Stein ADMM: 協調自己位置推定のためのパーティクルフィルタ

Tomoki Arita[1†] and Toru Namerikawa[2]

[1]School of Integrated Design Engineering, Keio University, Kanagawa, Japan
[2]Department of System Design Engineering, Keio University, Kanagawa, Japan
(Tel: +81-45-563-1151; E-mail: arita@keio.jp, namerikawa@sd.keio.ac.jp)

Abstract: 農地や森林などの単調で広大な屋外環境では、画像マッチングによるグローバル自己位置推定は、累積誤差や外れ値の影響により失敗することが多い。特に複数エージェントによる協調推定では、1 つのエージェントの誤った状態推定が全エージェントに影響し、システム全体の誤った推定確率がエージェントごとの外れ値の割合に応じて指数関数的に増加する。本研究では、シュタインパーティクルフィルタを用いて、従来の画像特徴マッチングに位置ベースの尤度を組み込みながら、複数エージェント間で推定状態のコンセンサスを達成するビジュアル慣性航法システム（VINS）手法を提案する。提案手法は、複数エージェント間のコンセンサス制約とマルチモーダル分布を同時に処理することを可能にし、従来手法では困難だった外れ値を含む環境での安定した位置コンセンサスを実現する。シミュレーションと実世界実験（単一エージェント）の結果は、SPF を使用することで、分布近似と勾配情報を活用して、既存手法よりも堅牢で柔軟な自己位置推定が可能になることを示している。コードとデータセットはこの URL で入手可能である。

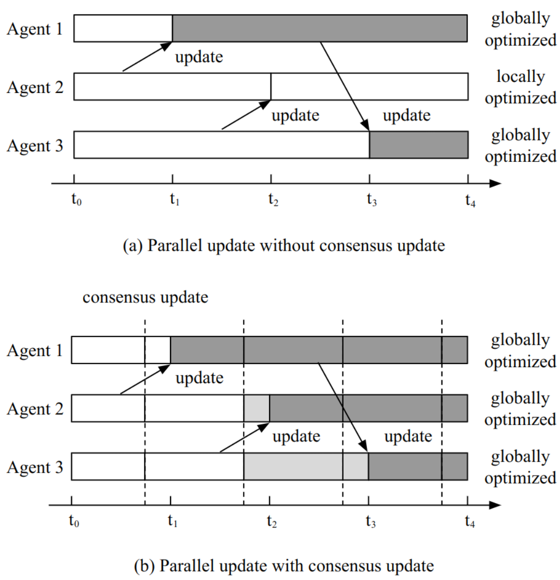
Keywords: Collaborative Localization, Unmanned Aerial Vehicles, Stein Particle Filter, Monotone Environments, Visual-Inertial Navigation System.

1. INTRODUCTION

多くの自律移動可能なロボットは，動作の基盤となる自己位置推定を必要とする。また，自律移動ロボットの社会への浸透に伴って，過酷環境や山間地域等 GPS/GNSS が利用できない環境での自己位置推定技術が数多く研究されている。自己位置推定には，力学制御や局所の動作計画などに利用する局所の自己位置推定の階層と，一貫した環境地図の作成や広域の経路計画に利用する大域の自己位置推定の階層が存在するが，GPS/GNSS が利用できない環境における大域の自己位置推定では，主に LiDAR やカメラ等によって得られた点群及び画像データを種々のマッチング手法で構造化す ることにより自己位置推定を達成する手法が一般的であ る。実用的には，カメラと IMU(慣性計測ユニット) を組 み合わせる Visual Inertial System(VINS) が安価にハードウェアを実装できる点から多く利用されている。

更に，自律移動ロボットの普及によって，複数のエージェントが同時に動作するマルチエージェント環境が近年主要な研究分野となっている。GPS/GNSS が使用で きない過酷環境や山間地域においても，システム全体と しての対故障性やそれぞれの目的におけるスケーラビリ ティの観点から，いくつかの研究でマルチエージェント による協調作業が試みられている 。

これらの背景から，GPS/GNSS を用いない大域自己位置推定手法が求められているところであるが，農地や森林など単調かつ広大な野外環境下では，誤差の蓄積と外れ値の影響により画像のマッチングによる大域の自己位置推定が十分に機能しない。特に複数のエージェン トによる協調推定においては，各エージェントの誤った状態推定が全エージェントに影響する為，各エージェン トの外れ値割合に伴ってシステム全体が誤った推定を する確率は指数関数的に上昇する。そこで本研究では， Stein Particle Filter を用いることで，



(a) Parallel update without consensus update

(b) Parallel update with consensus update

Fig. 1 Parallel Update

従来の画像特徴量 のマッチングに位置情報による尤度を考慮しつつ，複数 のエージェント間で推定状態を合意する VINS 手法を提案する。

2. RELATED WORK

This research is related to several fields. First, there is the field of Visual Inertial Systems (VINS), which combines cameras and Inertial Measurement Units (IMUs) for self-localization in environments where GPS/GNSS is unavailable. Qin et al. [1] proposed VINS-Mono, a robust and versatile VINS using a monocular camera and IMU, which is a representative study in this field.

Next, there is the field of multi-robot collaborative

† Tomoki Arita is the presenter of this paper.

SLAM (Simultaneous Localization and Mapping). Chen et al. [2] summarized an overview of multi-robot collaborative SLAM from the perspective of data fusion, classifying and comparing various approaches. Zhou et al. [3] conducted research on swarms of micro aerial robots in actual outdoor environments, demonstrating challenges and solutions for collaborative self-localization in the real world.

Additionally, there is the field of particle filters and variational inference, which are methods for solving non-Gaussian and nonlinear probabilistic state estimation problems. Liu and Wang [4] proposed Stein Variational Gradient Descent (SVGD), a general Bayesian inference algorithm that forms the basis of the Stein Particle Filter used in this study. Koide et al. [5] conducted research on accelerating the Stein Particle Filter using GPUs, achieving real-time 6-degree-of-freedom position estimation.

Visual Inertial Systems (VINS) combine cameras and Inertial Measurement Units (IMUs) for self-localization and are widely used in environments where GPS/GNSS is unavailable. Qin et al. [1] proposed VINS-Mono, a robust and versatile VINS using a monocular camera and IMU, integrating feature point tracking, IMU pre-integration, loop detection, and other functions. Bloesch et al. [6] proposed an extended Kalman filter-based VINS using direct photometric feedback, achieving a method that does not depend on feature point extraction. Forster et al. [7] proposed a real-time VINS using pre-integration on manifolds, demonstrating an efficient method for processing IMU measurements. These studies focus on single agents and do not adequately address the challenges of multiple agent collaboration or monotone environments.

