

画像特徴点の対応を制約とした 疎な LiDAR 点群からの Odometry 推定*

—ガウス過程回帰による深度推定とその信頼度を考慮した Odometry 推定—

清水政行** 川西康友** 出口大輔** 井手一郎** 村瀬 洋**

Odometry Estimation from Sparse LiDAR Point Cloud Constrained by Image Feature Correspondence
—Depth Estimation by Gaussian Process Regression and Odometry Estimation Considering Its Reliability—

Masayuki SHIMIZU, Yasutomo KAWANISHI, Daisuke DEGUCHI, Ichiro IDE and Hiroshi MURASE

Estimation of translation between consecutive frames, i.e., odometry, plays an important role in autonomous navigation. This paper presents an odometry estimation method using sparse LiDAR points and image feature points. In case of sparse LiDAR measurements, it is difficult to accurately estimate depth at image feature points. Image feature points with low-accuracy depth cause misconvergence in odometry optimization. To improve the robustness to the misconvergence, a new method with a Gaussian process that estimates not only the depth at image feature points but also the variance is proposed. By using this variance, it estimates the residual of image features in the world coordinate with depth, or in the image coordinate without depth. This allows more accurate and robust estimation than conventional methods in case of sparse LiDAR points. In an experiment with simulated sparse LiDAR points from the KITTI dataset, the proposed method is confirmed to estimate the odometry more accurately than conventional methods.

Key words: odometry estimation, image features, sparse LiDAR points, depth estimation, Gaussian process regression

1. 緒 言

自動運転車や自律移動ロボットが、あらかじめ設定された経路上を走行するためには、正確な自己位置を知ることが重要である。高精度な自己位置推定を行うためには、高精度な地図の生成が必要である。そのための要素技術として、自身の周囲をセンサで観測し、その観測結果に基づいて自己位置推定と地図生成を同時に行う Simultaneous Localization and Mapping (SLAM) 技術に関して多くの研究成果が発表されている^{1)~7)}。

最初期の SLAM 技術の研究では、LASER を走査して物体までの距離と走査角度を計測する、Light Detection and Ranging (LiDAR) 装置を用いた、LiDAR SLAM が提案された^{7)~11)}。これは、走査線数が 1 本の LiDAR を用いて平面上の位置を推定する、2 次元 SLAM である。自律移動ロボットは屋内で活用されることが多く、平面上を運動するため、実用上は、2 次元 SLAM で十分であった。一方、自動運転車の用途では、道路に勾配がある環境でも自己位置推定と地図生成を実現する必要がある。そのため複数本の走査線により、3 次元点群を計測できる LiDAR と、これを用いた 3 次元 LiDAR SLAM 技術が提案された¹⁾。

3 次元の LiDAR SLAM として代表的な LiDAR Odometry and Mapping (LOAM)¹⁾ は、連続する 2 フレーム間のセンサの移動量 (odometry) を推定する処理と、地図座標系上でセンサの位置を推定 (localization) して、地図を生成 (mapping) する処理から構成されている。Localization では、odometry を初期

値として最適化問題を解くことで推定が行われる。Odometry の推定誤差が大きいと、真値から大きく外れた初期値を用いて localization するため、推定結果が大域的最適解とは大きく異なる局所解に収束することがある。これを避けるためには、高精度な odometry 推定が必要である。

LiDAR による odometry 推定は、連続する 2 フレーム間の LiDAR 点群から LiDAR 点の対応付けを行い、対応付けされた LiDAR 点の距離を最小にするように最適化問題を解くことにより行われる。ここで、LiDAR により高い位置精度の 3 次元点群を得られるが、LiDAR 点の対応付けには曖昧性があり、誤りも多く含まれる。一方、カメラ画像から得られる特徴点は、局所特微量で対応付けできるため、正しく対応付けできる。高精度な odometry 推定のために、高い位置精度をもつ LiDAR 点と、正確に対応付けできる画像特徴点を組み合わせた手法が提案されている^{12) 13)}。

近年では、LiDAR 装置の小型化と低価格化が進み、急速に普及しつつある。しかし、密な LiDAR 点群を取得するための LASER 走査機構は大きさが大きく、価格も高くなる。そのため、密な LiDAR 点群の取得と装置の小型化・低価格化は両立しにくい。LiDAR の構造上スキャンラインを減らすことは小型化と低コスト化に繋がるため、本論文ではスキャンラインが疎な LiDAR とカメラによる高精度な odometry 推定について検討する。

