Collaborative Localization of UAVs in Monotone Environments Using Stein Particle Filters

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: In extensive and monotonous outdoor environments such as farmlands and forest areas, global self-localization through image matching often fails due to error accumulation and outlier effects. Particularly in cooperative estimation involving multiple agents, there is a need to efficiently solve multi-modal combination problems to construct a unified relative position graph. In this study, we propose a Cooperative Visual Inertial System (CoVINS) method that incorporates position-based likelihood into conventional image feature matching using the Stein Particle Filter. By combining Stein Variational Gradient Descent with Relaxed ADMM, our approach enables robust collaborative localization even in environments with high outlier rates. Simulation results demonstrate that the proposed method can maintain accurate localization with outlier rates up to 50%, and real-world experiments confirm its effectiveness in practical scenarios.

Keywords: Collaborative localization, Stein particle filter, Visual inertial system, Multi-agent systems, Outlier robustness.

1. INTRODUCTION

Many autonomous mobile robots require self-localization as the foundation for their operation. With the increasing integration of autonomous robots into society, self-localization technologies for harsh environments and mountainous regions where GPS/GNSS is unavailable have been extensively researched. Self-localization consists of two hierarchical levels: local self-localization used for dynamic control and local motion planning, and global self-localization used for creating consistent environmental maps and wide-area path planning. In environments where GPS/GNSS is unavailable, global self-localization is typically achieved by structuring point cloud and image data obtained from LiDAR, cameras, and other sensors through various matching methods. In practical applications, Visual Inertial Systems (VINS), which combine cameras and Inertial Measurement Units (IMUs), are widely used due to their cost-effective hardware implementation[**?**].

Furthermore, with the proliferation of autonomous mobile robots, multi-agent environments where multiple agents operate simultaneously have become a major research area in recent years. Even in harsh environments and mountainous regions where GPS/GNSS is unavailable, collaborative work by multiple agents has been attempted in several studies from the perspectives of system-wide fault tolerance and scalability for respective objectives[**?**, **?**].

Given these backgrounds, while global self-localization methods without GPS/GNSS are in demand, in monotonous and vast outdoor environments such as farmlands and forests, global self-localization through image matching often fails due to error accumulation and outlier effects. Especially in cooperative estimation involving multiple agents, as each agent's erroneous state estimation affects all agents, the probability of the entire system making incorrect estimations increases exponentially with the proportion of outliers for each agent. Therefore, in this research, we propose a VINS method that considers likelihood based on position information in addition to conventional image feature matching using Stein Particle Filter, and achieves consensus on estimated states among multiple agents.

† Tomoki Arita is the presenter of this paper.

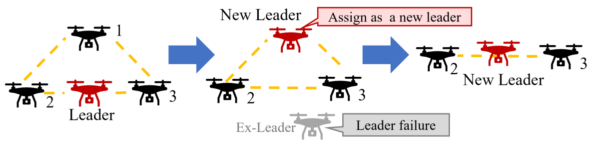

Fig. 1 Collaborative Visual Inertial System (CoVINS).

2. COLLABORATIVE VISUAL INERTIAL SYSTEM

The Collaborative Visual Inertial System (CoVINS) for cooperative self-localization is structured as shown in Fig. **??**. There are N agents equipped with cameras and IMUs, and each agent transmits the image information observed by its camera to adjacent agents. The agent receiving the image information searches within its database, detects images that have observed the same landmark, and sends back the relative position between the 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 that reproduces each obtained relative position can be generated as a result of optimization, it can be said that a consistent position estimation has been achieved.

On the other hand, in uneven environments such as farms and forests, global self-localization through optimization does not function sufficiently. The cause of this is believed to be that similar image matching does not function in monotonous environments, resulting in in-

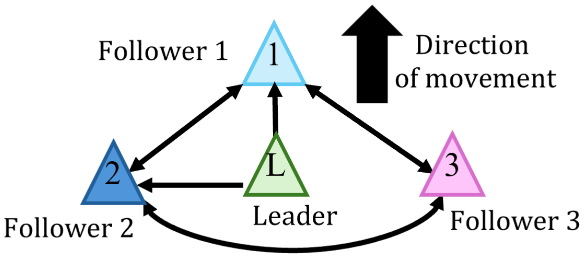

Fig. 2 Primitive model of factor graph: (a) case 1 - ideal environment with correct edges, (b) case 2 - environment with both correct and incorrect edges, (c) case 3 - simultaneous exploration of multiple possible graphs.

