

MegaParticles

GPU-Accelerated Stein Particle Filter による 点群 6 自由度姿勢推定

小出 健司^{*1}, 大石 修士^{*1}, 横塚 将志^{*1}, 阪野 貴彦^{*1}

MegaParticles

Range-based Localization with GPU-Accelerated Stein Particle Filter

Kenji Koide^{*1}, Shuji Oishi^{*1}, Masashi Yokozuka^{*1}, Atsuhiko Banno^{*1}

^{*1} National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Ibaraki 305-8568, Japan

This paper presents a 6-DoF range-based Monte Carlo localization method with a GPU-accelerated Stein particle filter. To update a massive amount of particles, we propose an approximated Gauss-Newton-based Stein variational gradient descent (SVGD). This method collectively updates particle states with gradient and neighborhood information, which provides efficient particle sampling. For an efficient neighbor particle search, it uses locality sensitive hashing and iteratively updates the neighbor list of each particle over time. The neighbor list is then used to propagate the posterior probabilities of particles over the neighbor particle graph. In experiments, the proposed method showed an extreme robustness to complete sensor occlusion (i.e., kidnapping), and enabled pinpoint sensor localization without any prior pose information.

Key Words : Monte carlo localization, Point cloud, GPU computing

1. 緒 言

自己位置推定は自律システムの基礎となる重要な機能であり、サービスロボットから自動運転車両まで幅広い応用にとって不可欠である。特に近年は三次元距離センサの高精度・廉価化をうけ、点群ベースの自己位置推定手法が盛んに開発されている。

3次元空間での姿勢推定手法としては、ICP⁽¹⁾やNDT⁽²⁾などのスキャンマッチングを反復的に適用するもの広く利用されている。点群・IMU制約の密結合⁽³⁾やウィンドウ最適化⁽⁴⁾を組み合わせた高速・高精度な自己位置推定手法が多数提案されているが、これらの殆どは点群データの時間的連続性に大きく依存しており、遮蔽や通信不良によるデータ欠落には瞬間的(e.g., 1 s)なものであっても致命的な影響を受ける。また、姿勢初期値への依存性も強く、自己位置の事前情報が誤差を含むあるいは全く与えられない場合への対処は極めて困難である(e.g., GNSSが無効な環境での初期位置推定など)。

Monte Carlo Localization (MCL) は有限個のサンプル集合によって状態分布を表現し、確率的状态推定によって自己位置を求める手法である⁽⁵⁾。代表的なもの

として適応的パーティクルフィルタを使った手法が知られており、2次元LiDARによる自己位置推定に広く用いられている⁽⁶⁾⁽⁷⁾⁽⁸⁾。これらの手法では、サンプリングによるノンパラメトリック分布表現によって非正規多峰性分布を柔軟に扱うことができるため、環境の繰り返しや観測の欠落により推定に強い曖昧性が起きる場合にも対処することができる。さらに、環境を覆うほどの十分な量のパーティクルを生成することで、初期姿勢が与えられない状況からでも大域的姿勢推定が行えることが知られている。このようにMCL手法は強力な分布表現能力や大域姿勢推定能力を持つ一方、6自由度姿勢推定への適用例は限定的なものにとどまっている。これは、次元が増えるにつれて単位空間を充足するのに必要なパーティクル数が指数的に増加するため(i.e., 次元の呪い)、2D MCLと同様の強力な状態推定能力を6自由度空間で発揮するためには、莫大な量のパーティクルが必要となるためである。そのため、多くの手法では状態空間を3自由度または4自由度に限定した上での3次元空間上のMCLが行われている⁽⁹⁾⁽¹⁰⁾。

近年、6自由度MCLを実現するための手法がいく

^{*1} 産業技術総合研究所 (〒 305-8568 茨城県つくば市梅園 1-1-1)
k.koide@aist.go.jp

補足動画および補足データ:<https://staff.aist.go.jp/k.koide/projects/rs2024/>

つか提案されている．これらの手法では車速情報の利用⁽¹¹⁾や並進・回転成分分解⁽¹²⁾によってサンプリング効率を改善することで，比較的少数のパーティクル (100 ～ 1000 個) を使った実時間での 6 自由度状態推定を実現している．特筆すべき手法としてパーティクルフィルタと Stein variational gradient descent (SVGD)⁽¹³⁾を組み合わせた Stein particle filter⁽¹⁴⁾が提案されている．SVGD では尤度関数の勾配情報とパーティクルの隣接関係情報を用いることで状態空間のサンプリング効率を改善し，少数のパーティクルによる効率的な状態推定を行うことができる．しかし，少数のパーティクルでは根本的に 6 自由度空間における多峰性分布を正確に表現できないため，曖昧性が強い状況において推定性能が大きく劣化する．また，SVGD ではパーティクル数 N に対して $O(N^2)$ の処理時間がかかるため一定数以上のパーティクルを扱うことができない．

本稿では，GPU 並列計算を活用した三次元地図上での 6 自由度 MCL 手法を提案する．提案手法では，GPU の性能を最大限引き出せるようにアルゴリズム全体を構築することで，Fig. 1 に示すような従来にない超多量 (1024² 個) のパーティクルを実時間で処理する．また，効率的な状態サンプリングのため，近似 Gauss-Newton SVGD に基づくパーティクル状態更新手法を提案する．提案アルゴリズムでは，locality sensitive hashing (LSH) により効率的に SE3 空間における近傍パーティクル探索を行うことで，SVGD の計算量を $O(N^2)$ から $O(N)$ に軽減させる．さらに，大量のパーティクルの中から推定代表値を抽出するため，近傍グラフ上でのベイズ推定に基づくパーティクル事後確率推定手法を提案する．