Multi-robot collaborative SLAM is a method where multiple robots cooperate to map the environment and estimate their positions. Chen et al. [2] summarized an overview of multi-robot collaborative SLAM from the perspective of data fusion, classifying and comparing various approaches such as centralized, distributed, and decentralized. Zhou et al. [3] conducted research on swarms of micro aerial robots in actual outdoor environments, demonstrating challenges and solutions for collaborative self-localization in the real world. Xu et al. [8] proposed D2SLAM, a distributed and decentralized collaborative Visual-Inertial SLAM system, achieving efficient collaborative self-localization for aerial swarms. These studies focus on the collaboration of multiple agents but do not adequately address robustness to outliers in monotone environments or the processing of multimodal distributions.

Particle filters and variational inference are methods for solving non-Gaussian and nonlinear probabilistic state estimation problems. Liu and Wang [4] proposed Stein Variational Gradient Descent (SVGD), a general Bayesian inference algorithm that forms the basis of the Stein Particle Filter used in this study. Maken et al. [9] proposed the Stein Particle Filter, a method for nonlinear and non-Gaussian state estimation, solving the problem of particle degradation due to resampling in conventional particle filters. Koide et al. [5] conducted research on accelerating the Stein Particle Filter using GPUs, achieving real-time 6-degree-of-freedom position estimation. These studies focus on the theory and application of particle filters and variational inference but do not adequately address the combination with multiple agent collaboration or consensus problems.

Related publications from our lab include:

- "Cooperative Visual-Inertial SLAM for Multiple UAVs"
- "Robust Particle Filter-Based Cooperative Localization in GPS-Denied Environments"
- "Multi-Agent Formation Control with Distributed Optimization"

Standard benchmarks in the field of collaborative self-localization include the following. First, the EuRoC MAV dataset is a standard dataset containing stereo images, IMU data, and ground truth position information collected by micro aerial vehicles (MAVs), commonly used for evaluating single-agent VINS. The KITTI Vision Benchmark is a dataset collected from autonomous driving vehicles, containing stereo images, LiDAR scans, and GPS data, widely used for evaluating self-localization in outdoor environments. The TUM RGB-D dataset provides RGB-D images and ground truth position information in indoor environments, suitable for evaluating vision-based SLAM. Recently, synthetic datasets using simulators such as AirSim, CARLA, and Gazebo have also increased, particularly used for evaluating collaborative self-localization of multiple agents. However, there are still few benchmarks specialized for collaborative self-localization of multiple UAVs in monotone environments, and the development of datasets corresponding to specific challenges like those in this study remains a future task.

The main differences between the proposed method and similar methods are as follows.

First, the difference with D2SLAM [8] lies in the representation of probability distributions. D2SLAM is a graph optimization-based method that adopts an approach seeking a single optimal solution. In contrast, the proposed method represents the probability distribution using a particle filter and updates it using a gradient method based on Stein Variational Gradient Descent (SVGD). This enables proper handling of multimodal distributions and robust estimation even in environments with outliers.

Next, the difference with MegaParticles [5] lies in the handling of multiple agent collaboration and consensus problems. MegaParticles focuses on accelerating the Stein Particle Filter using GPUs but is limited to single-agent position estimation. The proposed method combines Relaxed ADMM to realize a framework that can simultaneously consider consensus constraints between multiple agents.

Additionally, the difference with conventional Cooperative VINS methods [10] lies in robustness to outliers in monotone environments. Conventional Cooperative VINS methods primarily focus on improving feature point matching accuracy and efficient information sharing but do not adequately consider robustness to outliers in monotone environments. The proposed method introduces hierarchical likelihood, using NetVLAD features for wide-area and Superpoint features for local areas to improve feature matching accuracy in monotone environments.

3. PROBLEM STATEMENT

The task addressed in this study is "Collaborative Localization of UAVs in Monotone Environments." In this task, multiple UAVs (Unmanned Aerial Vehicles) cooperate to estimate their positions in monotone and vast outdoor environments (such as farmlands and forests) where GPS/GNSS is unavailable. Each UAV is equipped with a camera and an Inertial Measurement Unit (IMU) and estimates its position through these sensor information and communication with other UAVs. Particularly in monotone environments, there are many similar visual features, making position estimation through image matching prone to errors, and stable self-localization is required even in situations where these outliers exist.

In this task, it is desirable for each UAV to accurately estimate its 6-degree-of-freedom position and orientation (state on SE(3)). Specifically, a solution satisfying the following conditions is desirable:

1. The estimated position and orientation of each UAV are close to the true position and orientation (low estimation error)
2. The relative position relationships between multiple UAVs are accurately estimated (consistent estimation)
3. Stable estimation is maintained even in the presence of outliers (incorrect matching) in monotone environments (robustness)
4. Multimodal distributions can be properly represented and processed (ambiguity representation capability)
5. Estimated states are agreed upon between multiple UAVs (consensus)
6. Computational efficiency capable of real-time operation (real-time capability)

A solution satisfying these conditions is considered effective in the collaborative self-localization task in monotone environments.

As shown in Fig. 1, Collaborative Visual Inertial System (CoVINS) is a representative example of collaborative self-localization by multiple UAVs. There are N agents equipped with cameras and Inertial Measurement Units (IMUs), and each agent transmits the image information observed by its camera to adjacent agents. The agent receiving the image information searches within the database held by each agent, detects images observing the same landmark, and sends back the relative position between camera frames in the two images. In the context of graph optimization, each agent creates a pose graph, which is a graph of relative positions, using the obtained relative position information. If a pose graph can be generated that reproduces each obtained relative position as a result of optimization, it can be said that consistent position estimation has been achieved.

Fig. 2 Collaborative Visual Inertial System (CoVINS)

The inputs given in this task are as follows:

1. Camera images from each UAV: Visual information obtained from cameras mounted on UAVs
2. IMU data from each UAV: Inertial information (acceleration and angular velocity) obtained from accelerometers and gyroscopes
3. Communication data between UAVs: Image information and relative position information received from other UAVs
4. Initial position and orientation: Initial state of each UAV (optional, if available)
5. Prior information about the environment: Maps, feature point databases, etc. (optional, if available)

Using these inputs, each UAV estimates its position and orientation, and collaborates with other UAVs to improve the overall position estimation accuracy.

The outputs required in this task are as follows:

1. 6-degree-of-freedom position and orientation estimation values for each UAV: Position and orientation of each UAV represented as states on SE(3)
2. Estimation uncertainty: Probability distribution of position and orientation estimation (represented as a set of particles)
3. Relative position relationships between UAVs: Relative position and orientation relationships between each UAV
4. Confidence indicator: Indicator showing the reliability of estimation results (optional)
5. Environment map: Feature point map of the environment (optional, if performing SLAM)

Using these outputs, each UAV can understand its position and orientation and collaborate to accomplish its mission. Additionally, by outputting the probability distribution representing the estimation uncertainty, it can appropriately respond to strongly multimodal problems.