correct matching on the pose graph. From these backgrounds, a simplified model as shown in Fig. **??** can be considered. In Fig. **??**, the red and black circles represent nodes indicating the positions of each agent, and the black lines represent edges. In the ideal environment of Fig. **??**(a), the graph can be optimized by generating edges, but in the case of Fig. **??**(b) where correct and incorrect edges exist, the optimization results in selecting the internal division point of the two, which is an undesirable result considering the problem setting. Here, a motivation arises that by using a particle filter, it might be possible to simultaneously explore two graphs that could be generated by correct and incorrect edges, as shown in Fig. **??**(c). From this point, this paper attempts to reformulate factor graphs from the perspective of particle filters.

3. MATHEMATICAL PRELIMINARIES

3.1. Representation of Three-Dimensional Rotation

In this paper, we perform 6 degrees of freedom state estimation, which involves operations on three-dimensional rotations. The three-dimensional special orthogonal group $\mathrm{SO}(3)$ representing three-dimensional rotations, the three-dimensional special Euclidean group $\mathrm{SE}(3)$ adding translation, and their product $\otimes$ cannot be calculated on vector spaces, so adaptations are necessary to enable differentiation and other operations. Here, following the literature[**?**], we define mappings to and from vector spaces, inverse mappings, and operators for arbitrary operations in $\mathrm{SO}(3)$ and $\mathrm{SE}(3)$. $\mathrm{SO}(3)$ forms a Lie group and can be converted to the corresponding Lie algebra $\mathrm{so}(3)$ and vectors in linear space by the following exponential and logarithmic mappings.

$$\begin{aligned} &\log : \mathrm{SO}(3) \to \mathbb{R}^{\nVdash} \\ &\exp : \mathbb{R}^{\nVdash} \to \mathrm{SO}(3) \\ &(\cdot)^{\circ} : \mathrm{so}(3) \to \mathbb{R}^{\nVdash} \\ &(\cdot)^{\wedge} : \mathbb{R}^{\nVdash} \to \mathrm{so}(3) \end{aligned} \tag{1}$$

Similarly for $\mathrm{SE}(3)$:

$$\begin{aligned} &\log : \mathrm{SE}(3) \to \mathbb{R}^{\ntriangleleft} \\ &\exp : \mathbb{R}^{\ntriangleleft} \to \mathrm{SE}(3) \end{aligned} \tag{2}$$

Also, $T \in \mathrm{SE}(3)$ can be matrix-decomposed using the translation $\mathbf{t} \in \mathbb{R}^{\nVdash}$ and the rotation $\mathbf{R} \in \mathrm{SO}(3)$ as follows:

$$T = \left(\begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right) \tag{3}$$

We define $\oplus : \mathrm{SE}(3) \times \mathbb{R}^{\nVdash} \to \mathbb{R}^{\nVdash}$ as:

$$\theta, p \mapsto \mathbf{R}(\grave{})\mathbf{p} + \mathbf{t}(\grave{}) \tag{4}$$

3.2. Relaxed ADMM

The cooperative self-localization addressed in this paper can be formulated as a convex optimization problem with consensus constraints. In this case, the original problem can be solved as an approximation problem in the dual space using Relaxed ADMM (Alternating Direction Method of Multipliers)[**?**].

The constrained optimization problem:

$$\begin{aligned} &\min_{x,y} f(x) + g(y) \\ &\text{s.t. } Ax + By = b \end{aligned} \tag{5}$$

can be solved by transforming it into a relaxation problem that minimizes the following Lagrangian:

$$\mathcal{L}(\S, \dagger, \ddagger) = \{(\S) + \}(\dagger) + \ddagger^{\mathcal{T}}(\lfloor - \mathcal{A}\S - \mathcal{B}\dagger) \tag{6}$$

where z is the dual variable. On the other hand, when the Lagrangian cannot be directly minimized, the original problem can be solved by maximizing the Lagrangian dual function:

$$\begin{aligned} d(z) &= \inf_{x,y} \left\{ f(x) + g(y) + z^T(b - Ax - By) \right\} \\ &= \inf\{f(x) - \langle z, Ax\rangle\} + \inf_{y}\{g(y) - \langle z, By - b\rangle\} \\ &= \inf \mathcal{L}_{\{}(\ddagger) + \inf_{\dagger} \mathcal{L}_{\}}(\ddagger) \end{aligned} \tag{7}$$