GPU による超大量のパーティクル処理と SVGD による効率的な状態サンプリングの組み合わせによって，提案手法は強力な曖昧性表現能力および大域姿勢推定能力を有する．繰り返し構造を含む屋内環境や，動的物体や植生の変化のある屋外環境での実験を通し，提案手法が初期値なしでの大域初期姿勢推定および高精度なピンポイント自己位置推定を可能とし，さらに重度の誘拐状態からの復帰が可能であることを示す．

2. 提案手法

2.1 提案手法外観 本研究では，三次元環境地図点群 $\mathcal{M} = \{\mathbf{p}_k^M \in \mathbb{R}^3 \mid k=1, \dots, N^M\}$ が与えられたもとで，その地図上でのセンサ姿勢 $\mathbf{T}_t \in \text{SE3}$ をセンサ点群観測 $\mathcal{P}_t = \{\mathbf{p}_k^S \in \mathbb{R}^3 \mid k=1, \dots, N^S\}$ から求める．

アルゴリズムの大枠としては，通常のパーティクルフィルタと同様に推定対象となる状態分布を有限個のサンプル (パーティクル) 集合 $\mathcal{X}_t = \{\mathbf{x}_t^i \mid i=1, \dots, N^P\}$ で

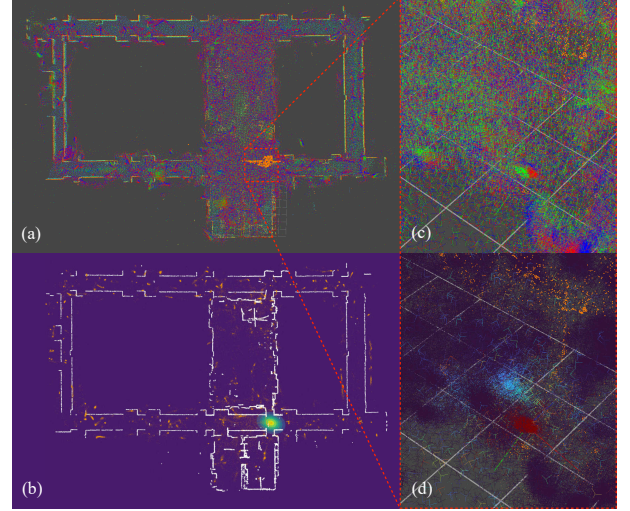


Fig. 1: (a) Proposed method performs 6-DoF sensor localization with one million particles. All the particles are evaluated and updated in real-time on a single GPU. (b) Posterior probability distribution. (c) A close look at the maximum posterior particle, and (d) the same view with particles colored based on the posterior probabilities.

表現し，予測ステップと修正ステップを再帰的に繰り返すことで状態推定を行う．各パーティクルは現時刻におけるセンサ姿勢の仮説を表す ($\mathbf{T}_t^i := \mathbf{x}_t^i$).

2.2 予測ステップ 予測ステップでは通常のパーティクルフィルタと同様に，状態遷移予測を基に各パーティクルの状態を独立に更新する．ここでは Generalized ICP (GICP) スキャンマッチング⁽¹⁵⁾を直前の点群 $\mathcal{P}_{t-1}$ と現在点群 $\mathcal{P}_t$ の間で適用し，センサ移動量予測 $\Delta \mathbf{T}_t \sim \mathbf{T}_{t-1}^{-1} \mathbf{T}_t$ を得る．同時にスキャンマッチング結果における誤差関数のヘッセ行列を基に予測誤差共分散行列 $\Sigma^{\Delta \mathbf{T}} \in \mathbb{R}^{6 \times 6}$ を求める．得られたセンサ移動量予測分布を基に，各パーティクルの状態を次式によって更新する．

$$\mathbf{T}_t^i = \mathbf{T}_{t-1}^i \Delta \mathbf{T}_t \exp(\boldsymbol{\delta}^i). \quad (1)$$

ここで $\boldsymbol{\delta}^i \sim \mathcal{N}(0, \Sigma^{\Delta \mathbf{T}})$ は $\Delta \mathbf{T}_t$ の接空間における正規ノイズを表す．

2.3 修正ステップ

2.3.1 尤度関数 パーティクルの更新に用いる対数尤度関数 $\log p(\mathcal{P}_t | \mathbf{x}_t^i)$ として GICP スキャンマッチングにおける分布対分布距離⁽¹⁵⁾を用いる．GICP では点群中の各点 $\mathbf{p}_k^*$ は局所的な幾何形状を表すガウス分布 $\mathcal{N}(\boldsymbol{\mu}_k^*, \Sigma_k^*)$ としてモデル化される．スキャン点群の各点 $\mathbf{p}_k^S$ に対する地図点群中の最近傍点 $\mathbf{p}_k^M$ を求め

たのち、各パーティクルの対数尤度を次式で求める。

$$\log p(\mathcal{P}_t | \mathbf{x}_t^i) = - \sum_k \mathbf{e}_k^\top \boldsymbol{\Omega}_k \mathbf{e}_k, \quad (2)$$

$$\mathbf{e}_k = \boldsymbol{\mu}_k^M - \mathbf{T}_t^i \boldsymbol{\mu}_k^S, \quad (3)$$

$$\boldsymbol{\Omega}_k = (\boldsymbol{\Sigma}_k^M + \mathbf{T}_t^i \boldsymbol{\Sigma}_k^S (\mathbf{T}_t^i)^\top)^{-1}. \quad (4)$$

ここで、 $\mathbf{e}_k$ は対応点間の残差であり、 $\boldsymbol{\Omega}_k$ は対応点の共分散行列を混合して得られる精度行列である。

効率的に地図上の任意地点における最近傍点を求めるため、事前に地図空間を一定解像度 (e.g., 0.1 m) で格子状に区切った各ボクセルに対する最近傍点を求めた最近傍マップ m^{nnf} を計算しておく。対数尤度の計算時には m^{nnf} から対応点をルックアップすることで近傍探索処理を省略する。

2.3.2 パーティクル状態更新 状態空間サンプリング効率を改善しつつ高速にパーティクル状態の更新を行うため、ここでは近似 Gauss-Newton SVGD アルゴリズムを提案する。

