

大域マッチングコスト最小化と LiDAR-IMU タイトカップリングに基づく三次元地図生成

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

Globally Consistent and Tightly Coupled 3D LiDAR Inertial Mapping

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

^{*1} Human-Centered Mobility Research Center, National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Ibaraki 305-0061, Japan

This paper presents a real-time 3D mapping framework based on global matching cost minimization and LiDAR-IMU tight coupling. The proposed framework comprises a preprocessing module and three estimation modules: odometry estimation, local mapping, and global mapping, which are all based on the tight coupling of the GPU-accelerated voxelized GICP matching cost factor and the IMU preintegration factor. The odometry estimation module employs a keyframe-based fixed-lag smoothing approach for efficient and low-drift trajectory estimation, with a bounded computation cost. The global mapping module constructs a densely connected factor graph that minimizes the global registration error over the entire map with the support of IMU constraints, ensuring robust optimization in feature-less environments. The evaluation results on several public datasets show that the proposed framework enables accurate and robust localization and mapping in challenging environments.

Key Words : Mapping, Localization, SLAM

1. 緒 言

地図生成 (SLAM) において、長距離センサが移動すると移動量推定に累積誤差 (ドリフト) が生じる。これを補正し、大域的整合性の取れた地図を得ることは地図生成における重要な課題である。LiDAR SLAM において大域軌跡最適化に用いられる手法としてポーズグラフ最適化が挙げられる⁽¹⁾。ポーズグラフ最適化では各時刻におけるセンサ姿勢を推定対象変数とし、その変数間の相対姿勢を制約としたグラフを構築・最適化することによりループクロージングなどを含んだ軌跡修正を行う。近年、多くの LiDAR SLAM 手法が提案されているが、そのほとんどはバックエンドとしてポーズグラフ最適化を採用している^{(2) (3) (4)}。

ポーズグラフでは各相対姿勢制約は固定の正規分布 (6次元の平均・共分散) としてモデル化される。しかし、通常この相対姿勢はスキャンマッチングなどで得られるが、スキャンマッチング結果を正規分布として表現するのは明らかに大きく近似されたアプローチと言える。スキャンマッチングには通常多くの局所解があるが、一度でも局所解にマッチングが収束すると、それを平均として構築された制約はその後大域最適化に悪

影響を与え続けることになる。また、スキャンマッチング結果の信頼性 (共分散) の推定を正確に行うことは実践的には困難である⁽⁵⁾。マッチング結果のヘッセ行列を基にした手法⁽⁶⁾は対応付けによる目的関数の変化を考慮することが出来ないため誤差を過小評価しやすく、一方でモンテカルロベースの手法⁽⁷⁾は高精度に信頼性推定ができるが計算コストが非常に大きい。計算コストと精度のバランスを取るためのデータドリブン手法⁽⁵⁾も提案されているが、多くの LiDAR SLAM 手法では単に固定の共分散行列⁽³⁾、単純な重み付け⁽²⁾、ヘッセ行列を使った手法⁽⁸⁾が共分散の設定に用いられている。これら不正確な制約モデリングにより、ポーズグラフを用いた手法は特に大きなループを閉じた時などに地図上に不整合を引き起こしうる。

より正確に地図の整合性を高める手法として大域マッチングコスト最小化が考えられる。グラフ SLAM の最初期の研究において Lu と Milios らは 2D LiDAR SLAM を大域マッチングコスト最小化 (マルチスキャンレジストレーション) として定式化した⁽⁹⁾。この手法はその後、明示的にループクロージメントを取

^{*1} 産業技術総合研究所 ヒューマンモビリティ研究センター
(〒 305-0061 茨城県つくば市梅園 1-1-1) k.koide@aist.go.jp

補 足 動 画 : <https://staff.aist.go.jp/k.koide/projects/rs2022>

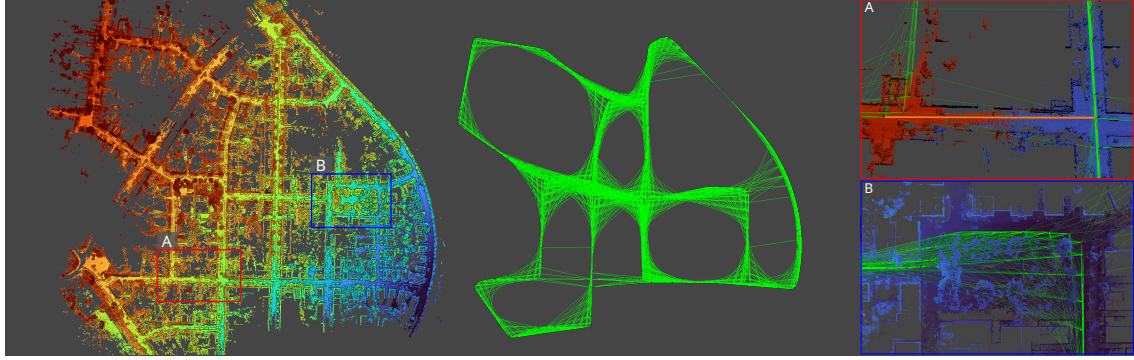


Fig. 1: Environmental map and factor graph generated by the proposed framework (KITTI 00 Sequence).

