the primary state of interest. Moreover, our approach emphasizes multi-sensor and robust filtering settings and focuses on characterizing multi-modal uncertainty to compute position error bounds, whereas the previous approaches do not address these aspects.

The contributions of this work are summarized as follows:

- We develop a novel Bayesian filtering framework that captures multi-modal uncertainty while accommodating diverse sensor measurements and possible outliers. Our approach tracks a probability distribution of the points for linearizing the dynamics, observation, and robust cost models, effectively accounting for errors from approximation and diverse noise sources.
- Drawing on recent advancements in differentiable filter design research [23, 44], we present a gradient descent-based optimization strategy for tuning the parameters in our filter. The optimization objective is expressed as a total loss function that includes both measurement and position loss terms.
- We present a method for estimating position error bounds along the lateral, longitudinal, and vertical directions of the vehicle's motion. Our proposed approach uses the multi-modal probability distribution obtained from our hybrid filter and explicitly accounts for uncertainties in both position and heading estimation.
- We validate our approach using real-world data from Hong Kong [45], a dense urban environment, and demonstrate its effectiveness in achieving improved reliability in the position error bounds, while maintaining competitive state estimation performance compared to existing methods. Moreover, we demonstrate that the computational requirements of our approach are comparable to existing methods, making it a practical choice for state estimation in urban environments.

The rest of the paper is structured as follows: in Sect. 2, we present background on robust Bayesian filters. Next, we derive our proposed robust filter for multi-modal uncertainty quantification in Sect. 3. Section 4 outlines the dynamics and the sensor modules in our multi-sensor setup. We describe the estimation of position error bounds from the captured multi-modal uncertainty in Sect. 5. The loss functions and optimization strategy for tuning the filter parameters are presented in Sect. 6.

Experimental results and discussions for the performance evaluation of our approach are provided in Sect. 7. Finally, we conclude the paper with Sect. 8.

2 Background on robust Bayesian filters

We consider a nonlinear discrete-time system with a time-invariant transition function f and a time-varying measurement function h_t given as follows:

$$x_t = f(x_{t-1}) + w_t, \qquad w_t \sim \mathcal{N}(0, Q_t), \tag{1}$$

$$m_t = h_t(x_t) + v_t, \qquad v_t \sim \mathcal{N}(0, R_t), \tag{2}$$

where x_t is the system state and m_t is the measurement at time t. The process noise w_t and the measurement noise v_t are zero mean with covariance matrix Q_t and R_t, respectively. The time-varying nature of h_t accommodates an asynchronous measurement setting, where measurements from different sensors are obtained at distinct time instances due to varying sampling rates.

The objective of Bayesian filtering is to estimate the probability distribution of the state vector x_t given the sequence of measurement vectors $m_{1:t} = \{m_1, \ldots, m_t\}$ and the prior probability estimate $p(x_{t-1}|m_{t-1})$. Using Bayes' rule and the law of total probability, the posterior probability of the state conditioned on the measurements is given as

$$p(x_t|m_{1:t}) \propto p(m_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|m_{1:t-1}) \mathrm{d}x_{t-1}, \tag{3}$$

where $p(m_t|x_t)$ is the likelihood function representing the probability of observing measurement m_t from state x_t. The term $p(x_t|x_{t-1})$ represents the transition probability from state x_{t-1} to x_t.

Conventional approaches to Bayesian filtering approximate the probability distribution $p(x_t|m_t)$ as a Gaussian distribution for efficiency and tractability. For example, the extended Kalman filter (EKF) uses a linearized version of the transition and measurement functions, and the unscented Kalman filter (UKF) uses nonlinear function evaluations at preselected sigma points to model $p(x_t|m_t)$ [46]. Using these approximations, $p(x_t|m_t)$ can be efficiently estimated from $p(x_{t-1}|m_{t-1})$ and the measurements through Eq. 3 and matrix operations. This process is commonly broken down into two steps, namely the predict and the update step [47]. However, the estimation accuracy of these approaches can substantially degrade when the process and measurement noise significantly deviates from the modeled Gaussian distribution, which is a common occurrence when using sensors in real-world settings [48–50].

To improve the accuracy in non-Gaussian settings, we can replace the Gaussian likelihood used in conventional Bayesian filtering approaches with a robust cost function-based likelihood

$$p(m_t|x_t) \propto \exp\left(-\rho(r_t)\right), \tag{4}$$

where $r_t = m_t - h_t(x_t)$ is the residual, and $\rho(r_t)$ is the robust cost function, such as Huber loss or Tukey biweight loss [51]. The robust cost function reweights the influence

of measurements based on their residuals, thereby reducing the impact of outliers and unmodeled errors.

The robust cost-based likelihood can be integrated into the EKF and UKF frameworks by modifying the update step [52–55]. A common method to do this involves reweighting the measurement covariance matrices R_t based on the first-order approximation of the cost function $\psi(r_t)$, also known as the influence function,

$$\psi(r_t) = \left.\frac{\mathrm{d}\rho(r)}{\mathrm{d}r}\right|_{r=r_t}. \tag{5}$$

The measurement covariance matrix R_t is then reweighted using $\psi(r_t)$ to obtain a robust covariance matrix $\tilde{R}_t$:

$$\tilde{R}_t = \psi(r_t)^{-1} R_t. \tag{6}$$

The robust measurement covariance matrix $\tilde{R}_t$ is then used in the update step of the EKF and UKF to compute the Kalman gain and update the estimate of the state and covariance.

The performance of robust Bayesian filtering depends on the careful selection of a robust cost function and its associated parameters, which can be challenging to identify as they depend on the sensor configuration and environmental characteristics [56]. Furthermore, the approximations employed by these filters to enhance computational efficiency—such as linearizing in transition, measurements, and robust cost—can also introduce inaccuracies in both the estimated state and the quantification of uncertainty. Therefore, it is important to account for the impacts of these error sources to effectively represent the uncertainty within the filtering framework.

3 Proposed robust Bayesian filter with multi-modal uncertainty

In the context of multi-sensor navigation in urban environments, the presence of diverse and dynamic noise can cause significant deviation from the typically assumed Gaussian behavior in the sensor measurements. Instead, the measurements may exhibit multi-modal behavior, which then propagates to a multi-modal uncertainty in the state estimation process [29, 49]. While the robust Bayesian filters discussed in the previous section can improve state estimation accuracy by relying on a single Gaussian distribution to represent the state probability distribution, they are ill-equipped to effectively capture the underlying multi-modal uncertainty.

To address this challenge, we propose a robust Bayesian filter that incorporates multi-modal uncertainty modeling into the estimation process. Our approach extends the standard Bayesian filtering framework by explicitly incorporating a linearization point, denoted as x_t^l, at each timestep t. The linearization point serves as a reference for linearizing the transition and measurement models, as well as for computing the influence function. The overall approach is illustrated in Fig. 1

We first rewrite Eq. 3 to estimate the posterior probability of the state x_t given the linearization point x_t^l and the measurements $m_{1:t}$

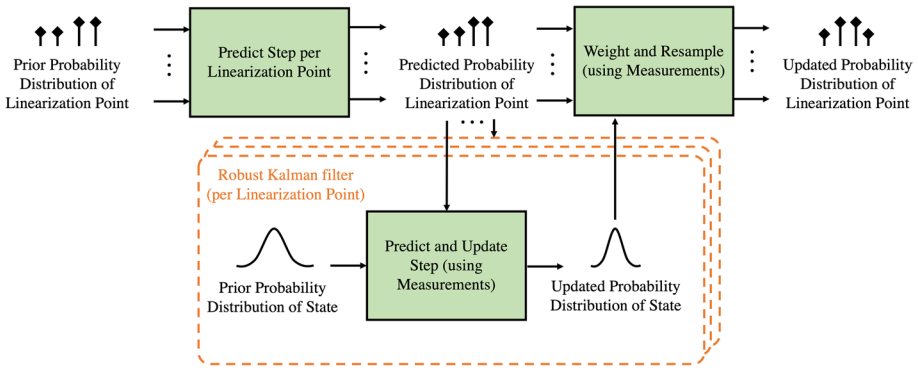


Fig. 1 Architecture of the proposed robust Bayesian filter. The filter takes a prior probability distribution of the linearization points and state and uses a system transition model and measurement model to estimate the posterior probability distribution of the state. The linearization point probability distribution is represented as weighted samples, and the state probability distribution is represented by a Gaussian distribution tracked by the underlying robust Kalman filter. The filter first applies the prediction step to each linearization point in the prior distribution. Using this predicted point for linearizing the robust filter system transition, measurement and cost models, we then apply the predict and update steps in parallel to update the state probability distribution for each linearization point. Based on the updated state probability distribution, the linearization point samples are refined and their weights are updated using the available measurements. Finally, the weighted linearization point samples are resampled and combined with the updated state probability distribution to represent the posterior state probability distribution

$$p(x_t|x_t^l, m_{1:t}) \propto p(m_t|x_t^l, x_t) \int p(x_t|x_t^l, x_{t-1}) p(x_{t-1}|x_t^l, m_{1:t-1}) \mathrm{d}x_{t-1}. \tag{7}$$

In this equation, the probability terms are computed in a manner analogous to the robust EKF approach described earlier in Sec. 2, with the key distinction that the linearization point x_t^l is used for linearizing the equations.

In order to incorporate the uncertainty in the selection of an appropriate linearization point x_t^l, we model a separate probability distribution. The corresponding filtering equation is expressed as

$$p(x_t^l|m_{1:t}) \propto p(m_t|x_t^l) \int p(x_t^l|x_{t-1}^l) p(x_{t-1}^l|m_{1:t-1}) \mathrm{d}x_{t-1}^l. \tag{8}$$

This equation has a similar form to Eq. 3 and is used to update the linearization point x_t^l based on available measurements m_t. To effectively model the multi-modal uncertainty, we employ a particle filter to estimate $p(x_t^l|m_{1:t})$. The filter uses a set of weighted particles to represent the probability distribution, where each particle is a linearization point. First, we use the prior distribution $p(x_{t-1}^l|m_{1:t-1})$ to generate a set of prediction particles given by the state transition model

$$x_t^l = f(x_{t-1}^l) + w_t^l, \qquad w_t^l \sim \mathcal{N}(0, Q_t^l), \tag{9}$$

where w_t^l is the process noise for the linearization point at time t with covariance matrix Q_t^l. For simplicity, we set the covariance $Q_t^l = Q_t$ in this paper. However, we note that the choice of Q_t can have a significant impact on the estimation accuracy and robustness. Future work can consider exploring the different choices of Q_t^l as a separate term—such as by taking advantage of the well-established methods in particle filter literature and the use of proposal distributions [57, 58]—to improve the estimation performance.