Eq. 2 で定義される対数尤度関数は最小二乗形式となっているため、これを最大化するための二次最適状態更新量は Gauss-Newton 法に従い、 $\boldsymbol{\psi}^i = \mathbf{H}^{-1} \mathbf{b}$ として求めることができる。ただし、情報行列 $\mathbf{H}$ と情報ベクトル $\mathbf{b}$ はそれぞれ次式で与えられる。

$$\mathbf{H} = \sum_k \mathbf{J}_k^\top \boldsymbol{\Omega}_k \mathbf{J}_k, \quad \mathbf{b} = \sum_k \mathbf{J}_k^\top \boldsymbol{\Omega}_k \mathbf{e}_k, \quad \mathbf{J}_k = \frac{\partial \mathbf{e}_k}{\partial \mathbf{T}_t^i}. \quad (5)$$

得られた二次最適状態更新量 $\boldsymbol{\psi}^i$ を基に、SVGD 更新式⁽¹³⁾を改変した次式によってパーティクル状態を更新する。

$$\mathbf{T}_{t+1}^i = \mathbf{T}_t^i \exp(\phi(\mathbf{x}_t^i, \mathcal{P}_t)), \quad (6)$$

$$\phi(\mathbf{x}_t^i, \mathcal{P}_t) = \frac{\sum_{\mathbf{x}_t^j \in \tilde{\mathcal{X}}_t^i} (k(\mathbf{T}_t^i, \mathbf{T}_t^j) \boldsymbol{\psi}^j + \nabla_{\mathbf{T}_t^j} k(\mathbf{T}_t^i, \mathbf{T}_t^j))}{\sum_{\mathbf{x}_t^j \in \tilde{\mathcal{X}}_t^i} k(\mathbf{T}_t^i, \mathbf{T}_t^j)}. \quad (7)$$

ここで、 k はパーティクル間の近接度を定義する正定値カーネル関数であり、 $\tilde{\mathcal{X}}_t^i$ は $\mathbf{x}_t^i$ 自身を含む近傍パーティクル集合である。

Eq. 7 において、 $\sum k(\mathbf{T}_t^i, \mathbf{T}_t^j) \boldsymbol{\psi}^j$ はカーネル関数によって重みづけされた近傍パーティクルの最適状態更新量の平均を表し、各パーティクルを尤度関数を最大化する方向へ移動させる効果を持つ。一方、 $\nabla_{\mathbf{T}_t^j} k(\mathbf{T}_t^i, \mathbf{T}_t^j)$ はカーネル関数の勾配方向を表しており、パーティクル同士を遠ざける斥力のような作用をもたらす。これによりパーティクルが尤度関数の極値に集中するのを防ぎ、状態空間におけるパーティクルの多様性を保つことでサンプリング効率を向上させる。

Algorithm 1 IterativeNeighborParticleSearch

```

1:  $\mathbf{T}^{\text{LSH}} \leftarrow$  Random SE3 transformation
2:  $\mathcal{B} \leftarrow [\emptyset]_{i=1, \dots, N^B}$  ▷ LSH buckets
3: for  $\mathbf{x}_t^i \in \mathcal{X}_t$  do ▷ Distribute loop
4:    $h = f^{\text{LSH}}(\mathbf{x}_t^i) \bmod N^B$ 
5:    $\mathcal{B}[h] \leftarrow \mathcal{B}[h] \cup \mathbf{x}_t^i$  ▷ Add particle to the bucket
6: for  $\mathbf{x}_t^i \in \mathcal{X}_t$  do ▷ Gather loop
7:    $h = f^{\text{LSH}}(\mathbf{x}_t^i) \bmod N^B$ 
8:   for  $\mathbf{x}_t^j \in \mathcal{B}[h]$  do
9:      $\tilde{\mathcal{X}}_t^i \leftarrow \tilde{\mathcal{X}}_t^i \cup \mathbf{x}_t^j$  ▷ Add to the neighbor list
10:    if  $|\tilde{\mathcal{X}}_t^i| > K$  then
11:       $\hat{\mathbf{x}} = \arg \min_{\mathbf{x}_t^k \in \tilde{\mathcal{X}}_t^i} k(\mathbf{x}_t^i, \mathbf{x}_t^k)$  ▷ Farthest particle
12:       $\tilde{\mathcal{X}}_t^i \leftarrow \tilde{\mathcal{X}}_t^i \setminus \hat{\mathbf{x}}$  ▷ Remove  $\hat{\mathbf{x}}$  from the neighbor list

```

カーネル関数 k はパーティクル間の距離に基づき、各パーティクルが互いの状態更新にどの程度影響するかを決定する。ここでは次式の指数カーネルを用いる。

$$k(\mathbf{T}_t^i, \mathbf{T}_t^j) = \exp(-\mathbf{d}_{ij}^\top \mathbf{W}^K \mathbf{d}_{ij}), \quad (8)$$

$$\mathbf{d}_{ij} = \log((\mathbf{T}_t^i)^{-1} \mathbf{T}_t^j). \quad (9)$$

$\mathbf{W}^K = \text{diag}([\sigma_r, \sigma_r, \sigma_r, \sigma_t, \sigma_t, \sigma_t])$ はパーティクル間の相対姿勢接ベクトルの各要素に対する重み行列である (e.g., $\sigma_r = 5.0 \text{ rad}^{-1}$ and $\sigma_t = 2.5 \text{ m}^{-1}$).

元の SVGD⁽¹³⁾では状態更新に全てのパーティクルを用いるため、パーティクル数が増大すると加速度的に処理量が上昇する。ここでは、遠方のパーティクルはカーネル関数で与えられる重みが小さくなり、状態更新量 $\phi(\mathbf{x}_t^i, \mathcal{P}_t)$ に小さな影響しか与えないことを考慮し、あるパーティクルに対する近傍 K 個のパーティクルのみを用いて状態更新量を求める (e.g., $K = 20$).