り扱うことによって3次元SLAMへと拡張された⁽¹⁰⁾⁽¹¹⁾。これらの手法では最適化の各イテレーションにおいて制約の誤差ベクトルとヘッセ行列が現在の推定位置で再評価・更新されるため、固定の平均・共分散行列が用いられるポーズグラフより高精度にフレーム間制約を作ることができる。さらに各制約において明示的に相対姿勢を決定する必要がないため、わずかな重なりしかなく6自由度分の拘束が得られないようなフレーム間にも問題なく制約を作ることができる。しかし、マッチングコストの計算コストは非常に大きく、その理論的優位性に反して大域マッチングコスト最小化は広くは利用されておらず、大規模かつリアルタイムの地図生成タスクへの応用は不可能と考えられてきた。近年提案されたVoxgraphではEuclidean Signed Distance Field表現を用いることにより軽量化されたマッチングコスト項を大域最適化に取り入れている。しかし、依然として計算コストは重たく、ランダムに選ばれたごく一部のマッチング残差(5%)のみを使用している⁽¹²⁾。

本研究では、近年の高速なGPUを最大限活用できるようにスキャンマッチング及びフレームワーク全体を構築することにより、大域マッチングコスト最小化に基づくリアルタイム地図生成フレームワークを実現する。図1に提案手法により生成された環境地図及びファクタグラフを示す。提案手法では従来の相対姿勢制約を排し、マッチングコスト制約を全面的に使用することにより重なり小さなフレーム間にも高精度に拘束をかけ(図1(A))、さらにGPU処理を活用することにより非常に高密度なファクタグラフを生成する(図1(B))。図1では重なり率5%以上の全てのサブマップ間にマッチングコスト制約を生成しており、最適化の各イテレーションにおいて合計4,500以上のファクタのそれぞれにおいて平均20,000点のコスト計算が必要となるが、ミドルクラスGPU(NVIDIA RTX1660Ti)で数秒以内に最適化可能である。

さらに本研究では大域マッチングコスト最小化グラフとIMU preintegration手法⁽¹³⁾を組み合わせることにより、フロントエンドからバックエンドまでの全ステージの最適化をLiDAR-IMUタイトカップリング方式により構築する。これによりジオメトリ特徴が乏しく、LiDAR点群から十分な制約が得られないような環境においても安定して高精度な地図を構築することを可能とする。

本研究のコントリビューションは以下の通りである。

1. GPUを活用することにより大域マッチングコスト最小化がリアルタイムで可能であることと、その精度面における優位性を示す。提案手法はリアルタイム手法としては初めて大域的にマッチングコスト最小化を行うものであり、KITTI DatasetにおいてLiDAR単体手法としては2位の精度を記録している¹。
2. 全最適化ステージにLiDAR-IMUタイトカップリングを採用したフレームワークを提案する。特にLiDAR SLAMにおいてバックエンドにLiDAR-IMUタイトカップリングを採用した例は他にない。
3. 提案手法ではセンサ種類に依存する処理を排しており、回転式LiDARのみならず多様な種類の距離センサに適用可能であることを示す。

2. 提 案 手 法

図2に提案システムの概略を示す。提案システムはプリプロセッシング、オドメトリ推定、ローカルマッピング、グローバルマッピングの4モジュールから成り、全ての推定モジュールはLiDAR-IMUタイトカップリング方式で構築される。オドメトリ推定ではセンサの動きによって生じる点群の歪みを補正しつつロバストにセンサ状態の推定を行う。得られた推定状態は後続のマッピングモジュールへ初期推定値として渡さ

¹GLIM on http://www.cvlibs.net/datasets/kitti/eval_odometry.php

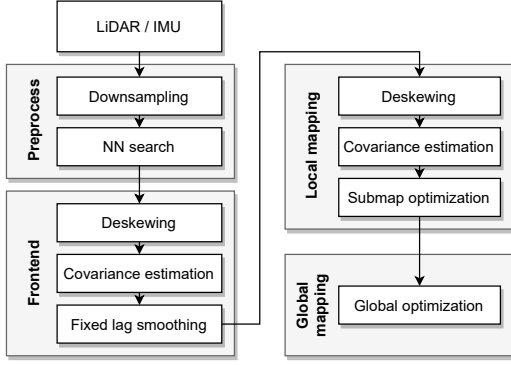


Fig. 2: System overview

れ、さらにリファインされる。ローカルマッピングでは 10~20 フレーム程度の点群を統合したサブマップを生成し、最後段のグローバルマッピングにおいてすべてのサブマップ間のマッチング誤差を最小化することで大域地図最適化を行う。各モジュールはそれぞれ別のスレッド上で非同期に動作する。

以下に推定対象となるセンサ状態 $\mathbf{x}_t$ を定義する。

$$\mathbf{x}_t = [\mathbf{T}_t, \mathbf{v}_t, \mathbf{b}_t]^T. \quad (1)$$

ここで、 $\mathbf{T}_t = [\mathbf{R}_t | \mathbf{t}_t] \in SE(3)$ はセンサ姿勢、 $\mathbf{v}_t \in \mathbb{R}^3$ はセンサ速度、 $\mathbf{b}_t = [\mathbf{b}_t^a, \mathbf{b}_t^\omega] \in \mathbb{R}^6$ は IMU バイアスを表す。LiDAR 点群 $\mathcal{P}_t$ と IMU 観測 (加速度 $\mathbf{a}_t$, 角加速度 $\boldsymbol{\omega}_t$) からセンサ状態 $\mathbf{x}_t$ の時系列推定を行う。なお、計算の効率と簡易化のため、LiDAR 点群は IMU 座標上へ変換し LiDAR・IMU データは同一のセンサ座標系上にあるものとして扱う。