疎な LiDAR 点群と画像から得られる特徴点群を組み合わせることで odometry を推定する場合、次の 2 つの課題がある。

1. LiDAR 点と画像特徴点を組み合わせて odometry を推定するためには、カメラから各特徴点までの距離 (深度) が必要になる。Odometry 推定の最適化において、深度推定

* 原稿受付 令和 2 年 10 月 27 日

掲載決定 令和 3 年 1 月 29 日

** 名古屋大学 (愛知県名古屋市千種区不老町)

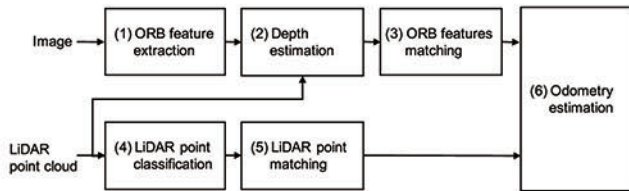


Fig. 1 Overview of the visually constrained sparse LiDAR odometry estimation

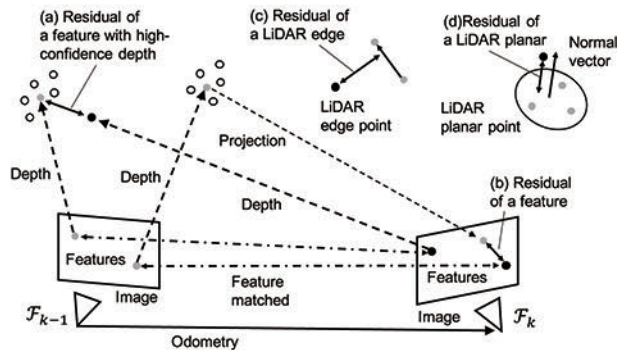


Fig. 2 Concept of the optimization of odometry estimation

の誤差が大きい特徴点が存在すると、推定精度が低下するという問題がある。

2. 各特徴点の深度は、画像平面上で特徴点に最も近い LiDAR 点から推定される。近傍に LiDAR 点が存在しない特徴点では、高精度に深度を推定することが難しい。特に LiDAR 点群が疎な場合には、このような特徴点が多数存在するという問題がある。

本論文は、これらの問題を解決した高精度な odometry 推定手法を提案する。提案手法は、具体的に次の 2 つの特徴を有する。

1. 特徴点の深度推定の信頼度に応じて、odometry を推定する最適化問題の目的関数を切り換える。目的関数は、対応付けされた特徴点間の距離と、対応付けされた LiDAR 点間の距離に基づいて定義する。対応付けされた特徴点間の距離は、深度推定の信頼度が高い場合、深度を用いて 3 次元座標系に変換して算出する。一方、信頼度が低い場合は、画像座標系で算出する。
2. 特徴点の深度と深度推定の信頼度は、疎な LiDAR 点群を入力として、ガウス過程回帰を用いて推定を行う。ガウス過程回帰は、深度の推定値が取りうる範囲をガウス分布として、その平均と分散を推定する。深度推定の信頼度は、ここで求めた深度の推定値分布の分散の逆数で定義する。

提案手法は信頼度が低い深度を用いずに odometry 推定の最適化を行う。このようにすることによって、深度推定の誤差に対して頑健で高精度な odometry 推定を実現することができる。

図 1 に提案手法の全体構成を示す。本手法は、(1) 特徴点と特徴量の抽出、(2) ガウス過程回帰による特徴点の深度推定、(3) 特徴点の対応付け、(4) LiDAR 点群の分類、(5) LiDAR 点の対応付け、(6) odometry 推定から構成される。ここで、(4) と (5) については LOAM と同じ手法を用いる。図 2 に (6) odometry 推定の概要を示す。図 2 の (a) と (b) は対応付けさ

れた画像特徴点の距離を示す。画像特徴点の距離はガウス過程回帰で算出される深度推定の信頼度によってどちらか一方が選択される。また、図 2 の (c) と (d) は対応付けされた LiDAR 点の距離を示し、LOAM と同じ手法で算出する。LiDAR 点と画像特徴点の距離を組み合わせた目的関数を定義して、これを最小にする odometry を推定する。