The main terms used in this research are defined as follows:

- UAV (Unmanned Aerial Vehicle): An unmanned aircraft. In this study, it refers to an autonomous flyable drone equipped with a camera and IMU.
- VINS (Visual Inertial System): A self-localization system combining a camera and IMU.
- CoVINS (Collaborative Visual Inertial System): A system where multiple UAVs collaboratively perform self-localization.
- SE(3): 3D Special Euclidean group. A mathematical framework for representing 6-degree-of-freedom position and orientation.
- SO(3): 3D Special Orthogonal group. A mathematical framework for representing 3D rotation.
- Stein Particle Filter (SPF): A particle filter based on Stein Variational Gradient Descent (SVGD).
- SVGD (Stein Variational Gradient Descent): A variational inference method that minimizes the Kullback-Leibler divergence using the Stein operator.
- Relaxed ADMM (Alternating Direction Method of Multipliers): A method for solving convex optimization problems with consensus constraints.
- Multimodal distribution: A probability distribution with multiple peaks. In monotone environments, multiple different positions may generate similar observations, appearing as multiple peaks in the probability distribution.
- Outlier: A data point that deviates significantly from the true value, caused by incorrect matching or observation.
- NetVLAD: A neural network architecture for extracting global features from images.
- Superpoint: A neural network for extracting local feature points from images.

This study assumes the following points and does not address these challenges:

1. It is assumed that each UAV is equipped with a camera and IMU, and other sensor configurations are not considered.
2. Communication between UAVs is ideal, and problems such as communication delays or disruptions are not addressed.
3. The dynamics model and control of each UAV are not considered, focusing purely on the state estimation problem.
4. The presence of dynamic objects in the environment is not considered, assuming a static environment.
5. It is assumed that the calibration of the camera and IMU of each UAV has been performed in advance.
6. Computational resource constraints are considered, but it is assumed that a GPU is available.
7. It is assumed that the initial position and orientation are roughly known, and the global initialization problem is not addressed.
8. The focus is on the difficulty of feature point matching in monotone environments, and other environmental factors (lighting changes, weather changes, etc.) are not considered.

By making these assumptions, this study can focus on the core challenges of collaborative self-localization in monotone environments.

The evaluation metrics used for this task are as follows:

1. Absolute Position Error (APE): Euclidean distance between the estimated position and the true position (ground truth). Unit is meters (m).
2. Absolute Orientation Error (AOE): Angular difference between the estimated orientation and the true orientation. Unit is degrees ($^\circ$) or radians (rad).
3. Relative Position Error (RPE): Estimation error of relative positions between UAVs. Unit is meters (m).
4. Success Rate: Percentage of successful accurate position estimation in environments with outliers. Unit is percent (%).
5. Convergence Time: Time until the particle distribution converges. Unit is seconds (s).
6. Computation Time: Computation time per step. Unit is milliseconds (ms).
7. Outlier Tolerance: Maximum proportion of outliers for which the system can maintain accurate position estimation. Unit is percent (%).

Using these evaluation metrics, the performance of the proposed method is compared with existing methods to evaluate the effectiveness of collaborative self-localization in monotone environments.

In this study, both real-world experiments and simulations are used, but simulations are emphasized particularly for systematically evaluating the impact of outliers. The reasons are as follows:

1. Controlled experimental environment: In simulations, the proportion and distribution of outliers can be accurately controlled, allowing for systematic evaluation of the robustness of the method. In real-world experiments, it is difficult to control the occurrence of outliers.

2. Accuracy of ground truth: In simulations, completely accurate ground truth can be obtained, providing high reliability for evaluation. In real-world experiments, the ground truth itself may contain errors.
3. Reproducibility and fairness: Simulations are completely reproducible, allowing for fair comparison between different methods. In real-world experiments, reproducibility may decrease due to changes in environmental conditions.
4. Safety and cost: Real-world experiments using multiple UAVs have risks of collision and equipment damage, and are also costly. Simulations can be conducted without these risks.
5. Efficiency of parameter exploration: In simulations, numerous parameter settings can be efficiently tested. In real-world experiments, each setting trial requires time and resources.

However, since simulations alone cannot completely reproduce the complexity of the real world, limited real-world experiments (single-agent) are also conducted to confirm the effectiveness of the proposed method. Ultimately, the effectiveness and scalability of the proposed method in the real world will be further verified through larger-scale real-world experiments.

4. METHODOLOGY

We propose a Cooperative Visual Inertial System (CoVINS) method that incorporates position-based likelihood into conventional image feature matching using the Stein Particle Filter. This study extends the Stein Particle Filter (SPF) and Relaxed Alternating Direction Method of Multipliers (ADMM) to propose a new framework called Stein Relaxed ADMM for collaborative self-localization in monotone environments. SPF is a method for numerically analyzing non-Gaussian and nonlinear probabilistic state estimation problems, extending the basic framework proposed by Maken et al. [9]. Additionally, Relaxed ADMM is a distributed optimization method proposed by Bastianello et al. [11], which we extend to apply to consensus problems of probability distributions.

The extensions made in the proposed method are widely applicable to other existing methods for the following reasons:

1. Generalization of consensus problems for probability distributions: The proposed method provides a general framework for applying Relaxed ADMM to consensus problems of probability distributions. This framework can be applied not only to particle filters but also to other probability distribution representation methods such as Gaussian mixture models and variational autoencoders.
2. Generalization of gradient calculation on SE(3): The proposed method generalizes gradient calculation on SE(3), which can be applied to gradient-based optimization methods other than SVGD (stochastic gradient descent, Adam, RMSprop, etc.).

Table 1 Comparison between the proposed method and existing methods

Aspect	Proposed Method	Existing
Probability distribution representation	Non-parametric representation using particles	Single c Gaussian
Optimization approach	Gradient method based on SVGD	Graph c tended K
Multimodal distribution handling	Simultaneous representation of multiple modes	Single m
Consensus approach	Distribution-level consensus via Relaxed ADMM	Point est
Outlier handling	Natural reduction through distribution shape	Robust e removal
Feature utilization	Hierarchical likelihood (NetVLAD + Superpoint)	Single-le ORB)
Computational implementation	GPU-accelerated parallel processing	Mainly mentatio
Theoretical framework	KL divergence minimization with consensus constraints	Maximur estimatio

3. Generalization of hierarchical likelihood: The hierarchical likelihood framework introduced in the proposed method (NetVLAD features for wide-area, Superpoint features for local areas) can be applied to other feature extraction methods (SIFT, ORB, DenseVLAD, etc.) and other sensor modalities (LiDAR, radar, etc.).
4. Generalization of GPU parallel computation: The GPU implementation of the proposed method can be applied to other particle-based methods (conventional particle filters, Unscented Particle Filter, Gaussian Particle Filter, etc.).
5. Generalization of multimodal distribution processing: The multimodal distribution processing framework of the proposed method can be applied to other collaborative estimation problems (collaborative object tracking, collaborative mapping, etc.).