続く 2.1, 2.2 節においてシステムの主要な構成要素となる LiDAR・IMU ファクタについて説明を行い、その後 2.3 節からシステムを構成する各モジュールについて説明を行う。

2.1 LiDAR マッチングコストファクタ 二つのセンサ姿勢 $(\mathbf{T}_i, \mathbf{T}_j)$ に対してそれらの点群間 $(\mathcal{P}_i, \mathcal{P}_j)$ のマッチングコストを制約として与える。ここではマッチングコストとして GPU 処理に適した Voxelized GICP (VGICP) コスト^{(14) (15)}を用いる。VGICP では各入力点 $\mathbf{p}_k \in \mathcal{P}_i$ を正規分布としてモデル化し $(\mathbf{p}_k = (\boldsymbol{\mu}_k, \mathbf{C}_k))$, もう一方の点群 $\mathcal{P}_j$ を解像度 r のグリッドでボクセル化した後、 $\mathcal{P}_i, \mathcal{P}_j$ 間のマッチングコスト e^M を GICP の distribution-to-distribution 誤差関数に基づき以下のように計算する。

$$e^M(\mathcal{P}_i, \mathcal{P}_j, \mathbf{T}_i, \mathbf{T}_j) = \sum_{\mathbf{p}_k \in \mathcal{P}_i} e^{D2D}(\mathbf{p}_k, \mathbf{T}_i^{-1} \mathbf{T}_j), \quad (2)$$

$$e^{D2D}(\mathbf{p}_k, \mathbf{T}_{ij}) = \mathbf{d}_k^T (\mathbf{C}_k' + \mathbf{T}_{ij} \mathbf{C}_k \mathbf{T}_{ij}^T)^{-1} \mathbf{d}_k. \quad (3)$$

ここで、 $\mathbf{p}_k' = (\boldsymbol{\mu}_k', \mathbf{C}_k')$ は $\mathbf{p}_k$ に対応する $\mathcal{P}_j$ のボクセルの平均・共分散、 $\mathbf{d}_k = \boldsymbol{\mu}_k' - \mathbf{T}_{ij} \boldsymbol{\mu}_k$ は対応点間の残差で

ある。VGICP では事前に Kd-Tree など高精度に求められた各点の $\boldsymbol{\mu}_k, \mathbf{C}_k$ の平均を取ることで各ボクセルの分布を得る。遠方の数点しか含まないようなボクセル上でも妥当な分布を得ることができるため、各ボクセルにおいて多数の点を必要とする Normal Distributions Transform より高精度な推定が可能となる⁽¹⁵⁾。

e^{D2D} は重み付き二乗誤差であり、ヤコビアンと残差ベクトルから容易に二次形式制約を得ることができる。これを $\mathcal{P}_i$ の全ての点において積算することによりフレーム $\mathbf{T}_i, \mathbf{T}_j$ を拘束する制約を得る。

2.2 IMU Preintegration ファクタ

IMU 残差を効率的にグラフに組み込むため、IMU preintegration 手法⁽¹³⁾を用いる。IMU 観測 $\mathbf{a}_t, \boldsymbol{\omega}_t$ が与えられたとき、センサ状態は以下のように時間発展する。

$$\mathbf{R}_{t+\Delta t} = \mathbf{R}_t \exp((\boldsymbol{\omega}_t - \mathbf{b}_t^\omega - \boldsymbol{\eta}_t^\omega) \Delta t), \quad (4)$$

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \mathbf{g} \Delta t + \mathbf{R}_t (\mathbf{a}_t - \mathbf{b}_t^a - \boldsymbol{\eta}_t^a) \Delta t, \quad (5)$$

$$\mathbf{t}_{t+\Delta t} = \mathbf{t}_t + \mathbf{v}_t \Delta t + \frac{1}{2} \mathbf{g} \Delta t^2 + \frac{1}{2} \mathbf{R}_t (\mathbf{a}_t - \mathbf{b}_t^a - \boldsymbol{\eta}_t^a) \Delta t^2. \quad (6)$$

$\mathbf{g}$ は重力ベクトル、 $\boldsymbol{\eta}_t^a, \boldsymbol{\eta}_t^\omega$ は IMU 観測上のノイズを表す。IMU preintegration では上式を時刻 i, j 間で積分し、以下の相対運動制約を得る (詳細な導出は⁽¹³⁾参照)。

$$\Delta \mathbf{R}_{ij} = \mathbf{R}_i^T \mathbf{R}_j \exp(\delta \boldsymbol{\phi}_{ij}), \quad (7)$$

$$\Delta \mathbf{v}_{ij} = \mathbf{R}_i^T (\mathbf{v}_j - \mathbf{v}_i - \mathbf{g} \Delta t_{ij}) + \delta \mathbf{v}_{ij}, \quad (8)$$

$$\Delta \mathbf{t}_{ij} = \mathbf{R}_i^T \left(\mathbf{t}_j - \mathbf{t}_i - \mathbf{v}_i \Delta t_{ij} - \frac{1}{2} \mathbf{g} \Delta t_{ij}^2 \right) + \delta \mathbf{t}_{ij}. \quad (9)$$