本論文の貢献は以下の通りである。

1. 画像特徴点において、深度の推定値が取りうる範囲をガウス分布と仮定して平均と分散を推定する手法の提案
2. 深度推定の分散に応じて、特徴点の目的関数を切り替える odometry 推定手法の提案

以降、2 章で関連研究について述べる。本論文は、LOAM とカメラを組み合わせた odometry 推定手法を提案するため、まず 3 章で LOAM による odometry 推定手法について紹介する。次に 4 章で、疎な LiDAR 点群から画像特徴点の深度を推定して odometry を推定する提案手法について詳細を説明する。5 章で評価実験について報告するとともに、6 章で考察を加える。最後に 7 章で本論文をまとめる。

2. 関連研究

LiDAR を用いた odometry 推定、localization や mapping について多くの研究成果が報告されている^{1) 7) 14) 15)}。LiDAR による odometry 推定は Iterative Closest Point (ICP) 法¹⁶⁾がよく利用されている。ICP 法は、2 つの点群を重ね合わせるためのアルゴリズムで、点の対応付けと、対応付けされた点から 2 つの点群を重ね合わせるために必要な移動量を算出する処理から構成される。

LOAM¹⁾ は ICP 法と、Velodyne 社¹⁷⁾ の LiDAR (以降、Velodyne) 等を用いた代表的な 3 次元の LiDAR SLAM である。LOAM は、Velodyne の走査形状の特性から LiDAR 点をコーナと面に分類する。分類結果に応じて最適化問題の目的関数を定義し odometry を推定する。しかし、ICP 法に基づく手法は、LiDAR の走査形状 (例えば Velodyne はリング形状) に起因する LiDAR 点の誤対応付けが原因で、odometry の推定精度が低下する場合がある^{12) 14)}。

LiDAR を用いずに、カメラを用いた Visual SLAM が提案されている。Visual SLAM は主に、特徴点の対応付け結果から odometry を推定する手法⁴⁾と、画像間の画素値の差分から目的関数を定義して odometry を推定する手法^{3) 18)}に大別できる。特徴点を利用する手法は、連続する 2 フレーム間で対応付けられた特徴点と、特徴点の深度から odometry を推定する。一方で、画像間の画素値を利用する手法は 2 フレーム間の画素値の差分が最小になるような odometry を推定する。特徴点が抽出しにくい環境でも odometry 推定が可能で、また、密な点群を得られる特徴がある。画像間の画素値を利用する手法は、画像間で画素値の差分を計算する時に深度が必要になる。

単眼カメラの SLAM^{3) 4) 18)} は、推定した odometry から深度を推定する。しかし、odometry と深度のスケールを一意に決定できない問題がある¹⁹⁾。画像と深度を同時に取得することが可能なステレオカメラによる SLAM²⁰⁾ や、RGB-D カメラによる SLAM⁵⁾ が実現されている。RGB-D カメラは深度の測距精度が高いが、検知距離が短く広い空間では深度を計測できない問題がある。

高精度に odometry を推定するために、高い位置精度をもつ

Table 1 List of variables

Variable	Definition
\mathcal{I}_k	k -th RGB image
\mathcal{P}_k	k -th LiDAR point cloud
\mathcal{E}_k	Edge point cloud from \mathcal{P}_k
\mathcal{H}_k	Planar point cloud from \mathcal{P}_k
\mathcal{O}_k	ORB feature point cloud from \mathcal{I}_k
$\mathbf{p}_{k,i}$	i -th LiDAR point in \mathcal{P}_k , $\mathbf{p}_{k,i} \in \mathbb{R}^3$
$\mathbf{e}_{k,i}$	i -th edge point in \mathcal{E}_k , $\mathbf{e}_{k,i} \in \mathbb{R}^3$
$\mathbf{h}_{k,i}$	i -th planar point in \mathcal{H}_k , $\mathbf{h}_{k,i} \in \mathbb{R}^3$
$\mathbf{o}_{k,i}$	i -th ORB feature point in \mathcal{O}_k , $\mathbf{o}_{k,i} \in \mathbb{R}^2$
$\mathbf{l}_{o_{k,i}}$	Landmark for i -th ORB feature point in \mathcal{O}_k , $\mathbf{l}_{o_{k,i}} \in \mathbb{R}^3$
\mathbf{F}_k	k -th camera pose
\mathbf{T}_k^{k+1}	Relative pose transformation from \mathbf{F}_k to \mathbf{F}_{k+1}
$\mathcal{T}(\mathbf{p}_{k,i})$	Camera projection function