Based on the linearization point x_t^l corresponding to each particle, we can estimate the posterior state probability distribution $p(x_t|x_t^l, m_{1:t})$ using the Bayesian filtering expression described in Eq. 8, while keeping the linearization point x_t^l unchanged. However, due to the randomness in the particle filter, tracking the linearization point in this manner can result in precision errors, particularly when the number of particles is low. To address this issue and improve the accuracy of the estimation process, we propose a two-step approach. First, we estimate $p(x_t|x_t^l, m_{1:t})$ using the linearization point x_t^l. Then, we refine the value of x_t^l by setting it to the mean of the posterior distribution $p(x_t|x_t^l, m_{1:t})$. This refinement step ensures that the linearization point is more closely aligned with the state probability distribution, improving the accuracy of the estimation.

In the second step of the particle filter, we update the weight of each prediction particle based on the available measurements m_t and likelihood $p(m_t|x_t^l)$. We model the likelihood as a normal distribution with mean based on the nonlinear measurement model $h_t(\cdot)$ and covariance R_t^l

$$p(m_t|x_t^l) = \mathcal{N}(m_t|h_t(x_t^l), R_t^l) \tag{10}$$

To improve estimation performance, appropriate values of R_t^l can be assigned based on the application. By modeling R_t^l separately from R_t, we can incentivize tracking distinct linearization points in high uncertainty scenarios without compromising on the estimation accuracy.

Finally, we perform a resampling step on the set of linearization points based on the updated weights to improve our estimate of the posterior distribution $p(x_t^l|m_{1:t})$. We use the soft resampling strategy proposed in [59] to maintain differentiability in the parameter optimization.

Algorithm 1: Proposed robust Bayesian filter

Input: Initial set of linearization points $x_{0,i}^l$, $i = 1, \ldots, N$, state estimates $x_{0,i}$, $k = 1, \ldots, N$; system transition model $f(\cdot)$; measurement model $h(\cdot)$; measurement data $m_{1:T}$; covariance matrices Q_t, Q_t^l, R_t, R_t^l

Output: Estimated posterior probability distribution $p(x_t|m_{1:T})$

Initialize particle weights $w_{0,i} = 1/N$, $i = 1, \ldots, N$; state covariance matrices $\Sigma_{0,i}$, $i = 1, \ldots, N$

for $t = 1$ **to** T **do**

  for $i = 1$ **to** N **do**

    Generate prediction particles $x_{t,i}^l$ based on Eq. 9;

    Update state $x_{t,i}$ and covariance $\Sigma_{t,i}$ using robust EKF and linearization point $x_{t,i}^l$

    Refine the linearization point $x_{t,i}^l \leftarrow x_{t,i}$;

    Compute weights $w_{t,i} \propto w_{t-1,i} \cdot p(m_t|x_{t,i}^l)$;

  end

  Resample particles to obtain a new set of particles $x_{t,i}^l$ and weights $w_{t,i}$;

  Normalize the weights $w_{t,i}$;

  Estimate the state probability distribution $p(x_t|m_{1:t}) \approx \sum_{i=1}^{N} w_{t,i} \cdot p(x_t|x_{t,i}^l, m_{1:t})$ using Eq. 11;

end

To estimate the overall probability distribution of the state $p(x_t|m_t)$, we combine the estimates from the particle filter and the robust EKF using the law of total probability and the Rao-Blackwell theorem

$$p(x_t|m_{1:t}) = \int p(x_t^l|m_{1:t})p(x_t|x_t^l, m_{1:t})\mathrm{d}x_t^l, \tag{11}$$

which allows us to estimate $p(x_t|m_{1:t})$ by conditioning on the linearization point x_t^l. This approach has a smaller estimation variance than Eq. 3 based on the Rao-Blackwell theorem, allowing us to better capture the uncertainty in the estimate of the state x_t. The overall algorithm is outlined in Algorithm 1.

The use of a particle filter allows us to capture multiple modes in the distribution of the linearization point, which in turn enables us to better represent the multi-modal uncertainty in state estimation. However, particle filters are often challenged by the "curse of dimensionality" when the state space is large. To address this challenge, we restrict the domain of the linearization point to the position domain, reducing the dimensionality of the particle filter state space. This enables us to maintain computational efficiency while modeling the uncertainty inherent in the system.

4 Multi-sensor state estimation

In this section, we describe how we apply the approach described in the previous sections for multi-sensor state and uncertainty estimation. Figure 2 illustrates the multi-sensor setup and the individual sensor modules. The camera module captures an image at each timestep, and utilizes robust features and depth estimated using deep neural networks to assess the motion. The GNSS module combines pseudorange measurements from multiple satellite constellations and a base station. The attitude and heading reference system (AHRS) provides the orientation and angular velocity measurements. Our proposed filter integrates all of these measurements to estimate the state and uncertainty. Further details regarding the filter and each of the sensor modules are provided in the subsequent sections.

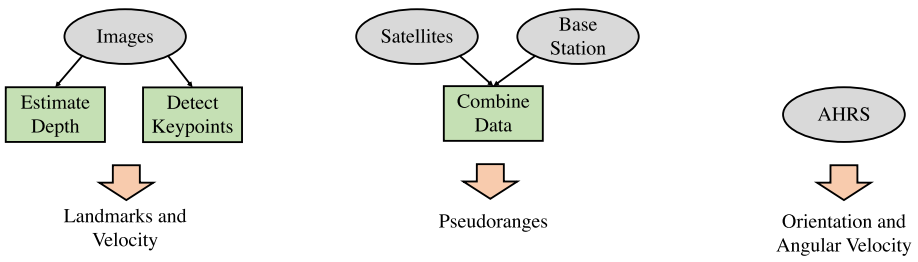


Fig. 2 Sensor modules in our multi-sensor setup. The setup includes **a** camera module with depth estimation and feature detection component, **b** GNSS module that combines pseudorange measurements from multiple satellite constellations and a nearby base station, and **c** AHRS module that provides orientation and angular velocity measurements. The camera module outputs velocity and landmark measurements, which are fused with the GNSS and AHRS measurements by our filter for state estimation

4.1 State space and dynamics model

The state vector $x_t = \begin{bmatrix} p_t & v_t & q_t & \tilde{\omega}_t \end{bmatrix}$ consists of position p_t, velocity v_t, orientation quaternion q_t and the bias in angular velocity measurements $\tilde{\omega}_t$ at time t. We model the vehicle motion as a constant velocity model using the angular velocity ω_t obtained from the AHRS:

$$f(x_{t-1}) = \begin{bmatrix} I_{3\times3} & \Delta t I_{3\times3} & 0_{4\times4} & 0_{3\times3} \\ 0_{3\times3} & I_{3\times3} & 0_{4\times4} & 0_{3\times3} \\ 0_{4\times4} & 0_{4\times4} & I_{4\times4} & 0_{3\times3} \\ 0_{4\times4} & 0_{4\times4} & 0_{4\times4} & I_{3\times3} \end{bmatrix} x_{t-1} + \begin{bmatrix} 0_{3\times3} \\ 0_{3\times3} \\ \Delta t \dot{q}(\omega_t - \tilde{\omega}_t) \\ 0_{3\times3} \end{bmatrix}, \tag{12}$$

where Δt is the time step between consecutive measurements, and $\dot{q}(\omega)$ is the quaternion derivative corresponding to the angular velocity ω. To avoid numerical stability issues, we re-normalize the quaternion q_t to a unit quaternion each time we update it based on the dynamics. The process noise covariance matrix Q_t is given as

$$Q_t = \begin{bmatrix} Q_t^p & 0_{3\times3} & 0_{4\times4} & 0_{3\times3} \\ 0_{3\times3} & Q_t^v & 0_{4\times4} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & Q_t^q & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{4\times4} & Q_t^{\tilde{\omega}} \end{bmatrix}, \tag{13}$$

where Q_t^p, Q_t^v, and $Q_t^{\tilde{\omega}}$ denote the diagonal process noise covariance matrices for position, velocity, and angular velocity bias, respectively. We construct the process noise covariance for the quaternion Q_t^q by linearizing the Euler-to-quaternion transformation about q_t, which enables us to incorporate correlation across terms and obtain better uncertainty characterization in the dynamics. To compute Q_t^q, we first obtain the Jacobian J_t^q of the Euler-to-quaternion transformation. We then use the diagonal 3×3 process noise covariance matrix $\tilde{Q}_t^q$ in the Euler space, and apply it to J_t^q as

$$Q_t^q = J_t^q \tilde{Q}_t^q (J_t^q)^\top. \tag{14}$$

To account for the varied acquisition rates of sensors, we utilize sequential or asynchronous updates within our filter [60, 61]. Specifically, the state probability distribution is updated incrementally as the measurements arrive, with different update steps executed for the measurements. This approach both improves computational efficiency and enables modularity, allowing easy addition or removal of the different sensor components.

In the following sections, we provide a detailed description of each sensor module and the corresponding filter observation model.

4.2 Camera module

Our visual odometry pipeline utilizes consecutive image frames $\{I_{t-1}, I_t\}$ from an onboard monocular camera to estimate the vehicle motion. The choice of a monocular camera is motivated by its advantages in terms of affordability, compactness, and versatility over other visual sensors [62]. While traditional techniques based on feature-matching from monocular images can estimate motion up to an unknown scale factor [63, 64], we leverage recent research on visual odometry to develop a more accurate approach. Our pipeline incorporates neural networks to both detect features and estimate their associated depth from camera images, which allows us to obtain

scale-consistent and robust visual odometry. This strategy, which combines monocular depth estimates with keypoints detected on the image, has been previously explored in [65, 66] and has been shown to outperform both geometry-based and learning-based methods. Throughout the paper, we refer to this pipeline as Monocular-Depth aided Visual Odometry (MD-VO).

MD-VO relies on keypoint detection and depth estimation to construct the observation model.

Keypoint detection In the first step, we detect a set of keypoints $\mathbf{k}_t = \{k_t^1, \ldots, k_t^M\}$ and descriptors $\mathbf{d}_t = \{d_t^1, \ldots, d_t^M\}$ from the input image I_t where M is the total number of detected features. Keypoints are the distinctive local regions in the image, while descriptors are vectors that represent these regions with invariance to motion and lighting deformations. However, real-world camera images are often subject to significant variations over time, leading to spurious feature matching and erroneous motion estimation [2]. To mitigate the effects of spurious feature-matching, we use the neural network-based SuperPoint detector [67], which has been optimized using real-world images. This detector is designed to identify keypoints and descriptors that are robust to real-world deformations and suitable for multiple-view geometry problems. By utilizing these features, we improve our feature extraction step and improve positioning performance in urban environments.