ただし、 $\delta \boldsymbol{\phi}_{ij}, \delta \mathbf{v}_{ij}, \delta \mathbf{t}_{ij}$ は積分過程のノイズを表す。

ジオメトリ特徴が少なく点群から十分な拘束を得ることができない環境においても、IMU ファクタを導入することによりシステムを十分に拘束することが可能となる。また、同時に重力方向情報がグラフに導入されるため、センサ移動量の推定ドリフトを 4 自由度に抑えることができる⁽¹⁶⁾。

2.3 プリプロセッシング

まず入力された各点群に対してボクセルグリッドフィルタによってダウンサンプリングを行う。ここでは各点の位置に加えて、後続の歪み補正処理に必要な各点の時刻もフィルタリングの対象とする。時刻の大きく異なる点 (e.g., スキャン開始点と終了点) をボクセル上で混合してしまうを防ぐため、ある点の時刻が対応するボクセルの時刻と大きく異なる場合には新たなボクセルを生成し、それに点を割り当てる。その後、Kd-Tree によって後に各点の共分散行列推定に必要な k 近傍点を求める。共分散行列推定は歪み補正処理の後に実行されるが、歪み補正では点の近傍関係に大きな変化は起きな

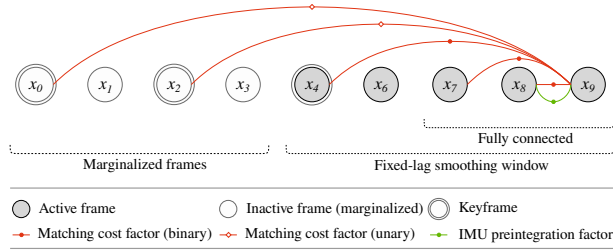


Fig. 3: Frontend factor graph. Only the factors related to the latest frame (x_9) are illustrated.

いと仮定し、この前処理段階で得られた近傍点を後続の処理ではそのまま用いる。

2.4 オドメトリ推定 オドメトリ推定ではセンサの急激な動きを補償し、頑強かつドリフトの少ない移動量推定を行う。まず IMU による動き予測を基に点群をスキャン開始時刻の IMU 座標へ変換することで、動きにより生じた点群上の歪みを補正する。その後、事前に求めておいた k 近傍点より各点の共分散行列を算出する。

図 3 にオドメトリ推定に用いるファクタグラフを示す。可視性のため図には最新のフレームに関連するファクタのみ示す。最適化の計算量を一定に保つため、Fixed-lag smoothing アプローチを用い、古い状態は周辺化することによりグラフから取り除く。ここでは効率的に低ドリフトな推定を実現するために、Direct Sparse Odometry (DSO)⁽¹⁷⁾ に似たキーフレームベースの推定手法を提案する。キーフレームは最新のフレームと一定の重なりを持ちつつ、空間上に広く分布するように選択されたフレームの集合であり、最新のフレームと全てのキーフレームの間にマッチングコストファクタを生成することで推定ドリフトを抑制する。キーフレームが Fixed-lag smoothing の最適化ウィンドウ内にまだ存在していればフレームとキーフレームの双方に作用する二項ファクタを生成し、すでにキーフレームが周辺化されていた場合にはキーフレーム姿勢が固定されたと考え、フレームの姿勢のみに作用する単項ファクタを生成する。

キーフレーム管理のために 2 つの点群 $\mathcal{P}_i, \mathcal{P}_j$ 間の重なり率を $\mathcal{P}_i$ の点のうち $\mathcal{P}_j$ に対応するボクセルが存在するものの割合と定義する。現在のフレームと最新のキーフレーム間の重なり率が一定以下 (e.g., 90%) となった時、そのフレームを新たにキーフレームリストに追加する。その後、キーフレーム数が一定以上 (e.g., 20) になった場合、以下のスコア関数を最小化するキーフレームをリストから取り除く。

$$s(i) = o(i, N^{odom}) \sum_{j \in [1, N^{odom}-1] \setminus \{i\}} (1 - o(i, j)). \quad (10)$$

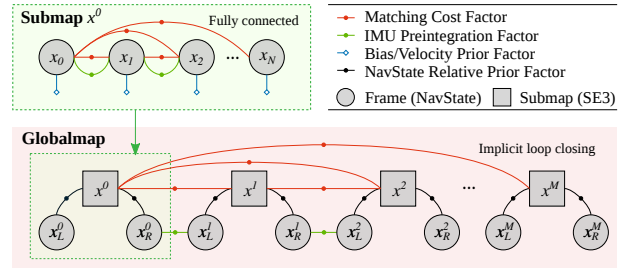


Fig. 4: Backend factor graph.