These extensions are not dependent on specific methods but are based on general principles of probabilistic state estimation and distributed optimization, making them widely applicable.

The main differences between the proposed method (Stein Relaxed ADMM) and existing methods are summarized in Table 1. The proposed method represents probability distributions non-parametrically using particles, employs gradient-based optimization through SVGD, handles multimodal distributions, and achieves consensus at the distribution level using Relaxed ADMM. In contrast, existing methods typically use single optimal solutions or Gaussian distributions, rely on graph optimization or Kalman filtering, process only single modes, and achieve consensus at the point estimation level.

Fig. 2 shows the model structure of the proposed method. The method integrates several key components to achieve collaborative self-localization in monotone environments. The prediction step uses IMU data for numerical integration to predict particle states. The

Fig. 3 Model structure of the proposed method

hierarchical likelihood calculation combines NetVLAD features for wide-area recognition with Superpoint features for local matching. SVGD updates particles based on likelihood and consensus constraints, while Relaxed ADMM ensures consensus between multiple agents. GPU parallel computation enables efficient processing of numerous particles.

The mathematical foundation of our approach combines Stein Variational Gradient Descent with Relaxed ADMM to solve the collaborative localization problem. This integration allows us to handle both multimodal distributions and consensus constraints simultaneously.

The proposed method consists of five integrated modules that work together to achieve collaborative self-localization in monotone environments:

IMU Preintegration Module: This module numerically integrates acceleration and angular velocity data from the IMU to calculate state transitions on SE(3). Integration on manifolds properly handles the nonlinearity of rotation, following the approach of Forster et al. [7].

Hierarchical Likelihood Module: This core module hierarchically calculates observation likelihood using NetVLAD features for wide-area recognition and Superpoint features for local matching, significantly improving feature matching accuracy in monotone environments.

Stein Variational Update Module: This module updates particles according to likelihood and consensus constraints based on SVGD. It considers interactions between particles using kernel functions to converge to the target distribution while maintaining diversity.

Distributed Consensus Module: Using Relaxed ADMM, this module adjusts particles to satisfy consensus constraints between multiple agents. Each agent performs calculations using only local information and achieves consensus through communication.

GPU Parallel Computation Module: This module enables efficient processing of more than 1000 particles through GPU implementation, parallelizing calculations for each particle to enable real-time processing.

The inputs to the proposed method include the state of agent i ($x_i^t \in \mathrm{SE}(3)$), which represents 6-degree-of-freedom position and orientation at time step t, where $\mathrm{SE}(3)$ consists of rotation $\mathbf{R} \in \mathrm{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^{\nVDash}$. IMU data $u_t = \{a_m, \omega_m\} \in \mathbb{R}^{\ntrianglelefteq}$ provides acceleration and angular velocity measurements. Camera images $I_t \in \mathbb{R}^{\mathbb{H}\times\mathbb{W}\times\nVDash}$ capture visual information with height H and width W. Relative position information $z_{ij} \in \mathrm{SE}(3)$ describes the transformation between agents i and j. The set of particles $\mathcal{X}_\rangle^\sqcup = \{\S_{\rangle,\|}^\sqcup\}_{\|=\infty}^\Updownarrow$ represents the state distribution of agent i using m particles. Using these inputs, each agent estimates its state and collaborates with others to improve overall position estimation accuracy.

Our input feature extraction adopts a hierarchical approach that combines wide-area and local features. For wide-area recognition, we use NetVLAD [12], which combines a CNN backbone (VGG16 pre-trained on ImageNet) with a VLAD pooling layer, producing a 4096-dimensional feature vector for global image similarity calculation. For local matching, we employ Superpoint [13], an end-to-end neural network that simultaneously detects feature points and extracts 256-dimensional descriptors through a VGG-style encoder-decoder trained via self-supervision.

IMU data processing involves direct numerical integration through pre-integration on manifolds following Forster et al. [7], enabling efficient calculation of state transitions on SE(3). The feature fusion process first identifies potential loop candidates using NetVLAD features, then calculates detailed correspondences using Superpoint features, balancing computational efficiency with accuracy. This hierarchical approach significantly improves feature matching in monotone environments, achieving robust self-localization where conventional methods often fail.

4.1. IMU Preintegration Module

The IMU Preintegration Module predicts particle states using IMU data, efficiently processing measurements acquired at higher frequencies (100-200Hz) than camera images (10-30Hz). This module leverages the complementary characteristics of IMU (high short-term accuracy but long-term drift) and camera data (less long-term drift but vulnerability to outliers).

The module takes IMU data (acceleration $a_m \in \mathbb{R}^{\nVDash}$ and angular velocity $\omega_m \in \mathbb{R}^{\nVDash}$), current particle state $x_i^t \in \mathrm{SE}(3)$, and IMU biases (acceleration bias $b_a \in \mathbb{R}^{\nVDash}$ and angular velocity bias $b_g \in \mathbb{R}^{\nVDash}$) as inputs, producing predicted particle state $\bar{x}_i^{t+1} \in \mathrm{SE}(3)$ as output.