Next, we match the descriptor set $\mathbf{d}_t$ with the descriptor sets $\{\mathbf{d}_{t-1}, \ldots, \mathbf{d}_{t-T_{SP}}\}$ from the previous $T_{SP} = 10$ frames. We rank the matches based on the sum of closest matching scores η_{SP}^i calculated as:

$$\eta_{SP}^i(t') = \min_j \|d_t^i - d_{t'}^j\| \tag{15}$$

$$\eta_{SP}^i = \sum_{t'=t-T_{SP}}^{t-1} \eta_{SP}^i(t'). \tag{16}$$

We apply a threshold τ_{SP} to the scores η_{SP}^i to select the 50 best-performing keypoints for subsequent motion estimation. This strategy ensures that the selected features persist in multiple frames, leading to more robust feature detection. To further enhance the robustness of subsequent motion estimation against outliers, we employ Random Sample Consensus (RANSAC) algorithm [68] to identify and remove large outliers.

For the remainder of this section, we will use the notation $\mathbf{k}_t$ to refer to the M' best keypoints identified using the aforementioned method. Additionally, we will use $\mathbf{k}_{t-1}$ to denote the matching keypoints (i.e., the minimizers of $\eta_{SP}^i(t-1)$) from the previous frame.

Depth Estimation In the depth estimation step, we estimate the 3D depth $I_{t-1}^D(k)$ associated with every pixel k in frame I_{t-1} from the previous time $t-1$. Projecting 2D image coordinates to 3D is an ill-posed and inherently ambiguous task. However, previous research has shown that this task can be effectively accomplished by leveraging patterns in the appearance of objects [69, 70].

In this work, we utilize pre-trained models from MonoVO [70] to estimate the depth. To reduce computation requirements, we initiate depth calculation from the previous image acquisition timestep, as we only require the 3D locations at the previous time.

Next, we estimate the 3D coordinates $\tilde{\mathbf{k}}_{t-1}$ that correspond to the 2D keypoints $\mathbf{k}_{t-1}$ as

$$\tilde{\mathbf{k}}_{t-1} = \left[\tilde{k}^1_{t-1}, \ldots, \tilde{k}^{M'}_{t-1}\right]^\top \tag{17}$$

$$\tilde{k}^i_{t-1} = K^{-1}_{\text{cam}} \begin{bmatrix} I^D_{t-1}(k^i_{t-1}) k^i_{t-1} \\ I^D_{t-1}(k^i_{t-1}) \end{bmatrix}, \tag{18}$$

where K_{cam} is the matrix containing camera intrinsic parameters. Using these estimated 3D coordinates, we construct observation models based on the current frame.

To construct the observation model for the camera module, we consider the sensor measurement m_t based on the current frame 2D keypoints $\mathbf{k}_t$ and an estimated body frame velocity $\tilde{v}_t$. The measurement vector m_t is given as

$$m_t = \left[[1\ 0]\mathbf{k}_t^\top \ [0\ 1]\mathbf{k}_t^\top \ \tilde{v}_t^\top\right]. \tag{19}$$

here $\mathbf{k}_t$ is rearranged in a column-major fashion to obtain a single vector, which makes it easier to use within the filter. We obtain the velocity $\tilde{v}_t$ from the matched 3D-2D keypoints using perspective-n-point algorithms [71]. We enforce $\tilde{v}_t$ to be nonzero only along the axis that aligns with the forward direction of the vehicle's motion. Our empirical evaluations indicate that adding this velocity measurement to the overall measurement vector helps regularize the filter's estimate of velocity and improves the overall stability of the filter. Based on the estimated 3D coordinates $\tilde{\mathbf{k}}_{t-1}$ of the previous frame keypoints $\mathbf{k}_{t-1}$, we model the measurement m_t as a function of both the vehicle's state x_t and an estimate of the keyframe's position $\tilde{p}_{t-1}$ and orientation $\tilde{q}_{t-1}$

$$h_t(x_t) = \left[[1\ 0]h_1(p_t, q_t)^\top \ [0\ 1]h_1(p_t, q_t)^\top \ h_2(v_t, q_t)^\top\right]^\top, \tag{20}$$

$$h_1(p_t, q_t) = \widetilde{k_{t-1}} \begin{bmatrix} R(\tilde{q}_{t-1})R(q_t)^\top \\ (p_t - \tilde{p}_{t-1})^\top \zeta R(q_t)^\top \end{bmatrix} K^{\text{T}}_{\text{cam}}, \tag{21}$$

$$h_2(v_t, q_t) = R(q_t)v_t \tag{22}$$

where $R(q_t)$ and $R(\tilde{q}_{t-1})$ are rotation matrices corresponding to the orientations q_t and $\tilde{q}_{t-1}$, respectively, and ζ is the scaling parameter. The measurement noise matrix R_t corresponding to m_t is set as a diagonal matrix with identical values for each keypoint. In addition, we assign small values to the diagonal terms corresponding to the zero-valued entries in $\tilde{v}_t$. This is because the vehicle can be assumed to have negligible or zero motion along non-forward directions with a high level of certainty. By assigning smaller values to these terms, the filter is incentivized to model the vehicle's motion primarily along the forward direction.

We estimate the keyframe's position $\tilde{p}_{t-1}$ and orientation $\tilde{q}_{t-1}$ separately from the vehicle's state x_t by running a concurrent EKF with identical dynamics and sensor modules, except for a modification to the camera module. In the camera module, we set the

observation model $h_t(x_t)$ to the expected body frame velocity obtained from Eq. 22, and measurement m_t to the velocity $\tilde{v}_t$ derived through 3D-2D matching. Based on our experiments, using a separate filter to estimate the keyframe location provides more stable estimates compared to using estimates based on the same filter.

4.3 GNSS module

Our GNSS pipeline utilizes pseudorange measurements $\boldsymbol{\rho}_t = \{\rho_t^1, \ldots, \rho_t^{M_t}\}$ from multiple constellations (GPS and Beidou) and a nearby base station, where M_t is the total number of visible GNSS satellites at time t. These measurements are obtained by calculating the time delay between the transmission of a signal from a satellite and its reception at the receiver onboard the vehicle. The kth pseudorange measurement ρ_t^k is related to the distance between the receiver and the satellite by the following equation [72]:

$$\rho_t^k = \left\| p_t - p_t^k \right\| + b_{\text{const}} - b^k + \epsilon_{\text{ion}} + \epsilon_{\text{trop}} + \epsilon_{\text{mp}} + \epsilon, \tag{23}$$

where p_t^k denotes the position of the satellite corresponding to the kth measurement at time t, b_{const} denotes the bias error in the receiver's clock that also depends on the GNSS constellation, b^k denotes the bias error in the satellite's clock, $\epsilon_{\text{ion}}, \epsilon_{\text{trop}}$ denote the errors due to ionospheric and tropospheric effects, ϵ_{mp} denotes the error due to multipath effects and ϵ denotes the random error. We use multiple GNSS constellations to compensate for limited satellite visibility in urban environments.

To utilize the pseudorange measurements $\boldsymbol{\rho}_t$ for positioning, it is necessary to account for the various error sources specified in Eq. 23. A common approach is to employ double-differenced (DD) measurements [73]. To construct DD measurements, measurements from the base station $\tilde{\boldsymbol{\rho}}_t$ are first subtracted from the received measurements $\boldsymbol{\rho}_t$, thereby canceling out the effects of ionospheric and tropospheric disturbances, as well as satellite clock bias. To account for the receiver clock bias, a reference satellite (index denoted as ref) is selected and its measurements are subtracted from all other measurements. For the kth satellite, the DD measurement $\nabla\Delta\rho_t^k$ is expressed as follows:

$$\nabla\Delta\rho_t^k = \Delta\rho_t^k - \Delta\rho_t^{\text{ref}}, \tag{24}$$

$$\Delta\rho_t^k = \rho_t^k - \tilde{\rho}_t^k. \tag{25}$$

The above approach successfully removes most error sources based on clock differences between the satellite and the receiver. However, DD measurements that form differences between measurements from different constellations still have a remaining receiver clock bias error due to lack of synchronization between the constellations [72]. This error—known as the Inter-System Bias (ISB)—does not change significantly over long periods of time. Therefore, we pre-estimate this error from initial measurement residuals and ground truth data.

The overall measurement vector m_t for the GNSS module is constructed using DD measurements as

$$m_t = \left[\nabla\Delta\rho_t^1 \ \ldots \ \nabla\Delta\rho_t^{M_t-1}\right]^\top. \tag{26}$$

The corresponding sensor observation model $h_t(x_t)$ is as follows:

$$h_t(x_t) = \left[\nabla\Delta\|p_t - p_t^1\| \ \ldots \ \nabla\Delta\|p_t - p_t^{M_t-1}\|\right]^\top, \tag{27}$$

where the double differences are constructed in a similar way to the DD measurements in Eq. 25.

We set the measurement noise matrix R_t corresponding to m_t as a diagonal matrix with identical entries for all measurements for simplicity, which is common practice in filtering literature. However, it is worth noting that alternative methods of assigning covariance exist that take into account additional properties such as signal strength and satellite elevation [74]. Investigating the impact of such techniques on filter performance could be a direction for future work.

4.4 AHRS module

In the AHRS module, we use the orientation measurements o_t to update the orientation quaternion q_t tracked by the filter. The sensor observation model $h_t(x_t)$ is the estimated quaternion q_t. The measurement m_t is set as the quaternion $q(o_t)$ corresponding to the measured orientation o_t, where $q(\cdot)$ is the Euler-to-quaternion transformation function. The measurement noise covariance matrix R_t is calculated using the Jacobian J_t^q of the Euler-to-quaternion transformation and a 3×3 diagonal covariance matrix $\tilde{R}_t$ for Euler angles, similar to Eq. 14

$$R_t = J_t^q \tilde{R}_t (J_t^q)^\top. \tag{28}$$

To ensure numerical stability, we set the off-diagonal terms in R_t to zero and only keep the diagonal terms.

5 Estimating position error bounds

To ensure the safety of our proposed filtering method for location estimation, we use the tracked probability distribution $p(x_t|m_{1:t})$ to estimate probabilistic bounds on the estimation error. These error bounds can then be compared against carefully designed safety limits, such as the ones described in [38], to detect situations when the filter's location output is unsafe for navigation. Incorporating these checks adds a layer of protection to the localization system against uncorrectable estimation errors, improving the system's reliability.

To estimate the position error bounds, we first utilize the vehicle's estimated orientation q_t to determine the transformation from the global coordinate frame to the coordinate frame aligned with the lateral (side-to-side), longitudinal (forward–backward), and vertical directions of the vehicle's motion. This transformation allows us to project the probability distribution $p(x_t|m_{1:t})$ onto the motion axes of the vehicle, enabling us to estimate the error along directions that are more relevant from a safety perspective.

The position part of the state probability distribution is represented as a weighted sum of Gaussian distributions, where the ith component's mean and covariance are denoted by $p_{t,i}$ and $\Sigma_{t,i}^p$, respectively. To estimate the probability distribution corresponding to the axes-aligned error vector e_t, we project each Gaussian distribution component along the axes specified by q_t.