$o(i, j)$ は i 番目と j 番目のキーフレーム間の重なり率を表す。上式は DSO⁽¹⁷⁾ のキーフレーム管理方式を参考にヒューリスティックに設計された関数で、最新フレーム付近に一定数のキーフレームを保持しつつ、キーフレームが空間的に広く分布するようなスコアを与える。

最新フレーム・キーフレーム間に加え、過去数フレーム全て (e.g., 3 フレーム) との間にもマッチングコストファクタを生成し急激な動きへのロバスト性を高める。さらに、連続するフレーム間には IMU ファクタを生成することで LiDAR データから十分な拘束が得られない環境においても安定した移動量推定を実現する。

2.5 ローカルマッピング オドメトリ推定グラフから周辺化されたフレームは、センサ状態の初期推定値としてローカルマッピングへ渡される。ローカルマッピングでは、後段のグローバルマッピングで推定対象となる変数の数を削減するために、一定数のフレームを統合し一つのサブマップを生成する。

まず、オドメトリ推定の最適化を経て推定されたセンサ状態が改善されていることが期待できるため、改めて現在の推定状態で点群の歪み補正と共分散行列推定を行う。その後、現在のフレームとサブマップ内の最後のフレームの重なり率を評価し、それが一定以下 (e.g., 90%) であればセンサに動きがあったとしてサブマップにそのフレームを追加する。

図 4 に示すように、ローカルマッピングではサブマップ内のすべてのフレーム間にマッチングコストファクタを生成する (i.e., all-to-all レジストレーション)。更に連続するフレーム間には IMU ファクタを挿入し、最適化を安定化させるためにオドメトリ推定で得られたセンサ速度・IMU バイアスの推定値を Prior ファクタとして挿入する。

サブマップ内のフレーム数が一定数を越えた時 (e.g., 20 フレーム)、Levenberg-Marquardt 法を用いグラフ最適化を行い、最適化されたセンサ姿勢を基に点群を統合しサブマップを生成する。サブマップ生成の時間間隔における推定ドリフトは小さく無視できるものと仮定し、後続のグローバルマッピングではサブマップ内

のフレーム間の相対姿勢は固定されたものとして扱う。

2.6 グローバルマッピング グローバルマッピングではサブマップ間の姿勢を最適化することで、大域的一貫性を保った地図を生成する。サブマップが生成される度に過去の全てのサブマップとの重なり率を評価し、それが小さく設定されたしきい値 (e.g., 5%) 以上であればサブマップ間にマッチングコストファクタを生成する。提案手法では僅かな重なりしかないサブマップ間にも極めてアグレッシブに制約を生成するため、図 1 のような非常に高密度なグラフが得られる。これによりすべてのサブマップはグラフ上で隣接するサブマップだけでなく位置的に近傍の全てのサブマップに対してもアラインされるため、ある地点を再訪した際には非明示的にループが閉じられ推定ドリフトを補正することができる。従来のポーズグラフでは制約を作る際に明示的な相対姿勢が必要となるため、スキャンマッチングが不安定になる重なり of the small フレーム間には高精度に制約を作ることが困難であった。提案手法では各ファクタにおいてはスキャンマッチングの最適化を行わず、地図全体を一つのシステムとして最適化するため、十分な拘束が得られない極めて小さな重なりしかないフレーム間にも高精度に制約を作ることができる。

さらに最適化の安定性を高めるため、連続するサブマップ間に IMU ファクタを生成する。しかし、サブマップは大きな時間間隔 (e.g., 10 s) で生成されるため、単純にサブマップ姿勢間に IMU ファクタを生成すると、IMU ファクタの共分散が大きくなりすぎてしまいシステムを強く拘束することができない⁽¹⁸⁾。さらに前段の推定で得られた速度や IMU バイアス情報を捨てることにもなってしまう。ここではセンサ速度・バイアス情報を保持しつつ強くサブマップ姿勢を拘束するため、各サブマップにつき *endpoints* と呼ぶ 2 つのセンサ状態変数 $\mathbf{x}_L^i, \mathbf{x}_R^i$ を導入し、それぞれにサブマップ内の最初と最後のフレームのセンサ状態推定値を保持させる (図 4 参照)。

サブマップ $\mathbf{x}^i$ 内のセンサ状態推定結果を $[\mathbf{x}_0, \dots, \mathbf{x}_{N_{sub}}]$ とし、サブマップ原点を $\mathbf{T}^i = \mathbf{T}_{N_{sub}/2}$ と定義した時、あるセンサ状態 $\mathbf{x}_t$ のサブマップ原点に対する相対状態は以下のように与えられる。

$$\mathbf{T}_t' = (\mathbf{T}^i)^{-1} \mathbf{T}_t, \quad (11)$$

$$\mathbf{v}_t' = (\mathbf{R}^i)^{-1} \mathbf{v}_t, \quad (12)$$

$$\mathbf{b}_t' = \mathbf{b}_t. \quad (13)$$

ここで、サブマップ姿勢 $\mathbf{x}^i$ と endpoints $\mathbf{x}_L^i, \mathbf{x}_R^i$ の間に式 (11)(12)(13) を満たすように相対状態ファクタを生

成する。これにより $\mathbf{x}_L^i, \mathbf{x}_R^i$ には現在のサブマップ姿勢と事前の推定量を基に世界座標系におけるサブマップ開始時・終了時のセンサ状態が保持される。あるサブマップの $\mathbf{x}_R^i$ とその次のサブマップの $\mathbf{x}_L^{i+1}$ の間隔は最小 (スキャン間隔) になるため、この間に IMU ファクタを生成することで推定情報の損失を抑えつつサブマップ姿勢を強く拘束することが可能となる。

一定数のサブマップ (e.g., 5) が挿入されるたび、iSAM2⁽¹⁹⁾を用いインクリメンタルにグラフを最適化する。ここでは GTSAM の実装を用いた。

3. 実験

3.1 LiDAR 単体による評価 大域マッチングコスト最小化の精度面での優位性を示すため、まず LiDAR のみを用いた条件で KITTI データセット上で評価を行った。IMU データは使用しないため、提案手法の LiDAR-IMU フロントエンドの代わりに MULLS⁽⁴⁾によるオドメトリ推定を行い、その後、IMU ファクタを除いたローカル・グローバルマッピングによってセンサ移動軌跡の推定を行った。また、大きな推定ドリフトを補正するために ScanContext⁽²²⁾によるループ検出結果を大域最適化に追加した。精度評価には Relative Trajectory Error (RTE)⁽²³⁾を用いた。