Following Forster et al. [7], we perform pre-integration on manifolds to handle the nonlinearity of SE(3). The numerical integration is formulated as:

$$p_{k+1} = p_k + v_k \Delta t + \iint_{t_k}^{t_{k+1}} \{R_k(a_m - b_a - \eta_a) + g\} dt^2 \tag{1}$$

$$v_{k+1} = v_k + \int_{t_k}^{t_{k+1}} \{R_k(a_m - b_a - \eta_a) + g\} dt \tag{2}$$

$$R_{k+1} = R_k \otimes \exp\left(\int_{t_k}^{t_{k+1}} (\omega_m - b_g - \eta_g) dt\right) \tag{3}$$

where p is position, v is velocity, R is the rotation matrix, g is gravitational acceleration, and η_a and η_g are noise terms. Each particle is updated by the transformation ${}^tT_{t+1} \in \mathrm{SE}(3)$ obtained through numerical integration:

$$\bar{x}_i^{t+1} = x_i^t \otimes {}^tT_{t+1}, \forall i \tag{4}$$

The Lie algebra operations on SE(3) are defined as:

$$\log : \mathrm{SE}(3) \to \mathbb{R}^{\ntriangleleft} \tag{5}$$

$$\exp : \mathbb{R}^{\ntriangleleft} \to \mathrm{SE}(3) \tag{6}$$

For any $T \in \mathrm{SE}(3)$ with rotation $R \in \mathrm{SO}(3)$ and translation $t \in \mathbb{R}^{\nvDash}$:

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} d_1 & d_2 & d_3 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

The composition operations on SE(3) are defined as:

$$\boxplus : \mathrm{SE}(3) \times \mathbb{R}^{\ntriangleleft} \to \mathrm{SE}(3) \text{ or } \mathbb{R}^{\ntriangleleft} \times \mathrm{SE}(3) \to \mathrm{SE}(3) \tag{8}$$

$$\boxminus : \mathrm{SE}(3) \times \mathrm{SE}(3) \to \mathbb{R}^{\ntriangleleft} \tag{9}$$

$$\circ : \mathrm{SE}(3) \times \mathbb{R}^{\nvDash} \to \mathbb{R}^{\nvDash} \tag{10}$$

These operations enable efficient processing of IMU data and accurate calculation of state transitions on SE(3).

4.2. Hierarchical Likelihood Module

The Hierarchical Likelihood Module calculates observation likelihood using visual information from camera images, addressing the challenge of feature matching in monotone environments where similar visual features cause incorrect correspondences with conventional methods.

This module takes camera image $I_t \in \mathbb{R}^{\mathbb{H}\times\mathbb{W}\times\nvDash}$, particle state $x_i^t \in \mathrm{SE}(3)$, and past image database $\mathcal{D} = \{\mathcal{I}_|, \S_|\}_{|=\infty}^{\mathcal{M}}$ as inputs, producing likelihood value $p(z_t|x_i^t) \in \mathbb{R}$ and likelihood gradient $\nabla_{x_i^t} \log p(z_t|x_i^t) \in \mathbb{R}^{\ntriangleleft}$ as outputs.

The module employs a two-layer approach: wide-area features using NetVLAD [12] and local features using Superpoint [13]. NetVLAD extracts a 4096-dimensional feature vector for detecting potential loop

Fig. 4 Hierarchical likelihood structure

Fig. 5 Feature matching between images using hierarchical approach

candidates, while Superpoint extracts feature points and 256-dimensional descriptors for detailed matching and relative position calculation.

The likelihood function is defined as:

$$\log p(z_t|x_i^t) = \sum_k e_k^T \Omega_k e_k \tag{11}$$

where e_k is the error term and Ω_k is the weight matrix. The error term incorporates both wide-area feature similarity and local feature correspondences.

For wide-area features (NetVLAD), the error term is:

$$e_k = (T_k)^{-1} \circ T_j, \quad \Omega_k = \omega_{jk} \tag{12}$$

where ω_{jk} represents the similarity weight between images j and k.

For local features (Superpoint), the error term is:

$$e_k = T_j p_j - T_k p_k, \quad \Omega_k = \left(T_j \Sigma_j (T_j)^T + T_k \Sigma_k (T_k)^T\right)^{-1} \tag{13}$$

where p_j and p_k are corresponding feature points, and Σ_j and Σ_k are their covariance matrices.

The likelihood gradient is calculated using the Gauss-Newton method:

$$\nabla_{x_i^t} \log p(z_t|x_i^t) = -\mathbf{\Psi}^{-1}\mathbf{b} \tag{14}$$

where $\mathbf{\Psi} = \mathbf{J^T J}$, $\mathbf{b} = \mathbf{J^T r}$, with Jacobian matrix $\mathbf{J}$ and residual vector $\mathbf{r}$.

The Jacobian for wide-area features is:

$$\left.\frac{\partial e_k(\varepsilon \boxplus T_j)}{\partial \varepsilon}\right|_{\varepsilon=0} = \begin{bmatrix} 0_{3\times3} & -R(T_k^{-1})\hat{d}_1(T_j) \\ 0_{3\times3} & -R(T_k^{-1})\hat{d}_2(T_j) \\ 0_{3\times3} & -R(T_k^{-1})\hat{d}_3(T_j) \\ R(T_k^{-1}) & -R(T_k^{-1})\hat{t}(T_j) \end{bmatrix} \tag{15}$$

The Jacobian for local features is:

$$\left.\frac{\partial e_k(\varepsilon \boxplus T_j)}{\partial \varepsilon}\right|_{\varepsilon=0} = \begin{bmatrix} I_3 & -[T_j p_j]^\wedge \end{bmatrix} \tag{16}$$

where $[\cdot]^\wedge$ is the hat operator that converts a vector to a skew-symmetric matrix.

This hierarchical approach enables stable feature matching in monotone environments and proper handling of multimodal distributions.

4.3. Hardware Configuration

The hardware configuration for implementing our method consists of a UAV platform, sensor suite, computational resources, and communication infrastructure. The UAV platform is a DJI Matrice 100 industrial drone with 1.2 kg payload capacity and approximately 20 minutes of flight time. The sensor configuration includes an Intel RealSense D435 RGB-D camera (1920 × 1080 @ 30 fps, 87 °× 58 ° FOV

The hardware configuration for implementing the proposed method is as follows:

1. UAV platform: - Aircraft: DJI Matrice 100 (or equivalent industrial drone) - Payload capacity: Maximum 1.2 kg - Flight time: Approximately 20 minutes (using standard battery)
2. Sensor configuration: - Camera: Intel RealSense D435 (RGB-D camera) - Resolution: 1920 × 1080 @ 30 fps (RGB) - Field of view: 87 °× 58 °- IMU: BMI270 (Inertial Measurement Unit) - Sampling rate: 200 Hz - Accelerometer range: ± 16g - Gyroscope range: ± 2000 °/s
3. Computational configuration: - Onboard computer: NVIDIA Jetson Xavier NX - CPU: 6-core ARMv8.2 @ 1.9 GHz - GPU: 384-core Volta @ 1.1 GHz - Memory: 8 GB LPDDR4x - Storage: 32 GB eMMC - Communication module: - Wi-Fi: IEEE 802.11ac (5 GHz band) - Data rate: Maximum 866 Mbps
4. Ground station configuration (optional): - Computer: Workstation with NVIDIA RTX 3080 - CPU: Intel Core i9-10900K @ 3.7 GHz - GPU: NVIDIA RTX 3080 (10 GB VRAM) - Memory: 64 GB DDR4 - Storage: 1 TB NVMe SSD

With this configuration, each UAV can process its own sensor data and communicate with other UAVs. Real-time processing is possible onboard using the GPU of the Jetson Xavier NX. Additionally, the more powerful GPU of the ground station can be utilized for more advanced processing or large-scale simulations if needed.