To compute the mean $\mu^e_{t,i}$ parameter of the ith projected Gaussian distribution component, we apply the rotation matrix $R(q_t)$ to compute the difference between $p_{t,i}$ and the filter position estimate p_t

$$\mu^e_{t,i} = R(q_t)(p_{t,i} - p_t). \tag{29}$$

To estimate the covariance parameter $\Sigma^e_{t,i}$, we also need to account for the uncertainty in estimating the orientation. We first generate M samples of orientation $\{q_1, \ldots, q_M\}$ as

$$q_j \sim \mathcal{N}\big(q_t, \Sigma^q_t\big), \qquad 1 \le j \le M \tag{30}$$

where Σ^q_t denotes the filter orientation covariance. We then project the covariance $\Sigma^p_{t,i}$ by applying the rotation matrix $R(q_j)$ and use Monte Carlo integration to estimate $\Sigma^e_{t,i}$:

$$\Sigma^e_{t,i} \approx \frac{1}{M} \sum_{i=1}^{M} R(q_j) \Sigma_{t,i} R(q_j)^\top. \tag{31}$$

To obtain a probabilistic error bound for a given direction d and confidence level α, we seek to compute the quantile τ^α_d from the error distribution $p(e_t)$ that satisfies the following equation:

$$p(e_t \mathbb{1}_d \le \tau^\alpha_d) \ge \alpha, \tag{32}$$

where $\mathbb{1}_d$ is a 3-dimensional unit vector with the entry corresponding to direction d set to one and the rest set to zero. However, as the error distribution is represented as a mixture of Gaussian distributions, we cannot compute this quantile analytically. Instead, we over-approximate the bound by maximizing across the component-wise quantiles. The position error bound τ^α_d for direction d is estimated as

$$\tau^\alpha_d = \max_{i \le N} \tau^{\alpha/2}_{d,i}, \tag{33}$$

where $\tau^{\alpha/2}_{d,i}$ denotes the quantile of the Gaussian distribution $\mathcal{N}(|\mu^e_{t,i}\mathbb{1}_d|, \mathbb{1}_i^\top \Sigma^e_{t,i}\mathbb{1}_d)$ for confidence level $\alpha/2$. The quantile $\tau^{\alpha/2}_{d,i}$ is computed using the absolute value of the mean to provide a one-sided bound on the position error, which is the maximum possible deviation in the positive or negative direction along the axis. Similarly, we use the confidence level of $\alpha/2$ for these bounds since it includes both sides of the probability distribution.

The position error bounds estimated from our approach are designed to account for the uncertainty resulting from the various sources of noise and linear approximations inherent in the filtering process. In instances where the noise sources are unbiased and the approximations hold true, the quantiles derived from the different components would exhibit similar behavior, resulting in a performance that is similar to an EKF. However, in cases where these assumptions do not hold, our methodology enables us to capture and quantify the uncertainty present across the different components, thereby offering an advantage over the Gaussian approximations utilized in the EKF.

6 Optimizing filter parameters

Designing effective filtering methods involves selecting appropriate values for all the filter parameters. For our filter, these parameters include standard deviation terms for the dynamics, sensor observation models and the scaling factor in the camera module. However, choosing suitable values for these parameters can be challenging and time-consuming due to the filter's complexity.

In settings with available prior ground truth data and measurements, we can address this challenge by leveraging ideas from recent research on differentiable filter design [23, 44]. We present a gradient-based optimization strategy that uses available measurements and ground truth position data to quickly find values for the filter parameters that result in good filter performance. Specifically, we minimize a total loss function that comprises measurement loss terms and position loss terms.

We employ a window-based strategy for optimization, wherein we randomly select windows of length $T = 5$ s from the available data. This approach allows us to incrementally tune the filter parameters on subsets of data, which is more computationally efficient than optimizing over the entire dataset. At the beginning of each window, we initialize the filter by sampling from a Gaussian distribution centered at the ground truth position with a standard deviation of 5 ms, which helps to regularize the optimization and to prevent overfitting to specific data instances.

The measurement loss $\mathcal{L}_m(\theta)$ with respect to the current parameters θ is the negative log probability of observing the measurements $m_{1:t}$ from the filter given the initial state.

$$\mathcal{L}_m(\theta) = -\sum_{t=2}^{T_m} \log p(m_t | m_{1:t-1}, \theta), \tag{34}$$

$$\approx -\sum_{t=2}^{T_m} \log p(m_t | x_t, \theta), \tag{35}$$

where T_m denotes the total measurement instances available within the window, and x_t is estimated using the filter.

The terms in this loss are available at the acquisition rate of the sensor measurements. This loss is similar to the unsupervised optimization objectives used for tuning Bayesian filter parameters in existing research [75].

The position loss $\mathcal{L}_p(\theta)$ is the mean squared error between the filter estimated positions $p_{1:T}$ and the corresponding ground truth positions $p^*_{1:T}$, which are usually available at a much slower rate than the sensor measurements. The position loss $\mathcal{L}_p(\theta)$ is expressed as

$$\mathcal{L}_p(\theta) = \sum_{t=2}^{T_p} \| p_t - p_t^* \|, \tag{36}$$

where T_p denotes the total ground truth instances available within the window. This loss provides a stronger signal for tuning the filter parameters where explicit supervision is available [76]. The total loss $\mathcal{L}(\theta)$ is expressed as the weighted sum of the position loss and each of the measurement losses

$$\mathcal{L}(\theta) = \mathcal{L}_p(\theta) + \lambda_{\text{cam}}\mathcal{L}_{\text{cam}}(\theta) + \lambda_{\text{gnss}}\mathcal{L}_{\text{gnss}}(\theta) + \lambda_{\text{ahrs}}\mathcal{L}_{\text{ahrs}}(\theta), \tag{37}$$

where $\mathcal{L}_{\text{cam}}, \mathcal{L}_{\text{gnss}}, \mathcal{L}_{\text{ahrs}}$ are calculated according to Eq. 35. The parameters $\lambda_{\text{cam}}, \lambda_{\text{gnss}}, \lambda_{\text{ahrs}}$ denote the weighting factors for the losses and are set to 0.2 in our case.

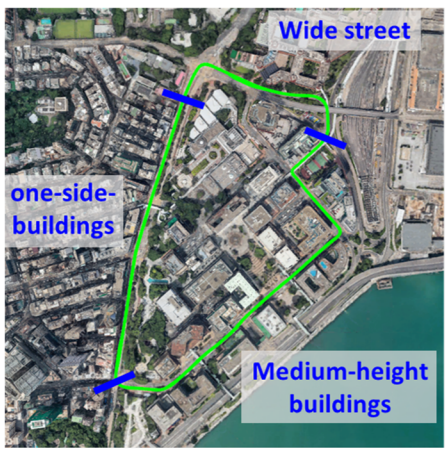
We optimize $\mathcal{L}(\theta)$ using gradient descent to update the parameters θ. To ensure numerical stability and prevent non-positive values during optimization, we use the softplus function—which is a smooth approximation of the rectified linear unit (ReLU) function—to enforce positivity for all parameters.

We note that the overall optimization problem consisting of several parameters is complex, and finding the global minimizer is not necessarily possible using this strategy since it depends on the initialization. Nevertheless, we find this strategy useful in refining the parameters starting from heuristically set initial values and reducing the overall tuning effort considerably.

7 Experimental results and discussion

7.1 UrbanNav dataset

We evaluate our proposed filter on the Hong Kong UrbanNav dataset [45] which includes measurements from diverse sensors like GNSS, LiDAR, camera, and IMU, gathered in an urban environment with distinct regions. These regions comprise a wide street, one-sided buildings, and medium-height buildings, creating non-line-of-sight and multipath errors in GNSS measurements. The environment also contains dynamic objects that contribute to errors in the camera module as shown in the dataset images in Fig. 3b. The dataset spans a total duration of 785 s, covering a path length of 3.64



(a) Vehicle trajectory from the UrbanNav dataset. The data was collected in Hong Kong and spans three distinct regions. These regions include a wide street, one-sided buildings, and medium-height buildings, each contributing to the sources of error in the available measurements.

(b) Randomly selected images from the UrbanNav dataset. The images demonstrate a diverse range of environmental conditions, including traffic and bridges, as well as dynamic objects that are present in the scene.

Fig. 3 Description of the UrbanNav Hong Kong dataset and sample images from the dataset collected in diverse environments

km with two loops of the same trajectory. The second loop was used to tune the filter's parameters, while the first loop was used to evaluate its performance.

In our evaluation, we utilize GNSS (GPS and Beidou), a monocular camera, and AHRS measurements from the UrbanNav dataset. The GNSS measurements were captured at 1 Hz using a commercial u-blox ZED-F9P receiver, with GPS and Beidou visibility ranging from 4 to 20 satellites at each time instant. AHRS data was obtained from an Xsens Mti 10, collected at 400 Hz. Images were captured at 15 Hz from a ZED2 camera with a resolution of 1920×1080. Ground truth was obtained from a post-processed RTK GNSS-INS integrated system, available at 1 Hz. In our experiments, the filter and the baselines were executed at 4 Hz. We focus on positioning performance and its uncertainty in this work.

7.2 Baselines

We compare the performance of our algorithm with respect to five filtering baselines, namely: a) Robust Naive RBPF (R-RBPF), b) Robust EKF with tight VO integration (R-EKF), c) Robust UKF with tight VO integration (R-UKF), d) Robust EKF with MD-VO only (R-EKF-VO), and e) Robust EKF with GNSS only (R-EKF-GNSS).

Some notes about these baselines are as follows:

- R-RBPF partitions the state space into position terms and tracks them using a standard particle filter with 20 particles to match our approach. The particle weights are computed using the robust version of the measurement likelihood function, as described in Eq 4. An EKF tracks the remaining terms, including orientation, velocity, and angular velocity bias. The position error bounds are determined by calculating the maximum deviation across the tracked particles along the lateral, longitudinal, and vertical directions.
- R-EKF and R-UKF use the robust cost function in their update step as described in Sect. 2. During the execution of R-UKF, it was observed that the covariance matrix becomes non-positive definite at certain time instances due to numerical errors associated with the UKF, a well-known challenge [77] attributed to the choice of filter parameters. To address this issue, we reinitialized the covariance matrix and continued executing the filter from the last estimated state.
- We use R-EKF-VO and R-EKF-GNSS as simple baselines to gain insights into the filter performance when only a single sensor is used.

We evaluate the filter's performance on the entire dataset. However, for ease of visualization, we present the results categorized into three distinct regions, as illustrated in Fig. 4. This approach enables a comprehensive evaluation of the filter performance within each typical category of regions present in urban environments.