2.3.3 LSHを用いた最近傍パーティクル探索 提案手法では非ユークリッド SE3 空間における超多数かつ動的なパーティクル集合を扱うため、一般的に用いられる直線探索や空間分割に基づく最近傍探索を用いることができない。

効率的な近傍パーティクル探索のため、ここでは安定分布に基づく LSH (局所性鋭敏型ハッシュ)⁽¹⁶⁾を SE3 空間に拡張した反復的近傍探索アルゴリズムを提案する。Algorithm 1 に処理の流れを示す。

LSH は確率的ハッシュ関数の一種で、高い確率で空間的に近い要素に同一のハッシュ値を割り当てる関数である。提案アルゴリズムではまず、LSH 関数を用い各パーティクルを N^B 個のバケットを持つハッシュテーブルに格納する (Line 3 - 5)。ここでは次式で定義される SE3 空間上での LSH 関数 $f^{\text{LSH}} : \text{SE3} \rightarrow \mathbb{Z}$ を用いる。

$$f^{\text{LSH}}(\mathbf{T}_t^i) = f^{\text{INT}}([\zeta[k]]_{k=1, \dots, 6}), \quad (10)$$

$$\zeta = \alpha \mathbf{W}^K \log((\mathbf{T}^{\text{LSH}})^{-1} \mathbf{T}_t^i) + \delta^{\text{LSH}}. \quad (11)$$

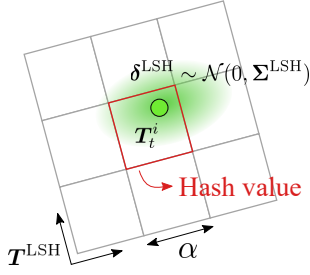


Fig. 2: SE3 locality sensitive hashing based on a stable distribution.

$f^{\text{INT}} : \mathbb{Z}^6 \rightarrow \mathbb{Z}$ は整数の組からハッシュ値を求める関数⁽¹⁷⁾であり, T^{LSH} はハッシュ値を計算する接空間を定義するランダムな姿勢を表す. $\delta^{\text{LSH}} \sim \mathcal{N}(0, \Sigma^{\text{LSH}})$ は正規ノイズ, α は姿勢を量子化するためのグリッドのスケールを定義する定数である.

Fig. 2 に f^{LSH} によるハッシュ値計算の模式図を示す. T^{LSH} によってパーティクル姿勢 T_t^i を量子化するグリッド系が定義され, これにランダムな姿勢を設定することでグリッド系に対するパーティクル位置が与えるバイアスをキャンセルする. グリッド系におけるノイズ δ^{LSH} を加えた後にパーティクルが落ちたグリッド上のセルに応じて一意なハッシュ値が与えられる. 近傍のパーティクルは高い確率で同じセルに落ちるため, ハッシュテーブル上の同じバケットに割り当てられたパーティクルを見つけることで効率的に近傍パーティクル候補を列挙することができる (Line 6-9). この方法では, 各パーティクルに対する近傍探索に定数時間しかかからないため, 全体としてパーティクル数に比例する程度の処理時間で近傍探索を行うことができる.

しかし, f^{LHS} は確率的関数であるため, 列挙されたパーティクル集合は擬陽性の近傍パーティクルを含む. これに対処するため, パーティクルの近傍関係は短い時間間隔では大きく変化しないと仮定し, 各パーティクルに対する近傍パーティクル集合 $\tilde{\mathcal{X}}_t^i$ を複数フレームにまたいで反復的に更新し, 上位 K 個の近傍パーティクルのみを保持する (Line 10-12). 近傍検出精度を高めるため, α を2倍ずつ変化させた複数 (e.g., 4つ) のハッシュテーブルを作成し, それぞれに対して Algorithm 1 を複数回 (e.g., 2回) 実行する.

2.4 近傍グラフ上での事後確率伝搬 通常のパーティクルフィルタと異なり, SVGD によるパーティクル状態更新ではリサンプリング処理を行わない. 尤度が小さい領域におけるパーティクルも全てそのまま生存するため, 提案手法は sample impoverishment 問題⁽¹⁸⁾の影響を受けず, パーティクル集合の多様性を保

つことができる. 一方で, 多数のパーティクル集合によって表現される分布は強い非線形性・多峰性を示すため, 単純なパーティクルの重み付け和や最密地点推定などでは妥当な推定結果の代表値を求めることができない. 安定して推定状態の代表値を求めるため, ここでは前述の反復的近傍パーティクル探索の副産物として得られる近傍グラフ上でのパーティクル事後確率伝播法を提案する.

提案手法では各パーティクル x_t^i に対して明示的に事後確率値 $p(x_t^i | \mathcal{P}_t)$ を推定する. まず, 各パーティクルの事前確率 $p(x_t^i)$ と Eq. 2 により与えられる尤度 $p(\mathcal{P}_t | x_t^i)$ を基に, 事後確率の初期値 $p(x_t^i | \mathcal{P}_t) \propto p(x_t^i)p(\mathcal{P}_t | x_t^i)$ を求める. その後, 次式に示すように, 近傍パーティクルの事後確率の重み付き平均を取ることによって各パーティクルの事後確率を周辺のパーティクルへと伝搬させ, 更新された事後確率 $p'(x_t^i | \mathcal{P}_t)$ を得る.

$$p'(x_t^i | \mathcal{P}_t) = \frac{\sum_{x_t^j \in \tilde{\mathcal{X}}_t^i} k(T_t^i, T_t^j) p(x_t^j | \mathcal{P}_t)}{\sum_{x_t^j \in \tilde{\mathcal{X}}_t^i} k(T_t^i, T_t^j)}, \quad (12)$$

Eq. 12 は各パーティクルを頂点とし近傍パーティクル間を辺として繋いだ近傍グラフ上で, ランダムウォーク仮定の下で事後確率を伝搬させていることに相当する. 複数回 (e.g., 10回) 反復的にこれを適用することで, 広範なパーティクルへ事後確率を伝搬させる. その後, 最大事後確率を持つパーティクルを代表値 $\tilde{T}_t$ として選択する.