State-of-the-art との比較: 比較対象として複数の LiDAR SLAM 手法 (LOAM⁽²⁴⁾, MULLS⁽⁴⁾, IMLS-SLAM⁽²⁵⁾, SuMa++⁽³⁾), と深層学習手法 (LO-Net⁽²¹⁾)を用いた。比較のため提案手法は 1) ループクローズなし (スライディングウィンドウ内のサブマップ同士間のみマッチングコストファクタを生成), 2) ループクローズあり の 2 つの設定で評価を行った。

表 1, 2 に精度評価結果を示す。なお、表内の Mean ST は RTE のサブトラジェクトリ誤差の平均, Mean S はシーケンス RTE の平均を示している。ループクローズなしの設定において、提案手法はシーケンス 00-10 で最も良い推定精度を示した (0.52 m / 0.14°)。特に移動距離の長いシーケンス (00, 02, 05, 08) における推定精度が良く、提案手法の高密度なグラフ生成が大規模環境における推定誤差低減に寄与していることが確認できる。また、シーケンス 11-21 においても非常に高精度な推定 (0.59 m / 0.15°) が得られており、これは KITTI リーダーボード上で LiDAR のみを使った手法としては 2 位の結果となっている。ループクローズを有効にした場合には Rotational RTE が大きく改善されていることが確認できる (0.11°)。一方、Translational RTE はわずかに悪化しているが (0.56 m), 同様の傾向は他の研究でも報告されており⁽⁴⁾, KITTI の点群に含まれる歪みが原因だと考えられる。

Table 1: Average rotational relative trajectory errors (RTEs) [°/100m] on the KITTI dataset

Sequence Num. Num. of Frames	Loop closure	00 4541	01 1101	02 4661	03 801	04 271	05 2761	06 1101	07 1101	08 4071	09 1591	10 1201	00-10 Mean (ST/S)	11-21 Mean (ST)
Proposed (matching cost)		0.16	0.10	0.12	0.19	0.10	0.10	0.07	0.11	0.18	0.11	0.15	0.14 / 0.13	0.15
Proposed (matching cost)	✓	0.12	0.09	0.10	0.19	0.10	0.06	0.08	0.10	0.14	0.08	0.15	0.11 / 0.11	-
Proposed (SE3)	✓	0.18	0.15	0.17	0.33	0.17	0.21	0.10	0.17	0.50	0.17	0.31	0.24 / 0.22	-
LOAM ⁽²⁰⁾		-	-	-	-	-	-	-	-	-	-	-	- / -	0.13
MULLS ⁽⁴⁾		0.18	0.09	0.17	0.22	0.08	0.17	0.11	0.18	0.25	0.15	0.19	- / 0.16	0.19
MULLS ⁽⁴⁾	✓	0.13	0.09	0.13	0.22	0.08	0.07	0.08	0.11	0.17	0.12	0.19	- / 0.13	-
SuMa++ ⁽³⁾	✓	0.22	0.46	0.37	0.46	0.26	0.20	0.21	0.19	0.35	0.23	0.28	0.29 / 0.29	0.34
LO-Net ⁽²¹⁾		0.42	0.40	0.45	0.59	0.54	0.35	0.33	0.45	0.43	0.38	0.41	- / 0.43	-

Red and **blue** respectively indicate the first and second best results.

Mean ST and S respectively indicate the means of sub-trajectory and sequence errors.

Table 2: Average translational relative trajectory errors (RTEs) [m/100m] on the KITTI dataset

Sequence Num. Num. of Frames	Loop closure	00 4541	01 1101	02 4661	03 801	04 271	05 2761	06 1101	07 1101	08 4071	09 1591	10 1201	00-10 Mean (ST/S)	11-21 Mean (ST)
Proposed (Matching Cost)		0.49	0.65	0.50	0.62	0.41	0.24	0.29	0.30	0.80	0.46	0.54	0.52 / 0.48	0.59
Proposed (Matching Cost)	✓	0.56	0.66	0.55	0.63	0.42	0.28	0.34	0.35	0.81	0.55	0.54	0.56 / 0.52	-
Proposed (SE3)	✓	0.58	0.61	0.60	0.69	0.44	0.38	0.34	0.37	1.51	0.68	0.74	0.74 / 0.63	-
LOAM ⁽²⁰⁾		0.78	1.43	0.92	0.86	0.71	0.57	0.65	0.63	1.12	0.77	0.79	- / 0.84	0.55
MULLS ⁽⁴⁾		0.51	0.62	0.55	0.61	0.35	0.28	0.24	0.29	0.80	0.49	0.61	- / 0.49	0.65
MULLS ⁽⁴⁾	✓	0.54	0.62	0.69	0.61	0.35	0.29	0.29	0.27	0.83	0.51	0.61	- / 0.52	-
SuMa++ ⁽³⁾	✓	0.64	1.60	1.00	0.67	0.37	0.40	0.46	0.34	1.10	0.47	0.66	0.70 / 0.70	1.06
LO-Net ⁽²¹⁾		0.78	1.42	1.01	0.73	0.56	0.62	0.55	0.56	1.08	0.77	0.92	- / 0.82	-

Red and **blue** respectively indicate the first and second best results.