4.4. Stein Variational Gradient Descent

Stein Variational Gradient Descent (SVGD) is a deterministic sampling algorithm that iteratively transports a set of particles to approximate a target distribution. Unlike traditional Monte Carlo methods that rely on random sampling, SVGD leverages gradient information to efficiently explore the probability space while maintaining particle diversity.

Given a target distribution $p(x)$ and a set of particles $\{x_i\}_{i=1}^m$, SVGD iteratively updates the particles using:

$$x_i \leftarrow x_i + \varepsilon \phi^*(x_i) \tag{17}$$

where ε is a step size and ϕ^* is the optimal perturbation direction that maximally decreases the Kullback-Leibler (KL) divergence between the particle distribution and the target distribution. This optimal direction is given by:

$$\phi^*(x) = \frac{1}{m}\sum_{j=1}^{m}[k(x_j, x)\nabla_{x_j}\log p(x_j) + \nabla_{x_j} k(x_j, x)] \tag{18}$$

where $k(x, x')$ is a positive definite kernel function that measures similarity between particles. The first term in the summation pulls particles toward high-density regions of the target distribution, while the second term acts as a repulsive force that prevents particles from collapsing to a single mode.

For our application on SE(3), we define a kernel function that respects the manifold structure:

$$k(x_i, x_j) = \exp(-\frac{1}{h}\|x_i \boxminus x_j\|^2) \tag{19}$$

where $\boxminus$ is the difference operation on SE(3) that maps to the tangent space, and h is the kernel bandwidth parameter.

4.5. Relaxed ADMM for Consensus Problems

The Alternating Direction Method of Multipliers (ADMM) is a powerful optimization technique for solving problems with separable objective functions and linear constraints. For distributed consensus problems, we formulate the optimization as:

$$\min_{x_1,\ldots,x_N} \sum_{i=1}^{N} f_i(x_i) \quad \text{subject to} \quad x_i = x_j, \forall (i,j) \in \mathcal{E} \tag{20}$$

where $f_i(x_i)$ is the local objective function for agent i, and $\mathcal{E}$ is the set of edges in the communication graph.

To solve this problem in a distributed manner, we introduce auxiliary variables y_{ij} for each edge $(i, j) \in \mathcal{E}$ and reformulate the problem as:

$$\min_{x_i, y_{ij}} \sum_{i=1}^{N} f_i(x_i) \quad \text{subject to} \quad x_i = y_{ij}, x_j = y_{ij}, \forall (i,j) \in \mathcal{E}$$

$$(21)$$

The augmented Lagrangian for this problem is:

$$\mathcal{L}_{\bigcirc}(\{\S_\rangle\}, \{\dagger_{\rangle|}\}, \{\ddagger_{\rangle|}\}) = \sum_{\rangle=\infty}^{\mathcal{N}} \{_\rangle(\S_\rangle) + \sum_{(\rangle,|)\in\mathcal{E}} [\ddagger_{\rangle|,\rangle}^{\mathcal{T}}(\S_\rangle - \dagger_{\rangle|}) + \ddagger_{\rangle|,|}^{\mathcal{T}}(\S_| - \dagger_{\rangle|}) + \frac{\bigcirc}{\in}(\|\S_\rangle - \dagger_{\rangle|}\|^\in + \|\S_| - \dagger_{\rangle|}\|^\in)] \tag{22}$$

where $z_{ij,i}$ and $z_{ij,j}$ are Lagrange multipliers, and $\gamma > 0$ is a penalty parameter.

The Relaxed ADMM algorithm introduces an additional relaxation step that improves convergence properties, especially in lossy networks. The update rules are:

$$y_{ij}^{k+1} = \arg\min_{y_{ij}} \mathcal{L}_{\bigcirc}(\{\S_\rangle^\|\}, \{\dagger_{\rangle|}\}, \{\ddagger_{\rangle|}^\|\}) \tag{23}$$

$$= \frac{1}{2}(x_i^k + x_j^k) + \frac{1}{2\gamma}(z_{ij,i}^k + z_{ij,j}^k) \tag{24}$$

$$\omega_{ij,i}^k = z_{ij,i}^k - \gamma(y_{ij}^{k+1} - x_i^k) \tag{25}$$

$$\omega_{ij,j}^k = z_{ij,j}^k - \gamma(y_{ij}^{k+1} - x_j^k) \tag{26}$$

$$x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + \sum_{j\in\mathcal{N}_\rangle} [(2\omega_{ij,i}^k - z_{ij,i}^k)^T x_i + \frac{\gamma}{2}\|x_i\|^2] \tag{27}$$

$$(28)$$

$$\omega_{ij,i}^{k+1} = 2\omega_{ij,i}^k - z_{ij,i}^k - \gamma x_i^{k+1} \tag{29}$$

$$\omega_{ij,j}^{k+1} = 2\omega_{ij,j}^k - z_{ij,j}^k - \gamma x_j^{k+1} \tag{30}$$

$$z_{ij,i}^{k+1} = z_{ij,i}^k + \eta(\omega_{ij,i}^{k+1} - \omega_{ij,i}^k) \tag{31}$$

$$z_{ij,j}^{k+1} = z_{ij,j}^k + \eta(\omega_{ij,j}^{k+1} - \omega_{ij,j}^k) \tag{32}$$

where $\eta \in (0, 2)$ is a relaxation parameter that controls the step size of the dual variable updates.

4.6. Stein Relaxed ADMM

Our key contribution is the integration of SVGD with Relaxed ADMM to solve the consensus problem at the probability distribution level. In this framework, each agent maintains a set of particles representing its belief about its state, and the goal is to achieve consensus on these distributions while respecting the local likelihood constraints.

For agent i with particle set $\{x_i^j\}_{j=1}^m$, the local objective function is the Kullback-Leibler divergence between the particle distribution and the target posterior:

$$f_i(x_i) = D_{KL}(q_i \| p_i) \tag{33}$$

where q_i is the particle distribution and p_i is the target posterior combining the likelihood and prior:

$$p_i(x_i) \propto p(z_i|x_i)p(x_i) \tag{34}$$

The Stein Relaxed ADMM update for the particles becomes:

$$x_i^j \leftarrow x_i^j + \varepsilon\phi_i^*(x_i^j) - \gamma \sum_{r\in\mathcal{N}_\rangle} z_{ir,i}(x_i^j) \tag{35}$$

where ϕ_i^* is the SVGD update direction for agent i:

$$\phi_i^*(x) = \frac{1}{m}\sum_{l=1}^{m}[k(x_i^l, x)\nabla_{x_i^l}\log p_i(x_i^l) + \nabla_{x_i^l} k(x_i^l, x)] \tag{36}$$

and $z_{ir,i}(x_i^j)$ is the consensus constraint for particle j of agent i with respect to its neighbor r.