7.3 Experimental setup

The filter and all baselines were tuned and executed on the Stanford Research and Computing Center's HPC cluster using an AMD 7502P CPU with 256 GB RAM and an NVIDIA Geforce RTX 2080Ti GPU. PyTorch is used for automatic differentiation and

tensor operations. The implementation of the filter and baselines is built upon the multi-modal sensor fusion codebase from [23].

To tune the filter parameters, we use R-EKF as a reference due to its fast execution time. We employ the same set of parameters for all filters to ensure a fair comparison. The parameters are tuned individually while the remaining parameters are held fixed. We optimize the parameters using the Adam optimizer [78] with a learning rate of 0.01 for 100,000 iterations. The final parameter values, along with the key manually set parameters, are listed in Table 1.

The Tukey biweight function [79] is used as the cost function in the filters. However, if the input residuals exceed a set threshold, this function becomes constant, causing the influence function to become non-invertible. To address this issue, we approximate the inverse by assigning a high value to the entries associated with residuals above the threshold. These threshold values are manually set and remain unchanged during the optimization of filter parameters using gradient descent.

7.4 Positioning performance

We first compare the positioning performance of our approach with respect to the baselines. Figure 4 shows the trajectory tracking plots of each approach and Fig. 5 shows the positioning performance with respect to time for qualitative analysis. The plots demonstrate that our proposed method produces position estimates with higher accuracy than the baselines. Specifically, in the region characterized by medium-height buildings, our approach exhibits clear improvement compared to R-EKF. Similarly, our approach outperforms R-UKF in a few regions where numerical errors occurred during covariance estimation, necessitating reinitialization. Notably, both R-EKF and R-UKF demonstrate smoother position estimates than our approach. Despite using the same number of particles as our approach, R-RBPF performs substantially worse. This observation is consistent with the widely recognized fact that particle filter methods alone require a large number of particles—that scales exponentially with the size of the state space—for good estimation performance [80]. However, for real-world applications, the computational costs associated with

Table 1 Experimental parameters

Parameter	Value	Parameter	Value
No. of particles	20	Confidence level α	0.95
No. of visual landmarks	50	GPS-Beidou ISB	17.6 m
Initial position uncertainty	5.0 m	Initial orientation uncertainty	0.2 rad
Initial velocity uncertainty	1.0 m/s	Initial bias uncertainty	1.0 rad/s
GNSS robust threshold	50 m	VO robust threshold	20 px
Propagation noise p_x	5.6 m	Propagation noise v_x	2.0 m/s
Propagation noise p_y	5.8 m	Propagation noise v_y	2.1 m/s
Propagation noise p_z	0.01 m	Propagation noise v_z	0.01 m/s
Propagation noise roll, pitch	0.02 rad	Propagation noise yaw	0.3 rad
Propagation noise bias	0.05 rad	Observation noise roll, pitch, yaw	0.02 rad
Observation noise velocity	2.2 m/s	Observation noise landmark	100.5 px
Observation noise pseudorange	5.3 m	Speed scale	0.5

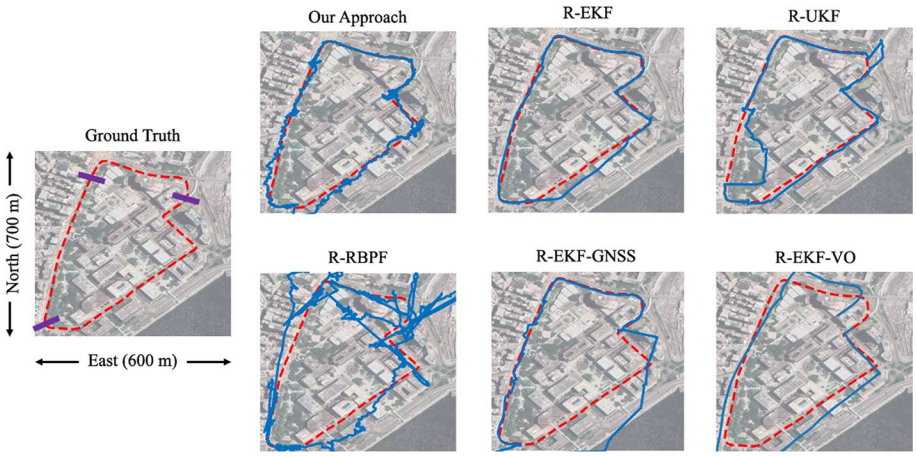


Fig. 4 Trajectory tracking plots of each approach (blue) compared to the ground truth positions (dashed red). Purple lines depict the boundaries of wide street, one-sided buildings, and medium urban regions shown in Fig. 3a. The trajectory estimated from our approach is visually closer to the ground truth trajectory in a majority of the regions compared to the baselines

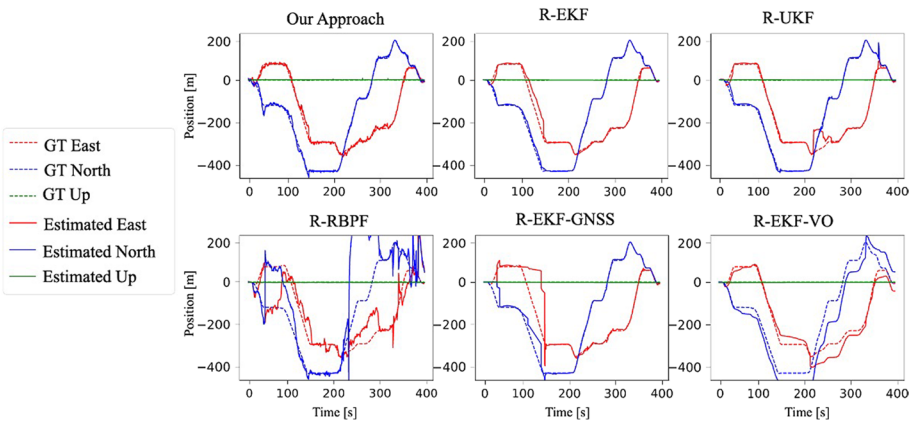


Fig. 5 Estimated positions over time along the east, north, and up directions compared to the ground truth positions over the entire trajectory of the dataset. The estimated positions from our approach show improved tracking performance and closely follow the ground truth positions compared to the baselines

increasing the number of particles may become prohibitively high. We also observe that, in general, baselines with multi-sensor measurements provide better positioning performance compared to those relying on single-sensor measurements.

To quantitatively evaluate the performance of our approach, we computed the mean, median, and 95th percentile of positioning error for all three sections of the trajectory. The results, presented in Table 2, demonstrate that our method produces positioning errors comparable to those of R-EKF and R-EKF-GNSS in the wide street and one-sided building regions. Additionally, our method exhibits smaller errors than all other baselines in these regions. In the medium urban region, our approach achieves smaller errors than all baselines except R-UKF. These results highlight the usability of our approach in real-world sensor fusion settings, where accurate state estimation is important.

Table 2 Positioning error statistics

Algorithm	Statistics	Wide street	One-sided buildings	Medium urban
Ours	Mean (m)	5.6	6.8	25.6
	Median (m)	4.5	4.9	19.2
	95th percentile (m)	19.3	22.1	51.7
R-EKF	Mean (m)	5.2	6.1	38.9
	Median (m)	4.1	5.8	23.3
	95th percentile (m)	22.3	12.9	85.4
R-UKF	Mean (m)	7.3	21.8	16.2
	Median (m)	4.3	15.3	14.1
	95th percentile (m)	44.0	81.2	40.7
R-RBPF naive	Mean (m)	81.5	80.2	81.3
	Median (m)	78.1	76.8	78.8
	95th percentile (m)	302.1	381.5	305.5
R-EKF-GNSS	Mean (m)	4.9	6.8	58.7
	Median (m)	4.3	4.9	39.9
	95th percentile (m)	8.2	16.4	255.4
R-EKF-VO	Mean (m)	74.3	67.1	32.3
	Median (m)	67.2	62.6	36.6
	95th percentile (m)	87.0	131.1	72.3

7.5 Position error bound evaluation

For qualitative evaluation of the position error bounding performance, we show plots that depict the estimated error bounds versus position error for each approach along the lateral, longitudinal, and vertical directions, as shown in Figs. 6, 7 and 8. The position error bounds lying in the upper half of the plot—separated by the dotted red line—successfully enclose the position error.

Our approach generates position error bounds that are more clustered in the upper half of the plots compared to the baselines in all regions, except for R-UKF in the medium urban region. For clarity purposes, we compare our approach qualitatively against R-EKF and R-UKF only, while other baselines are included in the quantitative comparisons.

For quantitative comparison, we employ the failure rate metric to evaluate the effectiveness of each method in bounding the position error. The failure rate metric specifies the fraction of instances where the estimated error bound fails to capture the actual position error, where a smaller value indicates better performance. The results are included in Tables 3, 4, and 5 for the failure rates along the lateral, longitudinal, and vertical directions, respectively.

Based on the presented results, our proposed approach outperforms all baseline filters, except for R-UKF in the medium urban region and R-EKF-GNSS in all three regions. R-EKF-GNSS shows promising performance, despite its comparatively lower positioning accuracy, due to its sole reliance on GNSS measurements. The EKF's Gaussian uncertainty modeling assumptions, which are used in R-EKF-GNSS, hold reasonably true for GNSS measurements. Similarly, R-UKF demonstrates good bounding performance in the medium urban region where it also has good

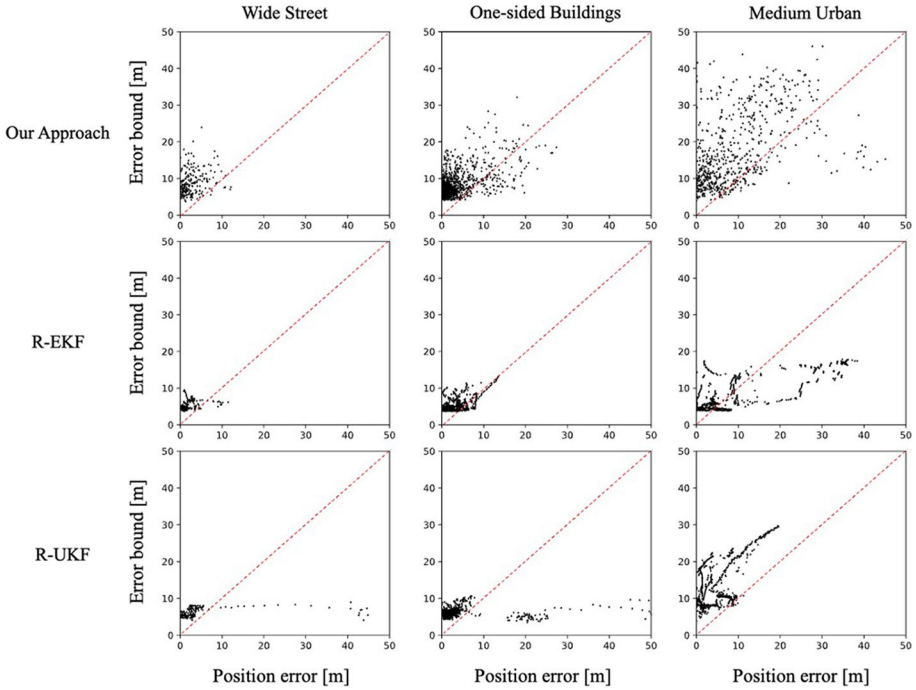


Fig. 6 Comparison of position error bounds and position error along the lateral direction for the wide street, one-sided building, and medium urban regions. The dashed red line separates the upper region where the bounds exceed position error and the lower region where the bounds are smaller. Our proposed approach demonstrates improved bounding performance as indicated by the majority of error bounds lying in the upper half

positioning performance. However, its performance deteriorates in the one-sided building and wide street regions, where it exhibits worse positioning accuracy.