Mean ST and S respectively indicate the means of sub-trajectory and sequence errors.

Table 3: Relative and absolute trajectory errors on the Newer College dataset

Metric	LIOM (odom)	LIO-SAM (odom)	LIO-SAM	Proposed (odom)	Proposed
RTE [m]	2.224 ± 1.402	2.215 ± 1.376	2.156 ± 1.357	2.140 ± 1.348	2.160 ± 1.356
ATE [m]	3.392 ± 1.653	1.176 ± 0.641	0.529 ± 0.259	0.899 ± 0.595	0.276 ± 0.093

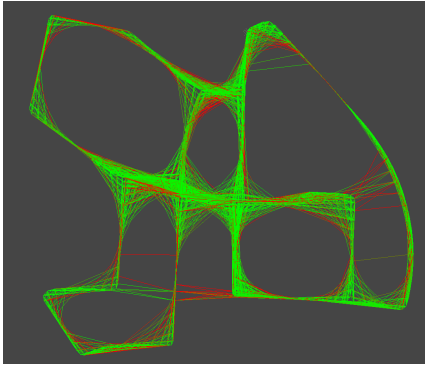


Fig. 5: Factor graph with SE3 relative pose factors. The line color indicates the magnitude of the error (Red: large error, Green: small error).

Ablation study: 従来のポーズグラフとの比較として、全てのマッチングコストファクタを相対姿勢(SE3)ファクタに置き換えたグラフで最適化を行った。提案手法で得られたセンサ姿勢を初期値として各ファクタにおいてGICP スキャンマッチングを行い、情報行列はマッチング結果のヘッセ行列から与えた⁽⁶⁾。スキャ

ンマッチングが失敗することを考慮し各ファクタにはHuber ロバストカーネルを適用した。

表 1, 2 から、グラフ構造(サブマップ接続関係)が変化していないにも関わらず、ポーズグラフによる推定精度は大きく劣化していることが確認できる。図 5 に KITTI 00 におけるポーズグラフを示す。線の色は各ファクタの推定値からの誤差の大きさを表している(緑: 誤差小, 赤: 誤差大)。特に距離の離れたサブマップ間のファクタに大きな誤差が見られ、重なりが小さいサブマップ間でスキャンマッチングが失敗したことが見て取れる。これらのファクタはロバストカーネルによって除去されたため最終的な最適化結果には寄与せず、結果として精度の悪化を招いたと考えられる。

なお、実際の運用においては今回の評価のような良い初期値は期待できないため、より多くのファクタが精度の悪い相対姿勢制約を持つことになる。この比較結果から、大域マッチングコスト最小化手法の従来のポーズグラフ手法に対する優位性が示された。

3.2 LiDAR-IMU システム評価 Newer College Dataset の *long_experiment* シーケンス上で LiDAR-IMU

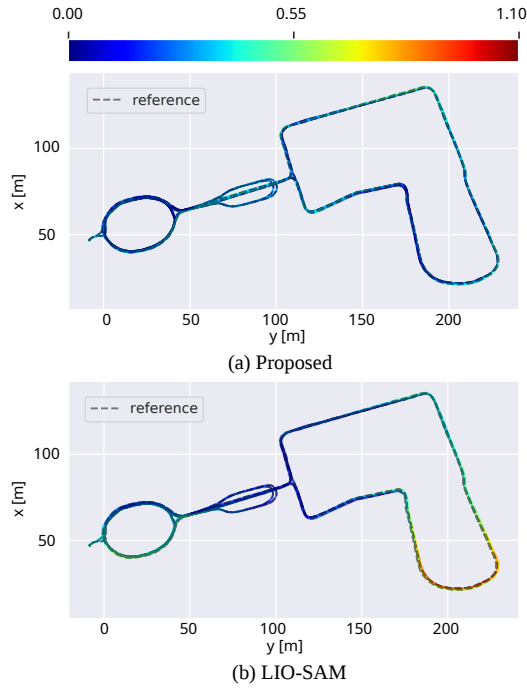


Fig. 6: Estimated sensor trajectories for the *long_experiment* sequence. The color indicates the magnitude of the ATE.

地図生成の評価を行った。比較対象としては state-of-the-art の LIO-mapping (LIOM)⁽²⁶⁾と LIO-SAM⁽²⁾を用いた。表 3 に各手法の Absolute/Relative Trajectory Error (ATE / RTE) を示す。提案手法のオドメトリ推定は LIO-mapping, LIO-SAM に対して優位な RTE を示しており (Proposed: 2.140 m, LIOM: 2.224 m, LIO-SAM: 2.215 m), 低ドリフトな推定が実現できていることが確認できる。また、ループクローズを有効にした場合、従来のポーズグラフを用いている LIO-SAM に対して提案手法は大きく良い ATE を示している (Proposed: 0.276 m, LIO-SAM: 0.529 m)。

図 6 に提案手法及び LIO-SAM によって推定されたセンサ軌跡を示す。軌跡の色はその地点における ATE の大きさを表している。LIO-SAM は大きなカーブにおいて推定精度を落としており、これは比較的フレームの密度が疎になった際にポーズグラフでは正確な制約が得られなかったのと、大域最適化に IMU 制約が組み込まれていないため重力方向情報が失われた結果であると考えられる。一方、提案手法では、高密度なグラフ生成と IMU 制約を大域最適化に組み込んでいることにより、カーブ地点においても高精度な軌跡推定ができています。