2.5 軌跡スムージング 提案手法では, 姿勢の連続性は考慮せずに最大事後確率を持つパーティクルを代表値として選択するため, 点群のノイズなどの影響で瞬間的に推定姿勢がジャンプする事が起きる (i.e., ジッターリング). これを防ぐため, 推定姿勢代表値 $\tilde{T}_t$ の系列を元に, 次式の誤差 e^{traj} を最小化する平滑化された姿勢系列 $\hat{T}_t$ を求める.

$$e^{\text{traj}} = e^{\text{fitting}} + e^{\text{smooth}}, \quad (13)$$

$$e^{\text{fitting}} = \sum_{\tilde{T}_t} \left(\left\| \rho(\mathbf{W}^P \log(\tilde{T}_t^{-1} \hat{T}_t)) \right\|^2 \right), \quad (14)$$

$$e^{\text{smooth}} = \sum_{\tilde{T}_t} \left(\left\| \mathbf{W}^S \log(\hat{T}_{t-1}^{-1} \hat{T}_t) \right\|^2 \right). \quad (15)$$

e^{fitting} は代表値系列 $\tilde{T}_t$ に対するフィッティング誤差, e^{smooth} は前後フレームにおける姿勢の連続性 (スムースネス) 誤差を表す. ρ は Huber ロバストカーネル, $\mathbf{W}^P \in \mathbb{R}^{6 \times 6}$ および $\mathbf{W}^S \in \mathbb{R}^{6 \times 6}$ はフィッティングおよびスムースネス誤差の重み調整行列である.

3. 実験

3.1 屋内実験

3.1.1 実験設定 提案手法の大域姿勢推定性能を示すため、対称・反復的構造を持つ屋内環境で自己位置推定実験を行った。センサには Microsoft Azure Kinect を用い、点群を 10Hz で計測した。基礎的な姿勢推定性能を評価するための激しい動きやデータ欠落のない 2 シーケンス (Easy01, Easy02) と、長時間のデータ欠落に対する頑強性を評価するための 2 シーケンス (Kidnap01, Kidnap02) を計測した。Kidnap01 および Kidnap02 では、各シーケンスにおいて三回ずつ、センサの視界を長時間 (10 - 20 s) 完全に塞ぎながら部屋から別の部屋を移動しており、既存手法では対処が困難な重度の誘拐問題を含むシーケンスとなっている。

リファレンスとするセンサ姿勢系列を得るため、各スキャン点群を手動で地図点群にアラインしたのち、scan-to-map 点群レジストレーション制約と IMU 移動量制約のバッチ最適化を行い、スムーズかつ地図に対してアラインされた点群姿勢系列を求めた。

比較対象として、FAST_LIO_LOC¹ および hdl_localization⁽¹⁹⁾ を用いた。FAST_LIO_LOC は iterative error state Kalman filter 上での LiDAR-IMU 密結合に基づく FAST_LIO2⁽³⁾ を用いて移動量推定を行い、定期的に scan-to-map 点群レジストレーションを行うことで地図に対する姿勢推定誤差を補正する。hdl_localization⁽¹⁹⁾ は NDT を用いた scan-to-map 点群レジストレーションと IMU による状態予測を Unscented Kalman filter 上で統合する。なお、IMU データは比較対象手法にのみ用い、提案手法の推定は点群データのみを用いて行った。

提案手法の初期パーティクル集合として、地図全体を覆う SE3 空間上の一様分布 ($50 \text{ m} \times 35 \text{ m} \times 4 \text{ m} \times \text{full SO3}$) の基で合計 $1024^2 = 1,048,576$ 個のパーティクルを生成した。

推定結果のセンサ軌跡の評価には *evo* ツールキット² を用い、Absolute Trajectory Error (ATE)⁽²⁰⁾ を計算した。提案手法には初期姿勢情報が一切与えられていないため、各シーケンスにおいて事後確率分布が単一地点に収束するまでの最初の 10 - 30 s は除外して評価を行った。

3.1.2 大域初期姿勢推定 Fig. 3 に Easy01 において、提案手法のパーティクル事後確率分布が初期状態の一様分布からどのように収束したかを示す。可視化のため、パーティクル事後確率を重みとしてカーネル密度推定を適用し、2 次元座標上での事後確率密度分布を表示している。

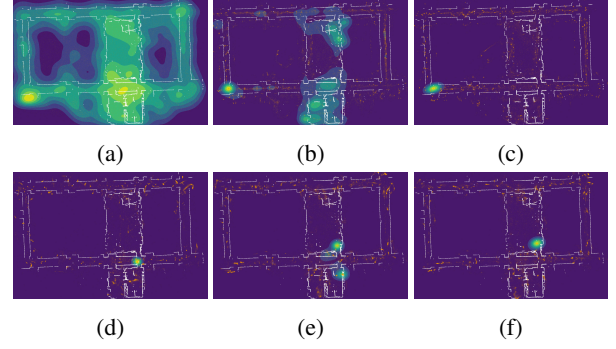


Fig. 3: Estimation result for Easy01 sequence. Particles were initialized with a uniform distribution without any prior information (a). The positional ambiguity was quickly resolved as the sensor moved (b, c). Although there were still two major hypotheses for the sensor orientation (upright and flipped) (d, e), it was resolved when the sensor entered a room (f).