LiDAR 点と、正確に対応付けできる画像特徴点を組み合わせる手法が提案されている^{12) 13) 21)}。Zhen らはステレオカメラと LiDAR を合わせて odometry を推定する手法を提案している¹²⁾。この手法では、ステレオカメラから得られる SURF 特徴点²²⁾、特徴点の深度と LiDAR 点群を組み合わせる最適化問題の目的関数を定義して odometry を推定する。また、Zhang らは LOAM に基づいて、単眼カメラと LiDAR を組み合わせた、Visual-LiDAR Odometry and Mapping (VLOAM)¹³⁾ を提案している。この手法は、前段で特徴点から odometry 推定を行い、後段で、それを初期値として localization と mapping を行う。

近年では、疎な LiDAR 点群を用いた 3 次元 SLAM の需要が高まっている。しかし、疎な LiDAR 点群から高精度に深度を推定することは難しい。Odometry 推定の最適化を行ううえで、深度推定の誤差が大きい特徴点が存在すると、odometry の推定精度が低下する問題がある。本論文では、LiDAR とカメラを組み合わせる疎な LiDAR 点群でも高精度な odometry を推定する手法を提案する。

3. LiDAR Odometry and Mapping

本論文では LiDAR とカメラを組み合わせた odometry 推定手法を提案する。提案手法は、LOAM とカメラ画像から得られる特徴点を組み合わせる、より高精度な odometry 推定を実現する。本章では、まず LOAM による odometry 推定について紹介する。LOAM による odometry 推定は、時間的に連続する 2 つの LiDAR 点群 $\mathcal{P}_k, \mathcal{P}_{k-1}$ から行う。推定は、(1) LiDAR 点群の分類、(2) LiDAR 点の対応付け、(3) 最適化による odometry の算出の 3 つの処理で構成される。このうち (2) と (3) は ICP 法と同様に目的関数が最小値に収束するまで交互に反復する。

最初に表 1 に本論文で使用する記号の一覧を示す。本論文では、時刻 k の LiDAR 点群を \mathcal{P}_k 、画像を \mathcal{I}_k と表す。LiDAR 点群 \mathcal{P}_k の i 番目の LiDAR 点を $\mathbf{p}_{k,i}$ と表す。また、カメラの光学中心を原点とし、Z 軸をカメラの光軸方向に一致させた座標系をカメラ座標系、画像平面上の 2 次元座標系を画像座標系と呼ぶ。カメラと LiDAR は事前に較正しておき、相互に座標変換を行うのに必要なパラメータは既知とする。なお、これら座標系の基本的な変換方法については付録にまとめる。

3.1 LiDAR 点群の分類

LOAM はまず、LiDAR 点群をコーナと面に分類する。本論文では、コーナに分類された LiDAR 点をエッジ上の点、面に分類された点をプレーン上の点と表記する。エッジ上の点とプレーン上の点の分類は式 (1) で定義される c から判定する。具体的には、 c がしきい値より大きい場合はエッジ上の点と判定

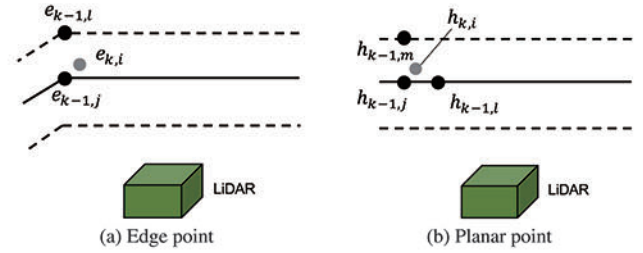


Fig. 3 Finding the correspondence of points

し、しきい値以下の場合はプレーン上の点と判定する。

$$c = \frac{1}{|S| \cdot \|\mathbf{p}_{k,i}\|} \left\| \sum_{j \in S, j \neq i} (\mathbf{p}_{k,i} - \mathbf{q}_{k,j}) \right\| \quad (1)$$

ここで $\mathbf{p}_{k,i}$ は LiDAR 点群 \mathcal{P}_k に含まれる i 番目の LiDAR 点を示し、 $\{\mathbf{q}_{k,j} | j \in S\}$ は $\mathbf{p}_{k,i}$ と同じ走査線から選択した、近傍の LiDAR 点の部分点群を示す。