3.3 処理速度・他の LiDAR への適用 処理速度及びより困難な環境でのデモンストレーションとして、KAIST Urban Dataset 上で地図生成を行った。このデー

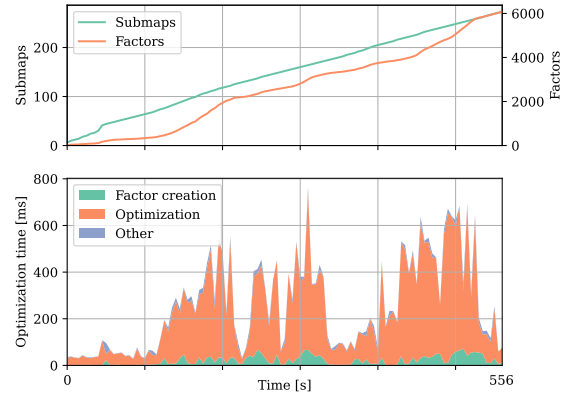


Fig. 7: Number of submaps and matching cost factors and the processing time of the global optimization through the KAIST07 sequence.

タセットでは LiDAR が垂直に取り付けられているため、車両が移動すると点群間の重なりが小さくなり、通常の LiDAR のみを用いた手法では対処することが難しいものとなっている。ここでは KAIST07, 17 シーケンスに対して提案手法を適用し、安定して地図生成可能であることを示した。(紙面上の制約のため詳細は割愛する。詳細は補足動画を参照頂きたい。)

KAIST07 シーケンスを通して提案手法はリアルタイムの 2 倍速で動作した。各モジュールの平均処理時間はプリプロセッシング: 23.3 [ms / frame], オドメトリ推定: 41.3 [ms / frame], サブマッピング: 122.0 [ms / 20 frames], グローバルマッピング: 242.5 [ms / 50 frames] であり、リアルタイム要求を十分に上回る速度で動作することが確認できた (センサ取得周期 10Hz)。図 7 に姿勢変数・マッチングコストファクタの増加に対するグローバルマッピングの最適化時間の伸びを示す。最終的には 6000 以上のマッチングコストファクタが生成されているが、GPU を用いたコスト評価とインクリメンタル最適化によって一秒未満で大域最適化が完了していることが確認できる。

また、提案手法は他の LiDAR 地図生成手法^{(2) (26) (3)}と異なり、センサ依存の特徴量抽出処理などを排しており、回転型以外の任意の種類の LiDAR にも適用することが可能である。実証として行った Livox Avia 及び Intel Realsense L515 を用いた環境地図生成の様子を補足動画において確認することができる。

4. 結 言

GPU を用いたマッチングコストファクタと IMU preintegration ファクタを組み合わせることにより、高精度で頑強な LiDAR-IMU 地図生成を提案した。精度及びリアルタイム性能を複数のデータセット上で評価

し、従来手法に対しての優位性を示した。現在は提案手法の発展版をオープンソースとして公開するための開発を進めている。

参 考 文 献

- (1) G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, Dec. 2010.
- (2) T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2020, pp. 5135–5142.
- (3) X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Nov. 2019.
- (4) Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, "MULLS: Versatile lidar slam via multi-metric linear least square," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2021.
- (5) D. Landry, F. Pomerleau, and P. Giguere, "CELLO-3D: Estimating the covariance of ICP in the real world," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2019.
- (6) O. Bengtsson and A.-J. Baerveldt, "Robot localization based on scan-matching—estimating the covariance matrix for the IDC algorithm," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 29–40, July 2003.
- (7) T. M. Iversen, A. G. Buch, and D. Kraft, "Prediction of ICP pose uncertainties using monte carlo simulation with synthetic depth images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Sept. 2017.
- (8) W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2016.
- (9) F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, Oct. 1997.
- (10) D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3D mapping with scan matching," *Robotics and Autonomous Systems*, vol. 56, no. 2, pp. 130–142, Feb. 2008.
- (11) J. Sprickerhof, A. Nüchter, K. Lingemann, and J. Hertzberg, "A heuristic loop closing technique for large-scale 6D SLAM," *Automatika*, vol. 52, no. 3, pp. 199–222, Jan. 2011.
- (12) V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 227–234, Jan. 2020.
- (13) 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, Feb. 2017.
- (14) A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems V*. Robotics: Science and Systems Foundation, June 2009, pp. 435–443.
- (15) K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2021.
- (16) 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, Aug. 2018.
- (17) J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, Mar. 2018.
- (18) L. V. Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2018, pp. 2510–2517.
- (19) M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the bayes tree," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, Dec. 2011.
- (20) J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, July 2014.
- (21) Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li, "LO-net: Deep real-time lidar odometry," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2019.
- (22) G. Kim and A. Kim, "Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2018.
- (23) 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.
- (24) J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, Feb. 2016.
- (25) J.-E. Deschaud, "IMLS-SLAM: Scan-to-model matching based on 3d data," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2018.
- (26) H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D lidar inertial odometry and mapping," in *IEEE International Conference on Robotics and Automation*. IEEE, May 2019, pp. 3144–3150.