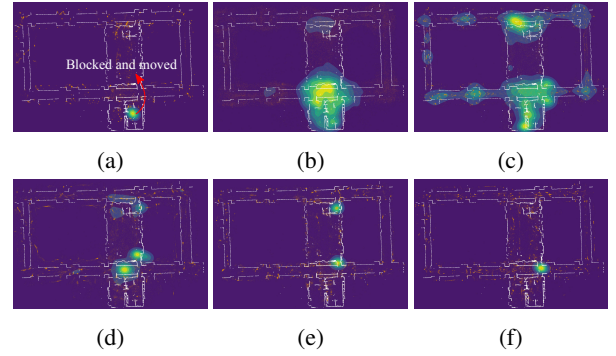


Fig. 4: Kidnapping experiment result (Kidnap02). The sensor view was completely occluded, and the uncertainty grew because of the lack of observations (a, b). When the occlusion was removed and the sensor began to see the world, the posterior probability distribution quickly converged to a few positions (c, d, e). Eventually, the posterior distribution successfully converged to the correct position (f).

パーティクルは事前情報を含まない一様分布で初期化されているが (Fig. 3 (a)), センサが動くにつれて位置曖昧性が解決され、急速に正解付近の一点に事後確率分布が収束した (Fig. 3 (b)(c)). 上下対象の廊下を直進しているため、この時点では向き曖昧性が残っており、正常姿勢と上下反転した姿勢が重なった推定状態となっている (Fig. 3 (d)). そのため、曲がりながら部屋に入った際に推定が逆方向に進行する 2 つの仮説に分岐したが (Fig. 3 (e)), 最終的に部屋内部の構造を観測することで曖昧性が解決され、正解姿勢付近の単一仮説に収束した (Fig. 3 (f)).

¹https://github.com/HViktorTsoi/FAST_LIO_LOCALIZATION

²<https://github.com/MichaelGrupp/evo>

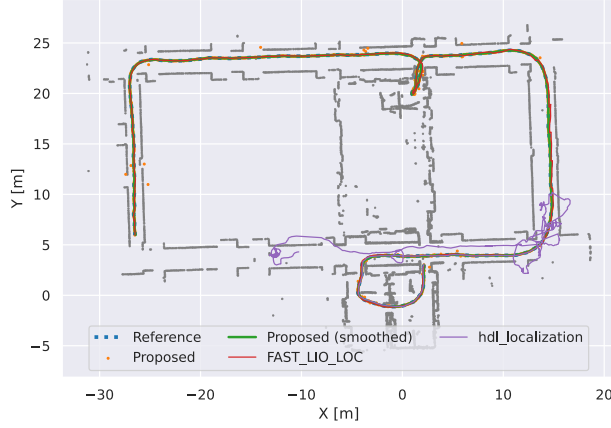


Fig. 5: Estimated trajectories for the Easy02 sequence.

Table 1: Absolute trajectory errors for indoor sequences

Method	ATE [m]			
	Easy01	Easy02	Kidnap01	Kidnap02
FAST_LIO (odom)	1.86 ± 0.85	6.16 ± 3.02	×	×
FAST_LIO_LOC	0.07 ± 0.05	0.14 ± 0.10	×	×
hdl_localization	0.14 ± 0.10	16.8 ± 10.3	×	×
Proposed	0.25 ± 0.24	0.13 ± 0.12	5.64 ± 4.86	5.94 ± 5.00
Proposed (smooth)	0.02 ± 0.01	0.02 ± 0.01	4.74 ± 4.13	4.75 ± 4.07

× indicates that the estimation became corrupted.

3.1.3 誘拐状態からの復帰 Fig. 4 に, Kidnap02 において提案手法が誘拐状態から復帰した際の様子を示す. センサの視界が塞がれ点群観測が完全に失われている間, 事後確率分布は環境全体に発散した (Fig. 4 (b) (c)). センサは視界を塞がれたまま別の部屋に移動されたが, 再び点群が得られるようになると急速に環境中で観測点群に近い形状を持つ数地点へ事後確率分布が収束し (Fig. 4 (d) (e)), 最終的に正解姿勢に収束し, 誘拐状態から無事に復帰した (Fig. 4 (f)). Kidnap01 および Kidnap02 を通して計 6 回の誘拐状態が起きたが, いずれの場合もセンサデータが再び有効になってから数秒以内に無事に誘拐状態から復帰することができた.

比較手法の FAST_LIO_LOC および hdl_localization はいずれも誘拐状態になった際に推定が破綻しており, この結果から提案手法の従来にない強力な大域姿勢推定能力を確認することができた.

3.1.4 推定精度評価 Fig. 5 に Easy02 における各手法による推定軌跡を示す. 提案手法は初期一様分布から収束して以降はシーケンスの最後まで安定した姿勢推定結果を示した. 最大事後確率パーティクルによる推定代表値には瞬間的な推定値のジッタが含まれているが, 軌跡フィルタリング処理をかけた結果は FAST_LIO_LOC と同等以上の精度を示した.

Table 1 に各手法による推定結果の ATE を示す. Easy01 および Easy02 において, 地図に対する姿勢補正のない FAST_LIO 単体での結果は大きな推定誤

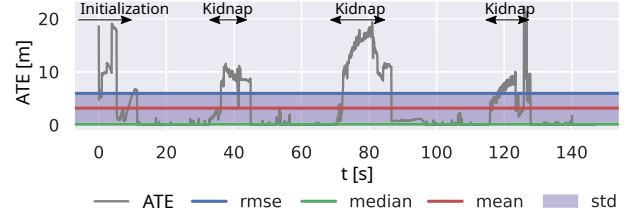


Fig. 6: ATE of the proposed method for the Kidnap02 sequence.