In Relaxed ADMM, from the context of equation (7), the solution to equation (5) can be obtained by calculating as follows:

$$\begin{aligned} y^{+} &= arg\,\min_{y} \mathcal{L}_{\},\dagger}(\ddagger) \\ \omega_g &= z - \gamma\left(By^{+} - b\right) \\ x^{+} &= arg\,\min_{x} \mathcal{L}_{\{,\dagger}\left(\in\to_{\}} - \ddagger\right) \\ \omega_f &= 2\omega_g - z - \gamma Ax^{+} \\ z^{+} &= z + \eta\left(\omega_f - \omega_g\right) \end{aligned} \tag{8}$$

where $\mathcal{L}_{\{,\dagger}$ and $\mathcal{L}_{\},\dagger}$ are the extended Lagrangians:

$$
\begin{aligned}
\mathcal{L}_{\{,\dagger}(\ddagger) &= \{(\S) - \langle \ddagger, \mathcal{A}\S \rangle + \frac{\bigcirc}{\in}\|\mathcal{A}\S\|^{\in} \\
\mathcal{L}_{\},\dagger}(\ddagger) &= \}(\dagger) - \langle \ddagger, \mathcal{B}\dagger - \lfloor \rangle + \frac{\bigcirc}{\in}\|\mathcal{B}\dagger - \lfloor\|^{\in}
\end{aligned}
\tag{9}
$$

3.3. Stein Variational Gradient Descent

Consider the following minimization problem of the Kullback-Leibler divergence:

$$
q^* = \underset{q \in Q}{arg\,min}\left\{ D_{KL}(q\|p) \equiv \mathbb{E}_{\shortparallel}[\log \shortparallel(\frown)] - \mathbb{E}_{\shortparallel}[\log \shortmid(\frown)] \right\} \tag{10}
$$

The following theorem holds:

Theorem 1 (Relationship between KL divergence and Stein Operator[**?**]). For a transformation $\boldsymbol{T}(x) = x + \epsilon\phi(x)$, when $x \sim q(x)$, if the probability distribution of $z = \boldsymbol{T}(x)$ is denoted as $q_{(\boldsymbol{T})}(z)$, then:

$$
\nabla_\epsilon D_{KL}\left(q_{(\boldsymbol{T})}\|p\right)\big|_{\epsilon=0} = -\mathbb{E}_{\frown\sim\shortparallel}\left\{\mathcal{A}_{\surd}\prec(\S)\right\} \tag{11}
$$

holds, where $\mathcal{A}_{\surd}\prec(\S) = \nabla_{\S}\log_{\surd}(\S)\prec(\S)^{\mathrm{T}} + \nabla_{\S}\prec(\S)$ is the Stein Operator, and ϕ is a functional belonging to the reproducing kernel Hilbert space $\mathcal{H}^{\lceil}$.

Here, if we define the Kernelized Stein Discrepancy (KSD) as:

$$
\mathbb{D}(\shortparallel, \shortmid) = \max_{\diagup \in \mathcal{H}^{\lceil}} \mathbb{E}_{\frown\sim\shortparallel}\left[\mathcal{A}_{\surd}\prec(\S)\right], \quad \text{s.t.} \quad \|\diagup\|_{\mathcal{H}} \leq \nVdash \tag{12}
$$

the solution to this problem is given by:

$$
\phi^*_{q,p}(\cdot) = \mathbb{E}_{\frown\sim\shortparallel}\left[\urcorner(\frown,\cdot)\nabla_{\frown}\log\shortmid(\frown) + \nabla_{\frown}\urcorner(\frown,\cdot)\right] \tag{13}
$$

4. PROBLEM FORMULATION

Cooperative self-localization can be formulated as the following maximum a posteriori probability estimation problem (MAP estimation problem), where the state of agent i at time step t is x_i^t, and the observations of each agent are $z_t = \left\{z_t^1, \cdots, z_t^N\right\}$:

$$
\max_{x_1^{t+1},\cdots,x_N^{t+1}} \sum_{i=1}^{N} P\left(x_i^{t+1} \mid Z_t\right) Q\left(z_{t+1} \mid x_i^{t+1}\right) \tag{14}
$$

Here, the distribution P represents the probability distribution of agent i's state before the observation information z_t is given at time step t, and the distribution Q represents the likelihood distribution of obtaining the observation information z_t at state x_i. Also, $Z_t = [z_0, \cdots, z_t]$.