The results indicate that both R-UKF and R-EKF exhibit a higher failure rate along the vertical direction, most likely due to overconfident uncertainty estimates. In contrast, our approach uses the uncertainty across multiple particles to generate more conservative error bounds in the vertical direction with considerably lower failure rates. These observations emphasize the significance of accurate uncertainty parameters in characterizing the position error bounds, particularly in challenging urban environments.

Overall, the qualitative and quantitative evaluations demonstrate that our proposed approach produces more reliable error bounds compared to the baselines. Our approach's improved position error bounding can be attributed to its capability to capture multi-modal uncertainty through multiple linearization points, which effectively captures uncertainty in situations where multiple plausible solutions exist based on the measurements. This reduces the proposed method's reliance on the choice of the uncertainty parameters, highlighting the potential of our approach for real-world applications in developing robust and reliable systems for autonomous navigation.

7.6 Computational statistics

We analyzed the average runtime per step for our proposed approach and the baselines, tabulated in Table 6.

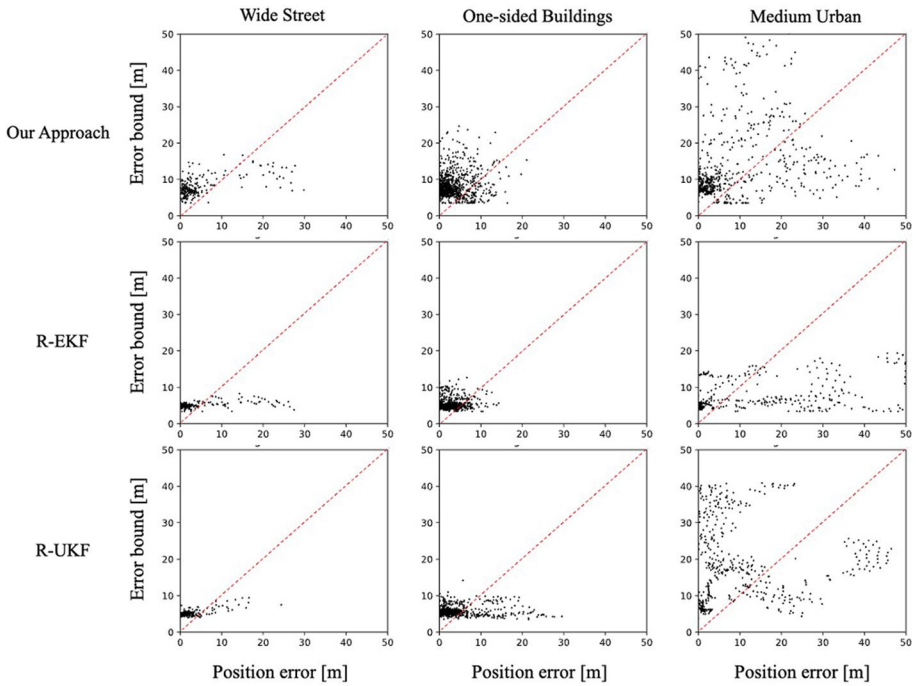


Fig. 7 Comparison of position error bounds and position error along the longitudinal direction for the wide street, one-sided building, and medium urban regions. The dashed red line separates the upper region where the bounds exceed position error and the lower region where the bounds are smaller. Our proposed approach demonstrates improved bounding performance as indicated by the majority of error bounds lying in the upper half

Our method and R-RBPF—both using 20 particles—exhibit an average step time that is less than twice that of the EKF and UKF-based baselines. This efficiency is attributed to the utilization of tensor operations to parallelize the EKF operations associated with the linearization points. From a theoretical standpoint, this aligns with the expected computational complexity of $O(nm^3 + nm^2 + n)$ —contributed by the robust filtering, weighting, and resampling steps—where n is the number of particles and m is the size of the measurement vector. It is worth noting that upon assuming optimal parallelization across all particles, the complexity reduces to $O(m^3 + n)$. This is comparable to the EKF's theoretical complexity of $O(m^3)$.

We also investigate the effect of the number of landmark features used in the measurement vector and the number of particles on the computational time, as shown in Fig. 9. Our findings indicate that the computational time stays under 100 ms till 200 features in the measurement vector and then increases at a sublinear rate till 1000 features. Furthermore, the computational time grows linearly with the particles at a slope smaller than 1, demonstrating the computational benefits of efficient tensor operations in our approach. These observations show that our approach can be employed in real-world applications without imposing a significant computational burden.

7.7 Discussion

Our experiments on positioning performance demonstrate that the proposed robust Bayesian filtering framework for positioning produces state estimates that are

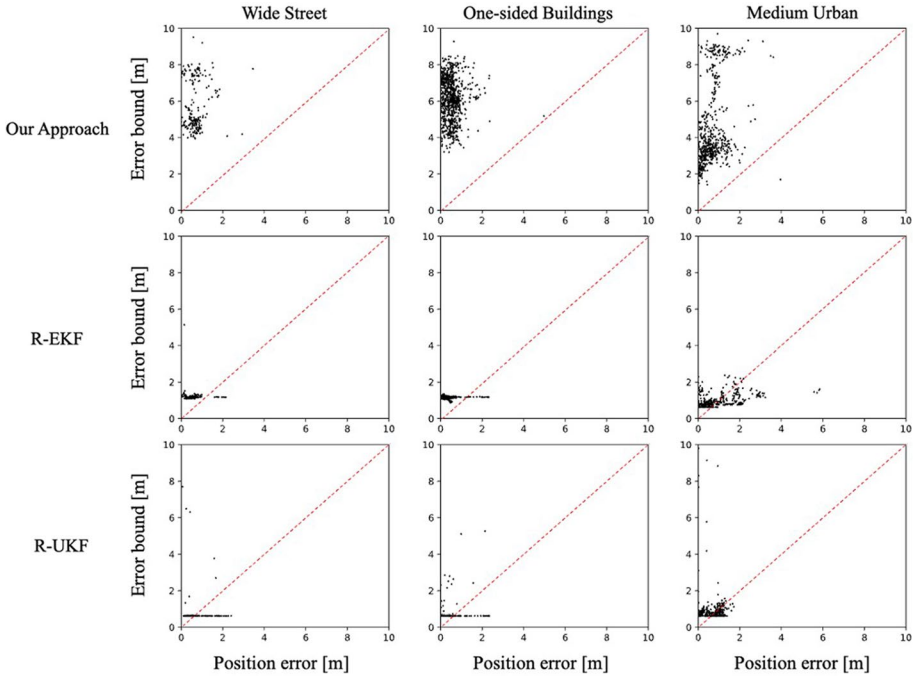


Fig. 8 Comparison of position error bounds and position error along the vertical direction for the wide street, one-sided building, and medium urban regions. The dashed red line separates the upper region where the bounds exceed position error and the lower region where the bounds are smaller. Our proposed approach demonstrates improved bounding performance as indicated by the majority of error bounds lying in the upper half

Table 3 Lateral failure rate for different sections of the trajectory

Trajectory section	Our algorithm	R-EKF	R-UKF	R-RBPF	R-EKF-GNSS	R-EKF-VO
Wide street	0.04	0.07	0.10	0.91	0.00	0.77
One-side buildings	0.12	0.14	0.21	0.70	0.01	0.59
Medium urban	0.16	0.43	0.02	0.85	0.01	0.81

Table 4 Longitudinal failure rate for different sections of the trajectory

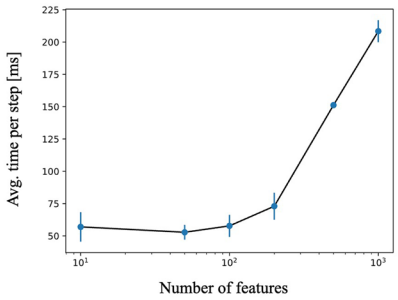
Trajectory section	Our algorithm	R-EKF	R-UKF	R-RBPF	R-EKF-GNSS	R-EKF-VO
Wide street	0.15	0.24	0.19	0.71	0.00	0.98
One-side buildings	0.14	0.18	0.19	0.62	0.02	0.72
Medium urban	0.24	0.49	0.18	0.39	0.00	0.87

Table 5 Vertical failure rate for different sections of the trajectory

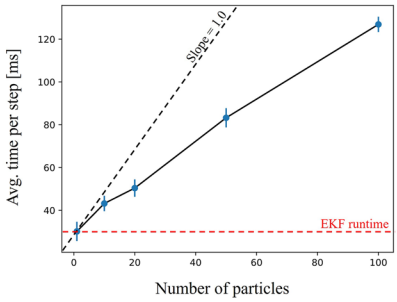
Trajectory section	Our algorithm	R-EKF	R-UKF	R-RBPF	R-EKF-GNSS	R-EKF-VO
Wide street	0.00	0.04	0.29	0.26	0.05	0.73
One-side buildings	0.00	0.02	0.05	0.32	0.02	0.47
Medium urban	0.01	0.28	0.37	0.08	0.02	0.65

Table 6 Average runtime per filter step for baselines and our algorithm

Algorithm	Time per step (milliseconds)
Ours	51
R-EKF	33
R-UKF	32
R-RBPF Naive	50
R-EKF-GNSS	26
R-EKF-VO	30



(a) Average runtime per step of our filter versus the number of features in measurement vector. The runtime remains under 100 ms for up to 200 features, showing computational efficiency.



(b) Average runtime per step of our filter versus the number of particles. The line corresponding to unity slope and EKF runtime is marked for reference. The runtime has a linear relationship with the number of particles with slope less than one, demonstrating the computational benefits of parallelization in our approach.