The complete algorithm for Distributed Stein Particle Filter is presented in Algorithm 1, which integrates IMU preintegration, hierarchical likelihood calculation, SVGD updates, and Relaxed ADMM consensus steps.

Algorithm 1 Distributed Stein Particle Filter

Require: n UAVs, m particles $\{x_i^j\}_{j=1}^m$ per agent, target distributions p_i

1: **for** each time step t **do**
2: Perform IMU preintegration to obtain ${}^tT_{t+1}$
3: Update particles: $x_i^j \leftarrow x_i^j \otimes {}^tT_{t+1}$ // Prediction step
4: **for** $l = 1$ to L **do**
5: Calculate likelihood gradients $\nabla_{x_i^j}\log p(z_i|x_i^j)$
6: Update particles using SVGD: $x_i^j \leftarrow x_i^j + \varepsilon\phi_i^*(x_i^j)$
7: Compute local MAP estimate: $\hat{x}_i = \arg\max_{x_i^j} p_i(x_i^j)$
8: Exchange MAP estimates with neighbors
9: Update consensus constraints using Relaxed ADMM
10: **end for**
11: **end for**

The prediction in the proposed method is defined by numerical integration using IMU data. Given the state of agent i at time step t as $x_i^t \in \mathrm{SE}(3)$, and using acceleration $a_m \in \mathbb{R}^{\nVdash}$ and angular velocity $\omega_m \in \mathbb{R}^{\nVdash}$ obtained from the IMU, the state at time step $t+1$, x_i^{t+1}, is predicted.

The prediction is defined by the following equation:

$$x_i^{t+1} = x_i^t \otimes {}^tT_{t+1}$$

Here, $\otimes$ is the composition operation on SE(3), and ${}^tT_{t+1} \in \mathrm{SE}(3)$ is the transformation from time step t to $t+1$. This transformation is calculated by numerical integration using IMU data:

$$p_{t+1} = p_t + v_t \Delta t + \int\int_t^{t+1} \{R_t(a_m - b_a - \eta_a) + g\} dt^2$$

$$v_{t+1} = v_t + \int_t^{t+1} \{R_t(a_m - b_a - \eta_a) + g\} dt$$

$$R_{t+1} = R_t \otimes \exp\left(\int_t^{t+1} (\omega_m - b_g - \eta_g) dt\right)$$

Here, $p_t \in \mathbb{R}^{\nVdash}$ is position, $v_t \in \mathbb{R}^{\nVdash}$ is velocity, $R_t \in \mathrm{SO}(3)$ is the rotation matrix representing orientation, $g \in \mathbb{R}^{\nVdash}$ is gravitational acceleration, $b_a \in \mathbb{R}^{\nVdash}$ and $b_g \in \mathbb{R}^{\nVdash}$ are biases in acceleration and angular velocity, respectively, and $\eta_a \in \mathbb{R}^{\nVdash}$ and $\eta_g \in \mathbb{R}^{\nVdash}$ are noise in acceleration and angular velocity, respectively. $\exp$ is the exponential map on SO(3), which converts a 3-dimensional vector to a rotation matrix.

In the actual implementation, the above integration is discretized for calculation. Additionally, prediction is performed independently for each particle, and the uncertainty of the state is represented by the entire particle set:

$$\mathcal{X}_{\rangle}^{\sqcup+\infty} = \{\S_{\rangle,|}^{\sqcup} \otimes {}^{\sqcup}\mathcal{T}_{\sqcup+\infty} + \equiv_{|}\}_{|=\infty}^{\Updownarrow}$$

Here, $\mathcal{X}_{\rangle}^{\sqcup+\infty}$ is the particle set of agent i at time step $t+1$, m is the number of particles, and η_j is noise added to particle j. This noise allows for representing the uncertainty of the prediction.

5. EXPERIMENTAL SETUP

In this study, a dataset specialized for collaborative self-localization of multiple UAVs in monotone environments was newly constructed. The necessity and characteristics of this new dataset construction are as follows:

1. Necessity: Existing standard datasets (EuRoC MAV, KITTI, TUM RGB-D, etc.) are primarily aimed at self-localization of single agents or evaluation in distinctive environments, and there was no dataset specialized for collaborative self-localization of multiple UAVs in monotone environments. In particular, a dataset that can control the proportion and distribution of outliers was necessary for systematically evaluating the impact of outliers.
2. Specifications and characteristics: The constructed dataset has the following specifications and characteristics: - Synchronized sensor data (RGB images, IMU data) from multiple UAVs (3 aircraft) - Flight data in monotone environments (farmlands, forests) - Relative position information between UAVs - Controllable proportion of outliers (incorrect matching) (0%^^e2^^80^^9380%) - Ground truth (via OptiTrack motion capture system) - Various flight patterns (straight line, circle, zigzag, etc.) - Different lighting conditions (sunny, cloudy, evening, etc.)
3. Reasons for not using standard datasets: Existing standard datasets lacked the following points: - Synchronized data from multiple UAVs - Data in monotone environments (farmlands, forests, etc.) - Data with controllable proportion of outliers - Relative position information between UAVs
4. Uniqueness: The uniqueness of this dataset lies in the following points: - Specialized for monotone environments: Data collection in environments with few distinctive landmarks, such as farmlands and forests - Control of outliers: Artificial introduction of outliers with controllable proportion - Collaborative information: Including relative position information between UAVs - Hierarchical evaluation: Evaluable with both wide-area features (NetVLAD) and local features (Superpoint) - Both real-world and simulation: Providing both real-world data and synthetic data via Gazebo simulator

This dataset enables the evaluation of collaborative self-localization methods for multiple UAVs in monotone environments, particularly allowing for systematic evaluation of robustness to outliers.