3.2 LiDAR 点の対応付け

LiDAR 点の対応付けは、各点 $\mathbf{p}_{k,i}$ に対し、直前に観測された LiDAR 点群 \mathcal{P}_{k-1} から、kd-Tree を用いて探索して対応付けを行う。この際、エッジ上の点はエッジ上の点同士、プレーン上の点はプレーン上の点同士で対応付けを行う。

エッジ上の点 $\mathbf{e}_{k,i}$ は、エッジに分類された点の点群 \mathcal{E}_{k-1} から 2 つの点を探索し、2 点を通る線とエッジ上の点の距離を目的関数として定義する。1 点目はエッジ上の点の最近傍点 $\mathbf{e}_{k-1,j}$ 。2 点目は、1 点目 $\mathbf{e}_{k-1,j}$ の走査線に隣接する走査線から、エッジ上の点 $\mathbf{e}_{k,i}$ に最も近い点 $\mathbf{e}_{k-1,l}$ を探索して対応付けを行う (図 3 (a))。

プレーン上の点 $\mathbf{h}_{k,i}$ は、プレーンに分類された点の点群 \mathcal{H}_k から 3 つの点を探索し、3 点を通る平面とプレーン上の点の距離を目的関数として定義する。1 点目は最近傍点 $\mathbf{h}_{k-1,j}$ 、2 点目は $\mathbf{h}_{k-1,j}$ と同じ走査線上で $\mathbf{h}_{k-1,j}$ に隣接する点 $\mathbf{h}_{k-1,l}$ 、3 点目はプレーン上の点 $\mathbf{h}_{k-1,j}$ の走査線に隣接する走査線の最近傍点 $\mathbf{h}_{k-1,m}$ を探索して対応付けを行う (図 3 (b))。

3.3 最適化による Odometry の算出

対応付けされたエッジ上の点とプレーン上の点から目的関数を定義して、目的関数を最小にする odometry を Levenberg-Marquardt 法で推定する。

目的関数は、エッジ上の点とプレーン上の点と、それぞれ対応付けされた点の距離の総和として定義される。エッジ上の点の距離は、対応付けされた 2 点を通る直線と点の距離として式 (2) で定義する。

$$d_{\mathcal{E}} = \frac{|(\mathbf{e}_{k,i} - \tilde{\mathbf{e}}_{k-1,j}) \times (\mathbf{e}_{k,i} - \tilde{\mathbf{e}}_{k-1,l})|}{|\tilde{\mathbf{e}}_{k-1,j} - \tilde{\mathbf{e}}_{k-1,l}|} \quad (2)$$

ここで、 $\tilde{\mathbf{e}}_{k-1,i}$ は $\mathbf{e}_{k-1,i}$ を \mathbf{F}_k 中心の座標系に変換した点を表し、式 (3) で定義する。

$$\tilde{\mathbf{e}}_{k-1,i} = \mathbf{T}_k^{k-1} \mathbf{e}_{k-1,i} \quad (3)$$

プレーン上の点の距離は、対応付けされた 3 点を含む平面と点の距離として式 (4)、(5) で定義する。

$$\mathbf{n} = \left(\tilde{\mathbf{h}}_{k-1,j} - \tilde{\mathbf{h}}_{k-1,l} \right) \times \left(\tilde{\mathbf{h}}_{k-1,j} - \tilde{\mathbf{h}}_{k-1,m} \right) \quad (4)$$

$$d_{\mathcal{H}} = \frac{|(\mathbf{h}_{k,i} - \tilde{\mathbf{h}}_{k-1,j}) \cdot \mathbf{n}|}{|\mathbf{n}|} \quad (5)$$

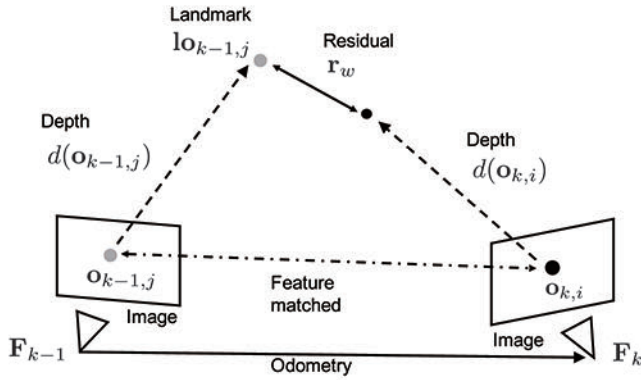


Fig. 4 Residual between landmark and image feature with high-confidence depth in the world coordinates