Fig. 9 Runtime in milliseconds of our algorithm as a function of the number of features in the measurement vector and as a function of the number of particles in the filter. The blue vertical lines denote standard deviation error bars. Our approach leverages parallelization through tensor operations to achieve low computational overhead and efficiency comparable to an EKF, as shown by the figures

competitive with the robust EKF and UKF baselines. Moreover, this performance can be achieved with minimal manual tuning through the proposed gradient descent-based optimization strategy. Additionally, our approach outperforms particle filter baselines with the same number of particles when applied to a real-world setting. However, the performance of our method is dependent on the underlying particle filter for tracking linearization points, which results in position estimates that are less smooth compared to the EKF and UKF-based methods. Therefore, it may be beneficial to explore improvements in particle filtering literature to further enhance the performance of our method. Incorporating improved methods for robust state estimation, such as factor graphs, in the framework is another direction for future research to enhance the overall performance.

Our position error bounding experiments indicate that the error bounds estimated using our approach are better correlated with the actual positioning error along each direction than the bounds from most baselines. It is important to note that a limitation of our approach is the difficulty in establishing theoretical guarantees on its performance

due to the need for knowing the individual error distributions of each measurement at each time instant. Nevertheless, our empirical results suggest that our approach can identify situations where the positioning error grows large, making it useful in preventing the unsafe use of the estimated location. Hence, our approach has practical utility in safety-critical applications.

Finally, our experiments on computational statistics illustrate that our proposed framework has comparable computational requirements to the underlying robust state estimation method, which is achieved using parallelization with tensor operations. Therefore, our approach can be integrated into real-time localization systems without imposing excessive computational overhead. Future work can further explore adaptively varying the number of tracked linearization points in our framework to improve computational efficiency.

8 Conclusions

In this paper, we introduced a robust Bayesian filtering framework that effectively captures multi-modal uncertainty in positioning using diverse sensor measurements while being robust to outliers. Our framework consists of two key components: the robust filter component, which uses techniques such as Extended Kalman Filters for efficient and robust state estimation, and the multi-modal uncertainty component, which tracks a probability distribution over points for linearizing the dynamics, measurement models, and robust cost in the robust filter component. We combined these two components using the Rao-Blackwell theorem to create a robust Bayesian filter that is resilient to measurement outliers and can capture multi-modal state uncertainty.

We validated our proposed filter on real-world data from a multi-sensor setup comprising a camera, GNSS, and AHRS. To tune the filter parameters, we utilized a gradient descent-based optimization strategy that leverages available measurements and ground truth position data. The results demonstrate our filter's competitive state estimation performance compared to existing filter-based robust state estimation methods and improved performance in bounding the position errors based on uncertainty. Furthermore, these improvements are achieved with less than twice the computational time of existing Kalman filter-based methods.

These results suggest that our proposed framework is suitable for real-world localization applications where the reliability of the estimated location is critical. Our approach enables more robust and accurate localization in complex environments, thus proving to be a valuable tool for autonomous navigation systems in urban environments.

Abbreviations

GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ISB	Inter-system bias
AHRS	Attitude and heading reference system
EKF	Extended Kalman filter
UKF	Unscented Kalman filter
DD	Double differenced
RTK	Real-time kinematic
IMM	Interacting multiple model
RBPF	Rao–Blackwellized particle filter
VO	Visual odometry
IMU	Inertial measurement unit
MDVO	Monocular depth-aided visual odometry

RANSAC Random sample and consensus

Acknowledgements
We would like to thank Asta Wu for peer reviewing this paper and the rest of the Stanford NAVLab for their insightful discussions and feedback.

Author contributions
SG conceived the idea and wrote the first draft of the manuscript and implemented the experiments; AM provided technical concepts and guided the research; and, SG, AM, and GG finalized the manuscript write-up, proofread and checked the technical correctness of the manuscript, and provided future perspectives of the research. All authors read and approved the final manuscript.

Funding
We would like to thank Ford Motor Company for providing the funds to support this project.

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 16 May 2023 Accepted: 25 September 2023
Published online: 06 October 2023

References

1. Y.J. Morton, F.S.T. Van Diggelen, J.J. Spilker, B.W. Parkinson (eds.), *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications* (First edition edn. Wiley/IEEE Press, Hoboken, 2020)
2. B. Kitt, F. Moosmann, C. Stiller, Moving on to dynamic environments: visual odometry using feature classification. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp. 5551–5556 (2010). https://doi.org/10.1109/IROS.2010.5650517. ISSN: 2153-0866
3. M. Camurri, M. Ramezani, S. Nobili, M. Fallon, Pronto: a multi-sensor state estimator for legged robots in real-world scenarios. Front. Robot. AI **7**, 68 (2020)
4. H. Du, W. Wang, C. Xu, R. Xiao, C. Sun, Real-time onboard 3d state estimation of an unmanned aerial vehicle in multi-environments using multi-sensor data fusion. Sensors **20**(3), 919 (2020)
5. Y. Jiang, S. Pan, Q. Meng, W. Gao, C. Ma, B. Yu, F. Jia, Robust Kalman filter enhanced by projection statistic detector for multi-sensor navigation in urban canyon environment. IEEE Sens. J. (2022). https://doi.org/10.1109/JSEN.2022.3230708
6. R. Sun, Y. Yang, K.-W. Chiang, T.-T. Duong, K.-Y. Lin, G.-J. Tsai, Robust IMU/GPS/VO integration for vehicle navigation in GNSS degraded urban areas. IEEE Sens. J. **20**(17), 10110–10122 (2020)
7. H.-P. Chiu, X.S. Zhou, L. Carlone, F. Dellaert, S. Samarasekera, R. Kumar, Constrained optimal selection for multi-sensor robot navigation using plug-and-play factor graphs. In: 2014 IEEE international conference on robotics and automation (ICRA), (IEEE, 2014), pp. 663–670
8. Q. Zeng, W. Chen, J. Liu, H. Wang, An improved multi-sensor fusion navigation algorithm based on the factor graph. Sensors **17**(3), 641 (2017)
9. W. Xiwei, X. Bing, W. Cihang, G. Yiming, L. Lingwei, Factor graph based navigation and positioning for control system design: a review. Chin. J. Aeronaut. **35**(5), 25–39 (2022)
10. W. Wen, X. Bai, Y.C. Kan, L.-T. Hsu, Tightly coupled GNSS/ins integration via factor graph and aided by fish-eye camera. IEEE Trans. Veh. Technol. **68**(11), 10651–10662 (2019)
11. F. Dellaert, Factor graphs: exploiting structure in robotics. Annu. Rev. Control Robot. Auton. Syst. **4**, 141–166 (2021)
12. M. Rhudy, Y. Gu, M.R. Napolitano, An analytical approach for comparing linearization methods in EKF and UKF. Int. J. Adv. Robot. Syst. **10**(4), 208 (2013)
13. L. Wei, C. Cappelle, Y. Ruichek, F. Zann, Intelligent vehicle localization in urban environments using EKF-based visual odometry and GPS fusion. IFAC Proc. Vol. **44**(1), 13776–13781 (2011)
14. C. Cappelle, M.E.B. El Najjar, D. Pomorski, F. Charpillet, Localisation in urban environment using gps and ins aided by monocular vision system and 3d geographical model. In: 2007 IEEE intelligent vehicles symposium, (IEEE, 2007), pp. 811–816
15. S.V.S. Chauhan, G.X. Gao, Joint gps and vision estimation using an adaptive filter. In: Proceedings of the 30th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2017), pp. 808–812 (2017)