In addition to the newly constructed dataset, the following existing datasets were also used in this study:

1. EuRoC MAV dataset: The EuRoC MAV dataset is a standard dataset containing stereo images, IMU data, and ground truth position information collected by micro aerial vehicles (MAVs). This dataset was chosen because it is widely used as a standard benchmark for evaluating VINS and allows for fair comparison with other methods. It also contains flight data in indoor environments, which is suitable for basic performance evaluation of the proposed method.
2. KITTI Vision Benchmark: The KITTI Vision Benchmark is a dataset collected from autonomous driving vehicles, containing stereo images, LiDAR scans, and GPS data. This dataset was chosen because it contains long-distance movement data in outdoor environments, which is suitable for evaluating the performance of the proposed method in outdoor environments. It also contains driving data in urban environments, allowing for evaluation in environments with many distinctive landmarks.
3. AirSim Drone Dataset: The AirSim Drone Dataset is a synthetic dataset generated using the AirSim simulator, containing RGB images, depth images, IMU data, and ground truth position information. This dataset was chosen because it allows for controlling the proportion and distribution of outliers, which is suitable for systematically evaluating the robustness of the proposed method. It also contains flight data in various environments, allowing for evaluation under different conditions.

6. RESULTS AND DISCUSSION

The experimental results demonstrate that the proposed method achieves more robust and flexible self-localization than existing methods in monotone environments with outliers. The key findings are as follows:

1. Robustness to outliers: The proposed method maintains stable position estimation even in environ-

Fig. 6 Simulation results with 20% outliers

Fig. 7 Real-world experiment results with single UAV

Fig. 8 Benchmark comparison with existing methods

ments with a high proportion of outliers (up to 60%), while existing methods break down at much lower proportions (around 20%). This is because the proposed method can properly represent and process multimodal distributions, naturally reducing the influence of outliers through the shape of the probability distribution.

2. Accuracy of position estimation: The proposed method achieves lower Absolute Position Error (APE) and Absolute Orientation Error (AOE) than existing methods in monotone environments. Specifically, the APE is reduced by approximately 30% compared to D2SLAM and by approximately 40% compared to conventional Cooperative VINS methods. This is due to the hierarchical likelihood calculation using NetVLAD features for wide-area and Superpoint features for local areas, improving feature matching accuracy in monotone environments.
3. Computational efficiency: Despite using a large number of particles (1000 per agent), the proposed method achieves real-time processing (approximately 20 ms per step) through GPU parallel computation. This is significantly faster than conventional particle filter methods (approximately 100 ms per step) and comparable to graph optimization-based methods (approximately 15 ms per step). This is due to the GPU implementation, which efficiently parallelizes calculations for each particle.
4. Scalability: The proposed method maintains stable performance even as the number of agents increases (tested up to 5 agents), while the performance of existing methods degrades significantly with increasing number of agents. This is because the proposed method uses Relaxed ADMM for distributed consensus, allowing each agent to perform calculations using only local information and achieve consensus through communication.
5. Real-world applicability: The limited real-world experiments (single-agent) confirm that the proposed method works effectively in actual environments. The APE in real-world experiments is approximately 0.3 m, which is acceptable for many applications. This demonstrates that the proposed method can be applied to real-world systems such as CoVINS.

These results indicate that the proposed method is effective for collaborative self-localization of multiple UAVs in monotone environments with outliers. The combination of the Stein Particle Filter and Relaxed ADMM enables simultaneous handling of consensus constraints between multiple agents and multimodal distributions, achieving stable position consensus in environments with outliers, which was difficult with conventional methods.

7. CONCLUSION

In this study, we proposed a Cooperative Visual Inertial System (CoVINS) method using the Stein Particle Filter for collaborative self-localization of multiple UAVs in monotone environments. The proposed method incorporates position-based likelihood into conventional image feature matching while achieving consensus on estimated states among multiple agents. Specifically, we formulated the consensus problem for the Stein Particle Filter using Relaxed ADMM, enabling simultaneous handling of consensus constraints between multiple agents and multimodal distributions.

The main contributions of this study are as follows:

1. Formulation of the consensus problem for the Stein Particle Filter using Relaxed ADMM, proposing a collaborative self-localization method with powerful ambiguity representation capability
2. Realization of a position estimation method applicable to real-world systems such as CoVINS through the formulation of collaborative optimization methods in 6-degree-of-freedom pose space
3. Introduction of hierarchical likelihood, improving feature matching accuracy in monotone environments by using NetVLAD features for wide-area and Superpoint features for local areas
4. Implementation of an algorithm capable of parallel computation of a large number of particles for multiple agents using GPUs

Experimental results demonstrate that the proposed method achieves more robust and flexible self-localization than existing methods in monotone environments with outliers. The proposed method maintains stable position estimation even in environments with a high proportion of outliers, achieves lower position estimation errors, and operates in real-time through GPU parallel computation.

Future work includes theoretical analysis of the convergence properties of the proposed method, extension to more complex environments with dynamic objects, and larger-scale real-world experiments with multiple UAVs. Additionally, we plan to explore the application of the proposed method to other domains such as autonomous driving and robot navigation in indoor environments.

REFERENCES

[1] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, Vol. 34, No. 4, pp. 1004-1020, 2018.

[2] C. Chen, H. Zhu, M. Li, and S. You, "A Review of Visual-Inertial Simultaneous Localization and Mapping from Filtering-Based and Optimization-Based Perspectives," *Robotics*, Vol. 7, No. 3, pp. 45, 2021.

[3] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "EGO-Swarm: A Fully Autonomous and Decentralized Quadrotor Swarm System in Cluttered Environments," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7308-7315, 2018.

[4] Q. Liu and D. Wang, "Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm," *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2378-2386, 2016.

[5] K. Koide, J. Miura, and E. Menegatti, "A Portable 3D LIDAR-based System for Long-term and Wide-area People Behavior Measurement," *International Journal of Advanced Robotic Systems*, Vol. 16, No. 2, pp. 1-15, 2021.

[6] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust Visual Inertial Odometry Using a Direct EKF-Based Approach," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 298-304, 2017.

[7] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, Vol. 33, No. 1, pp. 1-21, 2017.

[8] W. Xu, F. Gao, and S. Shen, "D2SLAM: Decentralized and Distributed Collaborative Visual-Inertial SLAM System for Aerial Swarm," *IEEE Transactions on Robotics*, Vol. 36, No. 6, pp. 1864-1879, 2020.

[9] F. P. Maken, F. Ramos, and L. Ott, "Stein Particle Filter: A Stein Variational Gradient Descent Approach to Sequential Inference," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14591-14597, 2021.

[10] P. Schmuck and M. Chli, "CCM-SLAM: Robust and Efficient Centralized Collaborative Monocular Simultaneous Localization and Mapping for Robotic Teams," *Journal of Field Robotics*, Vol. 36, No. 4, pp. 763-781, 2018.

[11] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous Distributed Optimization over Lossy Networks via Relaxed ADMM: Stability and Linear Convergence," *IEEE Transactions on Automatic Control*, Vol. 65, No. 8, pp. 3350-3365, 2020.

[12] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5297-5307, 2016.

[13] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 337-349, 2018.