ここで、 $\tilde{\mathbf{h}}_{k-1,i}$ は、 $\tilde{\mathbf{e}}_{k-1,i}$ と同様に、 $\mathbf{h}_{k-1,i}$ を \mathcal{F}_k 中心の座標系に変換した点を表す。

4. 疎な LiDAR 点群と画像特徴点による Odometry 推定

前章で、LiDAR 点群から odometry を推定する代表的な手法である LOAM について説明した。本章では、提案手法である、LiDAR 点群と画像を組み合わせた odometry 推定手法について詳細に説明する。

Odometry 推定の最適化の目的関数は、式 (2) と式 (5) で定義した LiDAR 点群から得られるコストと、画像特徴点から得られるコストから式 (6) で定義する。Levenberg-Marquardt 法で、これを最小にする odometry を求める。

$$E = \sum_i d_\varepsilon + \sum_i d_H + \sum_i d_O \quad (6)$$

ここで d_O は、対応付けされた特徴点間の距離を示す。回転運動や奥行き方向の移動量を含む、6 自由度の odometry 推定を行うためには、 d_O を算出する時に特徴点の深度が必要になる。深度推定の誤差が大きい特徴点が存在すると、odometry の推定精度が低下する問題がある。提案手法は、特徴点の深度推定の信頼度に応じて、odometry 推定の最適化問題における目的関数を切り替える。対応付けされた特徴点間の距離 d_O は、深度推定の信頼度が高い特徴点の場合、深度を用いて 3 次元座標系に変換して算出する。一方、信頼度が低い特徴点の場合は画像座標系で算出する。

特徴点の深度と深度推定の信頼度は、疎な LiDAR 点群を入力として、ガウス過程回帰を用いて推定を行う。ガウス過程回帰は、深度の推定値が取りうる範囲をガウス分布として、その平均と分散を推定する。深度推定の信頼度は深度の推定値分布の分散の逆数で定義する。以降、4.1 節で特徴点間の距離 d_O の算出について詳細を述べて、4.2 節で深度の推定値が取りうる範囲をガウス分布として推定する手法を説明する。

4.1 対応付けされた画像特徴点間の距離

対応付けされた画像特徴点の距離は、特徴点 $\mathbf{o}_{k,i}$ とそれに対応したランドマーク $\mathbf{l}_{o_{k-1,j}}$ から定義する。特徴点のランドマーク $\mathbf{l}_{o_{k-1,j}}$ は、特徴点 $\mathbf{o}_{k-1,j}$ を深度 $d(\mathbf{o}_{k-1,j})$ でカメラ座標系に変換したもので式 (7) で定義される。

$$\mathbf{l}_{o_{k-1,j}} = \mathcal{T}^{-1}(\mathbf{o}_{k-1,j}, d(\mathbf{o}_{k-1,j})) \quad (7)$$

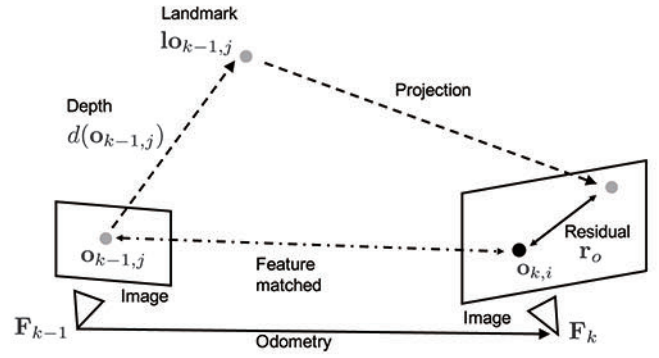


Fig. 5 Residual between landmark and image feature in the image coordinates

ここで、 $\mathcal{T}(\mathbf{p}_{k,i})$ はカメラ座標系の点 $\mathbf{p}_{k,i}$ を画像 \mathcal{I}_k の画像座標系へ変換する関数を表し、 $\mathcal{T}^{-1}(\mathbf{o}_{k,i})$ はその逆関数を表している。なお、この座標変換の詳細については付録で説明する。