Table 2: Processing time for indoor environments

Process	Time [ms]
Neighbor list update	26.67 ± 0.35
Likelihood evaluation	55.17 ± 3.84
Particle state update	1.59 ± 0.03
Posterior probability update	7.30 ± 1.02
Total	90.8 ± 4.20

差 (1.86 m, 6.16 m) を含んでいるが, 地図補正を行う FAST_LIO_LOC では精度が大きく向上している (0.07 m, 0.14 m). hdl_localization は Easy01 に対しては良い推定精度 (0.14 m) を示したが, Easy02 では幾何特徴の乏しい廊下環境での点群マッチングに失敗し, 大きな推定誤差を示した (16.8 m). 提案手法は推定結果のジッタによって瞬間的に推定誤差が大きくなる場面はあったが, 全体を通して FAST_LIO_LOC と同等程度の推定精度を示している (0.25 m, 0.13 m). さらにフィルタリング処理によって平滑化された軌跡では従来手法を上回る推定精度を示した (0.02 m, 0.02 m).

Kidnap01 および Kidnap02 においては, FAST_LIO_LOC および hdl_localization は誘拐状態において完全に推定が破綻した. 一方で提案手法はいずれのシーケンスにおいても無事に誘拐状態から復帰し, 最後まで姿勢推定を継続することができた. Fig. 6 に Kidnap02 における提案手法の ATE 推移を示す. シーケンス開始時に急速に正解姿勢付近に推定が収束して ATE が減少した後, 複数回の誘拐状態が起きたあとも正常に推定が復帰し ATE が減少しており, 最後まで姿勢推定を継続できていることが確認できる.

3.1.5 計算時間 Table 2 に提案手法の各ステップにおける処理時間を示す. システム全体は GPU の性能を最大限引き出せるように高度に並列化して実装されており, 単一 GPU 上で 1 フレームあたり平均 91msec で処理を行うことができた.

3.2 屋外実験

3.2.1 実験設定 Fig. 7 に示す屋外環境において自己位置推定実験を行った. センサには Livox MID360 を用い, 点群を 10Hz で計測した. 屋内実験と同様に,

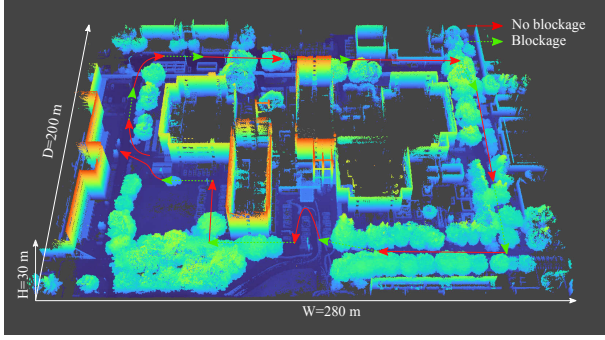


Fig. 7: Outdoor experimental environment. During the experiment, complete blockages of the LiDAR (Livox MID-360) made the point cloud feed unavailable for 10-30 s.

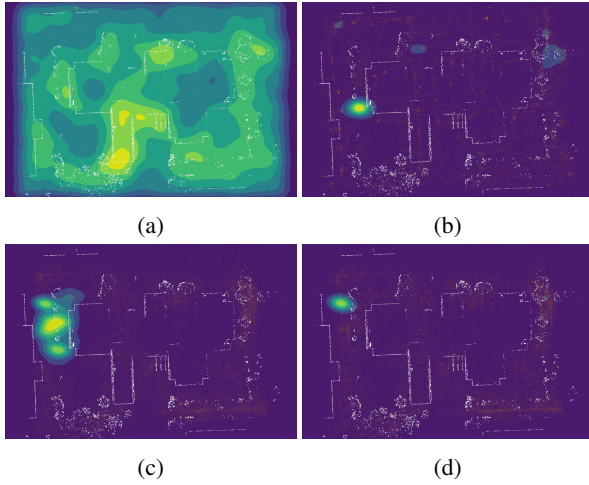


Fig. 8: Outdoor experiment results. Uniformly initialized particles quickly converged to the correct position (a, b), and successfully recovered from kidnapping (c, d) during the experiment.

完全にセンサの視界を塞ぎながら移動する誘拐状態を合計 8 回行った。なお、姿勢推定用データは地図データの計測から約 1 ヶ月後に計測されたため、動的物体(停車車両や歩行者など)や植生の変化が含まれている。

3.2.2 推定結果 Fig. 8 に実験中の提案手法の事後確率分布の推移を示す。一様分布で初期化されたパーティクル集合 (Fig. 8 (a)) は、実験開始後約 3 秒で正解地点に収束した (Fig. 8 (b))。誘拐状態においては事後確率分布は近辺に発散したが (Fig. 8 (c))、センサデータが有効になると急速に正解地点に収束し約 2 秒で推定を復帰することができた (Fig. 8 (d))。

Fig. 9 に推定軌跡を示す。実験を通して複数回かつ長時間点群データが欠落しているが、提案手法はいずれの場合も問題なく推定を復帰し、最後まで姿勢推定を継続することができた。ほとんどの場合において数

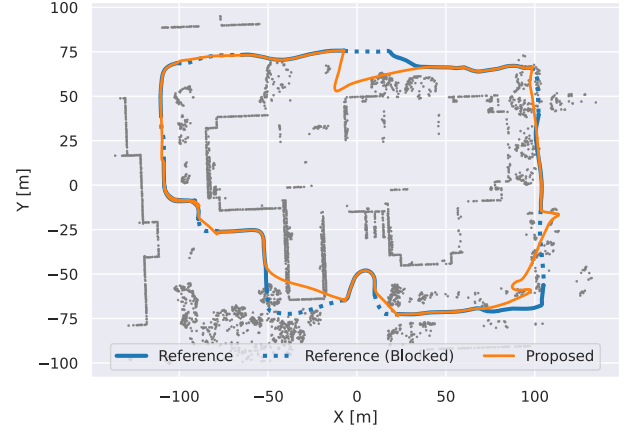


Fig. 9: Estimated trajectory for the outdoor experiment. Although the sensor was completely occluded eight times, the proposed method successfully recovered the estimation during the experiment.