16. A. Shetty, G.X. Gao, Vision-aided measurement level integration of multiple GPS receivers for uavs. In: Proceedings of the 28th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2015), pp. 834–840 (2015)
17. M. Dawood, C. Cappelle, M.E. El Najjar, M. Khalil, D. Pomorski, Vehicle geo-localization based on imm-ukf data fusion using a GPS receiver, a video camera and a 3d city model. In: 2011 IEEE intelligent vehicles symposium (IV), (IEEE, 2011), pp. 510–515
18. S. Yazdkhasti, J.Z. Sasiadek, Multi sensor fusion based on adaptive Kalman filtering. In: Advances in aerospace guidance, navigation and control: selected papers of the fourth ceas specialist conference on guidance, navigation and control held in Warsaw, Poland, April 2017, (Springer, 2018), pp. 317–333
19. J.S. Liu, R. Chen, Sequential monte Carlo methods for dynamic systems. J. Am. Stat. Assoc. **93**(443), 1032–1044 (1998). https://doi.org/10.1080/01621459.1998.10473765
20. A. Mohanty, S. Gupta, G.X. Gao, A particle-filtering framework for integrity risk of GNSS-camera sensor fusion. Navigation **68**(4), 709–726 (2021)
21. H. Li, F. Nashashibi, G. Toulminet, Localization for intelligent vehicle by fusing mono-camera, low-cost gps and map data. In: 13th international IEEE conference on intelligent transportation systems, pp. 1657–1662 (2010). https://doi.org/10.1109/ITSC.2010.5625240
22. M. Rosencrantz, G. Gordon, S. Thrun, Decentralized sensor fusion with distributed particle filters. arXiv preprint arXiv:1212.2493 (2012)
23. M.A. Lee, B. Yi, R. Martín-Martín, S. Savarese, J. Bohg, Multimodal sensor fusion with differentiable filters. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), (IEEE, 2020), pp. 10444–10451
24. F. Daum, J. Huang, Curse of dimensionality and particle filters. In: 2003 IEEE aerospace conference proceedings (Cat. No. 03TH8652), vol. 4, (IEEE, 4–197941993) (2003)
25. E. Mazor, A. Averbuch, Y. Bar-Shalom, J. Dayan, Interacting multiple model methods in target tracking: a survey. IEEE Trans. Aerosp. Electron. Syst. **34**(1), 103–123 (1998). https://doi.org/10.1109/7.640267
26. M. Ulmschneider, C. Gentner, R. Faragher, T. Jost, Gaussian mixture filter for multipath assisted positioning. In: Proceedings of the 29th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2016), (Institute of Navigation, Portland, Oregon, 2016), pp. 1263–1269
27. K. Jo, K. Chu, M. Sunwoo, Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning. IEEE Trans. Intell. Transp. Syst. **13**(1), 329–343 (2012). https://doi.org/10.1109/TITS.2011.2171033
28. M. Kheirandish, E.A. Yazdi, H. Mohammadi, M. Mohammadi, A fault-tolerant sensor fusion in mobile robots using multiple model Kalman filters. Robot. Auton. Syst. **161**, 104343 (2023). https://doi.org/10.1016/j.robot.2022.104343
29. W. Wen, X. Bai, L.-T. Hsu, T. Pfeifer, Gnss/lidar integration aided by self-adaptive gaussian mixture models in urban scenarios: an approach robust to non-gaussian noise. In: 2020 IEEE/ION Position, location and navigation symposium (PLANS), (IEEE, 2020), pp. 647–654
30. J. Cheng, Y. Gao, J. Wu, An adaptive integrated positioning method for urban vehicles based on multi-task heterogeneous deep learning during GNSS outages. IEEE Sens. J. (2023). https://doi.org/10.1109/JSEN.2023.3302796
31. C. Li, S.L. Waslander, Towards End-to-end learning of visual inertial odometry with an EKF. In: 2020 17th conference on computer and robot vision (CRV), pp. 190–197 (2020). https://doi.org/10.1109/CRV50864.2020.00033
32. J. Liu, G. Guo, Vehicle localization during GPS outages with extended Kalman filter and deep learning. IEEE Trans. Instrum. Meas. **70**, 1–10 (2021). https://doi.org/10.1109/TIM.2021.3097401
33. C. Chen, C.X. Lu, B. Wang, N. Trigoni, A. Markham, DynaNet: neural Kalman dynamical model for motion estimation and prediction. IEEE Trans. Neural Netw. Learn. Syst. **32**(12), 5479–5491 (2021). https://doi.org/10.1109/TNNLS.2021.3112460
34. Y.-C. Lin, Y.-W. Huang, K.-W. Chiang, A neural-KF hybrid sensor fusion scheme for INS/GPS/odometer integrated land vehicular navigation system. In: Proceedings of the 19th international technical meeting of the satellite division of the institute of navigation (ION GNSS 2006), pp. 2174–2181, (2006)
35. K. Murphy, S. Russell, Rao-blackwellised particle filtering for dynamic bayesian networks. In: Sequential Monte Carlo methods in practice, pp. 499–515 (2001)
36. G. Hendeby, R. Karlsson, F. Gustafsson, The rao-blackwellized particle filter: a filter bank implementation. EURASIP J. Adv. Sign. process. **2010**, 1–10 (2010)
37. S. Gupta, A. Mohanty, G. Gao, Getting the best of particle and Kalman filters: GNSS sensor fusion using rao-blackwellized particle filter. In: Proceedings of the 35th international technical meeting of the satellite division of the institute of navigation (ION GNSS+ 2022), Denver, Colorado, pp. 1610–1623 (2022). https://doi.org/10.33012/2022.18470
38. T. Reid, S. Houts, R. Cammarata, G. Mills et al., Localization requirements for autonomous vehicles. SAE Intl. J CAV **2**(3), 173–190 (2019). https://doi.org/10.4271/12-02-03-0012
39. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, PyTorch: an imperative style, high-performance deep learning library. https://pytorch.org/. Accessed 20 Apr 2023 (2019)
40. O. Stepanov, V. Vasiliev, A. Toropov, Map-aided navigation algorithms taking into account the variability of position errors of the corrected navigation system. In: 2022 29th saint petersburg international conference on integrated navigation systems (ICINS), (IEEE, 2022), pp. 1–5
41. A.S. Stordal, H.A. Karlsen, G. Nævdal, H.J. Skaug, B. Vallès, Bridging the ensemble Kalman filter and particle filters: the adaptive gaussian mixture filter. Comput. Geosci. **15**, 293–305 (2011)
42. A. Giremus, A. Doucet, V. Calmettes, J.-Y. Tourneret, A rao-blackwellized particle filter for ins/gps integration. In: 2004 IEEE international conference on acoustics, speech, and signal processing, vol. 3, p. 964 (2004). https://doi.org/10.1109/ICASSP.2004.1326707
43. P. Vernaza, D.D. Lee, Robust GPS/ins-aided localization and mapping via GPS bias estimation, in *Experimental Robotics. Springer Tracts in Advanced Robotics*, vol. 39, ed. by O. Khatib, V. Kumar, D. Rus (Springer, Berlin, Heidelberg, 2008), pp.125–135
44. A. Corenflos, J. Thornton, G. Deligiannidis, A. Doucet, Differentiable particle filtering via entropy-regularized optimal transport. In: International conference on machine learning, (PMLR, 2021), pp. 2100–2111

45. L.-T. Hsu, W.W. Nobuaki Kubo, W. Chen, Z. Liu, T. Suzuki, J. Meguro, Urbannav: an open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas. In: Proceedings of the 34th international technical meeting of the satellite division of the institute of navigation, pp. 226–256 (2021)
46. S. Chen, Kalman filter for robot vision: a survey. IEEE Trans. Ind. Electron. **59**(11), 4409–4420 (2011)
47. S.J. Julier, J.K. Uhlmann, Unscented filtering and nonlinear estimation. Proc. IEEE **92**(3), 401–422 (2004)
48. L.-T. Hsu, Analysis and modeling GPS NLOS effect in highly urbanized area. GPS Solut. **22**(1), 7 (2018)
49. T. Cimiega, S. Badri-Hoeher, Enhanced state estimation based on particle filter and sensor data with non-gaussian and multimodal noise. IEEE Access **9**, 60704–60712 (2021)
50. Q. Li, J.P. Queralta, T.N. Gia, Z. Zou, T. Westerlund, Multi-sensor fusion for navigation and mapping in autonomous vehicles: accurate localization in urban environments. Unman. Syst. **8**(03), 229–237 (2020)
51. P. Meer, D. Mintz, A. Rosenfeld, D.Y. Kim, Robust regression methods for computer vision: a review. Int. J. Comput. Vis. **6**, 59–70 (1991)
52. M.A. Gandhi, L. Mili, Robust Kalman filter based on a generalized maximum-likelihood-type estimator. IEEE Trans. Sign. Process. **58**(5), 2509–2520 (2009)
53. G. Agamennoni, J.I. Nieto, E.M. Nebot, An outlier-robust Kalman filter. In: 2011 IEEE international conference on robotics and automation, (IEEE, 2011), pp. 1551–1558
54. L. Chang, B. Hu, G. Chang, A. Li, Huber-based novel robust unscented Kalman filter. IET Sci. Meas. Technol. **6**(6), 502–509 (2012)
55. C.-H. Park, J.-H. Chang, Robust LMmedS-based WLS and Tukey-based EKF algorithms under LOS/NLOS mixture conditions. IEEE Access **7**, 148198–148207 (2019)
56. H. Wang, H. Li, W. Zhang, J. Zuo, H. Wang, A unified framework for m-estimation based robust Kalman smoothing. Sign. Process. **158**, 61–65 (2019)
57. F. Gustafsson, Particle filter theory and practice with positioning applications. IEEE Aeros. Electron. Syst. Mag. **25**(7), 53–82 (2010). https://doi.org/10.1109/MAES.2010.5546308. (**Accessed 2022-09-09**)
58. J. Elfring, E. Torta, R. van de Molengraft, Particle filters: a hands-on tutorial. Sensors **21**(2), 438 (2021)
59. P. Karkus, D. Hsu, W.S. Lee, Particle filter networks with application to visual localization. In: Conference on robot learning, (PMLR, 2018), pp. 169–178
60. Y. Junjun, P. Dongliang, G. Quanbo, Sequential fusion for asynchronous multi-sensor system based on Kalman filter. In: 2009 Chinese control and decision conference, (IEEE, 2009), pp. 4677–4681
61. S.-M. Oh, Multisensor fusion for autonomous uav navigation based on the unscented Kalman filter with sequential measurement updates. In: 2010 IEEE conference on multisensor fusion and integration, (IEEE, 2010), pp. 217–222
62. K. Yokoyama, K. Morioka, Autonomous mobile robot with simple navigation system based on deep reinforcement learning and a monocular camera. In: 2020 IEEE/SICE international symposium on system integration (SII), (IEEE, 2020), pp. 525–530
63. M. He, C. Zhu, Q. Huang, B. Ren, J. Liu, A review of monocular visual odometry. Vis. Comput. **36**(5), 1053–1065 (2020). https://doi.org/10.1007/s00371-019-01714-6. (**Accessed 2022-09-16**)
64. D. Burschka, E. Mair, Direct pose estimation with a monocular camera. In: Robot vision: second international workshop, RobVis 2008, Auckland, New Zealand, February 18–20, 2008. Proceedings 2, (Springer, 2008), pp. 440–453
65. H. Zhan, C.S. Weerasekera, J.-W. Bian, I. Reid, Visual odometry revisited: what should be learnt? In: 2020 IEEE international conference on robotics and automation (ICRA), (IEEE, 2020), pp. 4203–4210
66. M. Geng, S. Shang, B. Ding, H. Wang, P. Zhang, Unsupervised learning-based depth estimation-aided visual slam approach. Circuits Syst. Sign. Process. **39**, 543–570 (2020)
67. D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: self-supervised interest point detection and description. CVPR Workshop, (2017). https://doi.org/10.48550/ARXIV.1712.07629
68. M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981). https://doi.org/10.1145/358669.358692
69. C. Zhao, Q. Sun, C. Zhang, Y. Tang, F. Qian, Monocular depth estimation based on deep learning: an overview. Sci. China Technol. Sci. **63**(9), 1612–1627 (2020). https://doi.org/10.1007/s11431-020-1582-8. (**Accessed 2022-09-10**)
70. C. Godard, O. Mac Aodha, M. Firman, G.J. Brostow, Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp. 3828–3838 (2019)
71. X.X. Lu, A review of solutions for perspective-n-point problem in camera pose estimation. J. Phys.: Conf. Ser. **1087**, 052009 (2018)
72. N. Nadarajah, P.J. Teunissen, N. Raziq, Beidou inter-satellite-type bias evaluation and calibration for mixed receiver attitude determination. Sensors **13**(7), 9435–9463 (2013)
73. P. Misra, P.K. Enge, The global positioning system: 6 signals, measurements, and performance. Int. J. Wirel. Inf. Netw. **1**, 83–105 (1994). https://doi.org/10.1007/BF02106512
74. R. Sun, Z. Zhang, Q. Cheng, W.Y. Ochieng, Pseudorange error prediction for adaptive tightly coupled GNSS/IMU navigation in urban areas. GPS Solut. **26**, 1–13 (2022)
75. G. Revach, N. Shlezinger, T. Locher, X. Ni, R.J. van Sloun, Y.C. Eldar, Unsupervised learned Kalman filtering. In: 2022 30th European signal processing conference (EUSIPCO), (IEEE, 2022), pp. 1571–1575
76. G. Revach, N. Shlezinger, X. Ni, A.L. Escoriza, R.J. Van Sloun, Y.C. Eldar, Kalmannet: neural network aided Kalman filtering for partially known dynamics. IEEE Trans. Sign. Process. **70**, 1532–1547 (2022)
77. Y.-G. Choi, J. Lim, A. Roy, J. Park, Positive-definite correction of covariance matrix estimators via linear shrinkage (2015)
78. D.P. Kingma, J. Ba, Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
79. A.E. Beaton, J.W. Tukey, The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data. Technometrics **16**(2), 147–185 (1974). https://doi.org/10.1080/00401706.1974.10489171
80. A. Doucet, A.M. Johansen, A tutorial on particle filtering and smoothing: fifteen years later. Handb. Nonlinear Filter. **12**(656–704), 3 (2009)