5. STEIN PARTICLE FILTER

The Stein Particle Filter (SPF) is a method for numerically analyzing non-Gaussian and nonlinear probabilistic state estimation problems, with its basic framework proposed in the literature[**?**]. In SPF, probability distributions are represented by particle sets, which are updated using a gradient method based on Stein Variational Gradient Descent (SVGD).

While traditional particle filters may suffer from particle degradation due to resampling, SPF smoothly transforms the particle distribution using SVGD and converges it to the target distribution. SVGD executes the minimization of the Kullback-Leibler information amount (KL divergence) using an operator called the Stein operator. Here, considering:

$$
q^* = \underset{q}{arg\,min} D_{KL}(q\|p) \tag{15}
$$

by formulating variational inference to converge from q to p using the Stein operator as in equation (11), particles can be moved according to the gradient to approach p.

Equation (14) is difficult to solve analytically using methods like the Kalman filter when P cannot be assumed to be a Gaussian distribution. In the Stein Particle Filter, the distribution $P\left(x_i \mid Z_t\right)$ is approximated as:

$$
P\left(x_i \mid Z_t\right) = \frac{1}{m}\sum_{j=1}^{m} P\left(x_j^t \mid Z_t\right) \tag{16}
$$

using a set of particles $\mathcal{X} = \left\{\S_{\rangle}^{\sqcup}\right\}_{\rangle=\infty}^{\Updownarrow}$, and by moving each particle according to the gradient method, the MAP estimation problem can be solved without algebraic approximation of the probability distribution. Like various filtering methods, the Stein Particle Filter is divided into a prediction step that calculates $P\left(x_{t+1} \mid z_t\right)$ and an update step that calculates $Q\left(z_{t+1} \mid x_{t+1}\right)$, as described below.

5.1. Calculation of $P\left(x_{t+1} \mid z_t\right)$ (Prediction Step)

In the prediction step, it is necessary to calculate $P\left(x_{t+1} \mid z_t\right)$ in equation (14). This is an operation to predict the state at step $t+1$ from the information up to step t, and in VINS, it is obtained by numerically integrating the acceleration a_m and angular velocity ω_m obtained from the IMU. Numerical integration on SE(3) can be calculated using a framework similar to the literature[**?**] as follows:

$$
\begin{aligned}
p_{k+1} =& p_k + v_k \Delta t \\
& + \int \int_{i \in \{z_t, i_{t+1}\}} \frac{\{R_k (a_m - b_n - \eta_{md}) + g\} \, dt^2}{\dot{a}} \\
v_{k+1} =& v_k + \int_{i \in \{z_t, i_{t+1}\}} \dot{a} dt \\
R_{k+1} =& R_k \otimes \exp \left(\int_{i \in \{z_t, i_{t+1}\}} \frac{(\omega_m - b_g - \eta_{gd}) \, dt}{\dot{\omega}} \right)
\end{aligned}
\tag{17}
$$

where p is position, v is velocity, R is the rotation matrix representing attitude, b_n and b_g are the biases of acceleration and angular velocity respectively, and η_{md} and η_{gd} are white noise. Each particle is updated by the transformation ${}^tT_{t+1} \in \mathrm{SE}(3)$ obtained by numerical integration as follows:

$$
\bar{x}^i_{t+1} = x^t_i \otimes^t T_{t+1}, \forall i \tag{18}
$$

5.2. Calculation of $Q(z_{t+1} \mid x_{t+1})$ (Update Step)

In the update step, it is necessary to calculate the likelihood $Q(z_{t+1} \mid x_{t+1})$, which represents the plausibility of the observation, and to transform the prediction distribution $P(x_{t+1} \mid z_t)$ by some method. In the Stein Particle Filter, equation (14) is replaced with the KL divergence minimization problem:

$$
\begin{aligned}
& \max_{x^i_{t+1}} P\left(x^i_{t+1} \mid Z_t\right) Q\left(z_{t+1} \mid x_{t+1}\right) \\
&= \max_{x^i_{t+1}} \underset{P_{QT}}{arg\,min} D_{KL} \left(P\left(x^i_{t+1} \mid Z_t\right)_{[T_i]} \| P_Q \right)
\end{aligned}
\tag{19}
$$