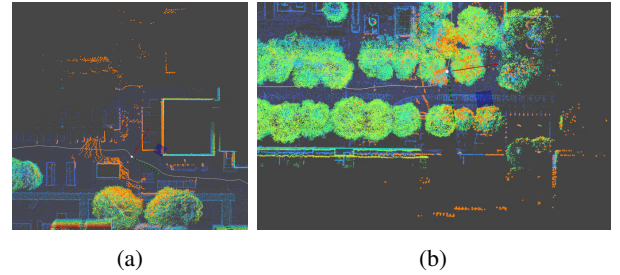


Fig. 10: Places where re-localization took a longer time due to vegetation changes and scan points out of the map.

秒以内に推定を復帰することができたが、二箇所において正解地点に収束するまで比較的長い時間 (15 s, 32 s) がかった。これはスキャン点群が地図範囲外にはみ出ていることや植生の変化によって尤度関数が正解地点で最小化されず、大域姿勢推定が困難となったためと考えられる (Fig. 10)。

4. 結 言

本稿では、GPU 並列計算による SVGD-based パーティクルフィルタを用いた 6 自由度 MCL アルゴリズムを提案した。実験を通し、提案手法が強力な大域姿勢推定性能および推定復帰性能を有することを示した。以下に、提案手法の現時点での制限および今後の研究の方向性について議論する。

計算資源: 現在の実装では点群のみを使った完全な 6 自由度推定を実現するために大量のパーティクルが必要となっており、実時間処理には高性能な GPU (NVIDIA A100) が求められる。しかし、IMU 統合によって重力方向を限定したり、センサのおおよその地面高などの軽微な事前情報を組み入れることで、推定に必要なパーティクル数を大幅に減らすことが可能であり、Jetson

Xavier や Orin などの組み込み GPU での実用的な処理も可能になると見込まれる。

メモリ使用量に関しては、現実装で最もメモリを消費しているのは地図上の最近傍点マップであるが、実験に用いた屋外環境でも 1.6 GB と、現代的な GPU にとっては問題ならない程度の容量にとどまっている。効率的な階層的空間表現⁽¹¹⁾などを用いることで、より大規模な地図へも適用可能になると考えられる。

大域姿勢推定性能: 実験を通して示した通り、大部分の場面において提案手法は極めて強力かつ高速な大域姿勢推定を実現している。一方、Sec. 3.2.2 で言及した通り、尤度関数が正解地点で最小化されないような場合、原理上、大域姿勢推定に長時間かかる、あるいは推定が不可能になる。今後は、各パーティクルにおけるマッチング結果の信頼性評価⁽²¹⁾やウィンドウ最適化による点群の連続性などを考慮することで、さらに安定した大域姿勢推定性能を獲得することを計画している。

参 考 文 献

- (1) D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, “The trimmed iterative closest point algorithm,” in *Object recognition supported by user interaction for service robots*. IEEE, 2002, pp. 545–548.
- (2) M. Magnusson, “The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection,” Ph.D. dissertation, Örebro universitet, 2009.
- (3) W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast direct LiDAR-inertial odometry,” *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- (4) K. Koide, M. Yokozuka, S. Oishi, and A. Banno, “Globally consistent and tightly coupled 3d LiDAR inertial mapping,” in *IEEE International Conference on Robotics and Automation*. IEEE, May 2022.
- (5) D. F. Sebastian Thrun, Wolfram Burgard, *Probabilistic Robotics*. The MIT Press, 2005.
- (6) D. Fox, “Adapting the sample size in particle filters through KLD-sampling,” *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, Dec. 2003.
- (7) G. Grisetti, C. Stachniss, and W. Burgard, “Improving grid-based SLAM with rao-blackwellized particle filters by adaptive proposals and selective resampling,” in *IEEE International Conference on Robotics and Automation*. IEEE, 2005.
- (8) J.-L. Blanco, J. Gonzalez, and J.-A. Fernandez-Madrigal, “An optimal filtering algorithm for non-parametric observation models in robot localization,” in *IEEE International Conference on Robotics and Automation*. IEEE, May 2008.
- (9) J. Saarinen, H. Andreasson, T. Stoyanov, and A. J. Lilienthal, “Normal distributions transform monte-carlo localization (NDT-MCL),” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Nov. 2013.
- (10) F. J. Perez-Grau, F. Caballero, A. Viguria, and A. Ollero, “Multi-sensor three-dimensional monte carlo localization for long-term aerial robot navigation,” *International Journal of Advanced Robotic Systems*, vol. 14, no. 5, Sept. 2017.
- (11) N. Akai, T. Hirayama, and H. Murase, “3d monte carlo localization with efficient distance field representation for automated driving in dynamic environments,” in *IEEE Intelligent Vehicles Symposium*. IEEE, Oct. 2020.
- (12) X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, “PoseRBPF: A rao-blackwellized particle filter for 6-d object pose tracking,” *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1328–1342, Oct. 2021.
- (13) Q. Liu and D. Wang, “Stein variational gradient descent: A general purpose bayesian inference algorithm,” in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.
- (14) F. A. Maken, F. Ramos, and L. Ott, “Stein particle filter for nonlinear, non-gaussian state estimation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5421–5428, Apr. 2022.
- (15) A. Segal, D. Haehnel, and S. Thrun, “Generalized-icp,” in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.
- (16) M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, “Locality-sensitive hashing scheme based on p-stable distributions,” in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, June 2004.
- (17) M. Teschner, B. Heidelberger, M. Müller, D. Pomerantes, and M. H. Gross, “Optimized spatial hashing for collision detection of deformable objects,” in *Vmv*, vol. 3, 2003, pp. 47–54.
- (18) M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- (19) K. Koide, J. Miura, and E. Menegatti, “A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, Mar. 2019.
- (20) Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2018, pp. 7244–7251.
- (21) N. Akai, “Reliable monte carlo localization for mobile robots,” *Journal of Field Robotics*, vol. 40, no. 3, pp. 595–613, Jan. 2023.