Furthermore, by changing the formulation of equation (18) to variational inference, the MAP solution can be obtained as follows:

$$
\begin{aligned}
& \min_{P_i(x)} \nabla_x D_{KL} \left(P\left(x^j_{i+1} \mid Z_i\right)_{[T_i]} \| P_{Q_i} \right)_{|\epsilon=0} \\
& \qquad\qquad \Rightarrow T := T \oplus \phi^* \\
& \phi^*(x) = \frac{1}{m} \sum_{j=1}^{m} \left(\nabla_{x_j} \log p(x_j) \, k(x, x_j) \right. \\
& \qquad\qquad \left. + \nabla_{x_j} k(x, x_j) \right)
\end{aligned}
\tag{20}
$$

where $k(\cdot, \cdot)$ is the kernel (generalized inner product) on SE(3):

$$
\begin{aligned}
k(x_i, x_j) &= \exp\left(-d^2_{ij} W d_{ij}\right) \\
d_{ij} &= x_j \boxminus x_i
\end{aligned}
\tag{21}
$$

6. STEIN RELAXED ADMM

In this paper, we propose a Stein Relaxed ADMM framework that combines the Stein Particle Filter and Relaxed ADMM to solve the cooperative VINS problem in environments where outliers frequently occur. This enables simultaneous realization of Stein variational gradient and constraint consensus by ADMM for relative position relationships that are consistent among multiple agents. As shown in detail in the derivation in the text, by combining the KL divergence minimization problem and Relaxed ADMM, a probabilistic consensus problem dealing with multi-modal distributions can be interpreted by the Stein gradient method.

From equations (14) and (18), the cooperative self-localization problem can be formulated as the following problem:

$$
\begin{aligned}
& \max_{x^t_1, \cdots, x^t_N} \sum_{i=1}^{N} \arg\min_{P_i(x)} D_{KL} \left(P_i \left(x^j_i \mid z^j_{i-1} \right)_{[T_i]} \| P_{Q_i} \right) \\
& \quad \text{s.t. } x^j_i = x^i_j, \forall (i, j) \in \mathcal{E}
\end{aligned}
\tag{22}
$$

Equation (21) can be relaxed as:

$$
\begin{aligned}
& \max_{x^t_1, \cdots, x^t_N} \sum_{i=1}^{N} \arg\min_{P_i(x)} D_{KL} \left(P_i \left(x^j_i \mid z^j_{i-1} \right)_{[T_i]} \| P_{Q_i} \right) \\
& \quad \text{s.t. } x^j_i = y_{ij}, x^i_j = y_{ij}, \forall (i, j) \in \mathcal{E}
\end{aligned}
\tag{23}
$$

to be converted to the form of equation (5).

Definition 1. Here, as the extended Lagrangian in Relaxed ADMM in equation (9), we redefine the extended Lagrangian that takes probability distributions as arguments as:

$$
\mathcal{L}_{\mathcal{F},\dagger}\left(\S_\rangle, \asymp\right) = \mathcal{F}_\rangle\left(\S_\rangle\right) - \int_{\mathcal{X}} \asymp \lceil \S_\rangle + \frac{\bigcirc}{\in} \int_{\mathcal{X}} \lceil \S_\rangle^\in \tag{24}
$$

This allows equation (22) to be transformed into the following algorithm:

$$
\begin{aligned}
y^*_{ij} &= \arg\min_{y_{ij}} \mathcal{L}_{\amalg,\dagger}\left(\ddagger_{\rangle|,\rangle} + \ddagger_{\rangle|,|}\right) \\
&= \arg\min_{y_{ij}} \langle z_{ij,i} + z_{ij,j}, y_{ij} \rangle + \frac{\gamma}{2} \|y_{ij}\|^2 \\
(\omega_g)_{ij,i} &= z_{ij,i} - \gamma y^*_{ij} \\
x^*_i &= \arg\min_{x_i} \mathcal{L}_{\mathcal{F},\dagger}\left(\S_\rangle, \ddagger_\rangle\right) \\
&= \arg\min_{x_i} D_{KL}\left(P_i(T_i) \| P_{Q_i}\right) \\
& \quad - \sum_{j \in \mathcal{E}(\rangle)} \int_{P_i} \left\{ 2(\omega_g)_{ij,i} - z_{ij,i} \right\} dx_i + \frac{\gamma}{2} |\mathcal{E}(\rangle)| \int_{\mathcal{P}_\rangle} \lceil \S_\rangle^\in
\end{aligned}
\tag{25}
$$

$$
(\omega_f)_{ij,i} = 2(\omega_g)_{ij,i} - z_{ij,i} - \gamma x^*_i \tag{26}
$$