特徴点 $\mathbf{o}_{k,i}$ とランドマーク $\mathbf{l}_{o_{k-1,j}}$ の対応付けは、直前に撮影された画像 \mathcal{I}_{k-1} から抽出された特徴点の集合 \mathcal{O}_{k-1} から局所特徴量による対応付けで探索する。特徴点の深度推定の信頼度が高い場合、特徴点 $\mathbf{o}_{k,i}$ のコストは、特徴点 $\mathbf{o}_{k,i}$ を深度 $d(\mathbf{o}_{k,i})$ でカメラを原点とした 3 次元のカメラ座標系に変換して、ランドマーク $\mathbf{l}_{o_{k-1,j}}$ との差 \mathbf{r}_w から算出する (図 4)。ここで、 \mathbf{r}_w を式 (8) で定義する。

$$\mathbf{r}_w = \mathbf{T}_{k-1}^k \cdot \mathbf{l}_{o_{k-1,j}} - \mathcal{T}^{-1}(\mathbf{o}_{k,i}, d(\mathbf{o}_{k,i})) \quad (8)$$

特徴点の深度推定の信頼度が低い場合、ランドマーク $\mathbf{l}_{o_{k-1,j}}$ を 2 次元の画像座標系に投影して、特徴点間の差 \mathbf{r}_o を算出する (図 5)。特徴点間の差 \mathbf{r}_o は、画像座標系の 2 次元ベクトルになり、式 (9) で定義する。

$$\mathbf{r}_o = \mathcal{T}(\mathbf{T}_{k-1}^k \cdot \mathbf{l}_{o_{k-1,j}}) - \mathbf{o}_{k,i} \quad (9)$$

特徴点の深度推定の信頼度は、深度の推定値が取りうる範囲を示す分散 $\sigma^2(\mathbf{o}_{k,i})$ の逆数で定義する。信頼度が大きい値 Th より大きい時は信頼度が高いと判定し、小さい時は信頼度が低いと判定する。最終的に、ランドマーク $\mathbf{l}_{o_{k-1,j}}$ と特徴点 $\mathbf{o}_{k,i}$ の距離を式 (10) で定義する。

$$d_O = \begin{cases} \mathbf{r}_w^T \mathbf{W}_w^{-1} \mathbf{r}_w & (1/\sigma^2(\mathbf{o}_{k,i}) < \text{Th}) \\ \mathbf{r}_o^T \mathbf{W}_o^{-1} \mathbf{r}_o & (\text{otherwise}) \end{cases} \quad (10)$$

ここで、 \mathbf{W}_w と \mathbf{W}_o は重み行列を示す。重み行列は、特徴点間距離の誤差を分散とした共分散行列とみなして値を設定する。このようにすることで特徴点間の距離 d_O はマハラノビス距離になる。3 次元のカメラ座標系中のメートル単位の距離と、2 次元の画像座標系中の画素単位の距離を直接加算することはできないため、各々についてマハラノビス距離に変換することによって加算できるようになる。

このように、特徴点の深度推定の信頼度が高い時だけ、深度を用いて 3 次元の座標系で d_O を算出することで、深度推定の誤差が小さい特徴点による 6 自由度の odometry 推定の精度向上と深度推定の誤差が大きい特徴点による odometry 推定の精度低下を防ぐことができる。

4.2 ガウス過程回帰による深度推定

ガウス過程回帰は、訓練データ集合が与えられると、新しい入力に対する予測分布をガウス分布として回帰することができ

る。これを深度推定に当てはめると、LiDAR 点群を入力とし、特徴点の深度の推定値が取りうる範囲をガウス分布として、その平均と分散を推定することができる。ガウス過程回帰のカーネル関数として式 (11) のガウスカーネルを用いる。ここで、 $\mathbf{u}, \mathbf{u}' \in \mathbb{R}^2$ は画像座標系の任意の点を表す。

$$K(\mathbf{u}, \mathbf{u}') = \exp\left(-\frac{\|\mathbf{u} - \mathbf{u}'\|^2}{2\sigma^2}\right) \quad (11)$$

画像座標系に投影した LiDAR 点群と特徴点をカーネル関数に入力して得られる出力を並べたベクトル \mathbf{K} と、LiDAR 点群から得られる深度を並べたベクトル \mathbf{t} を式 (12), (13) で表す。

$$\mathbf{K} = [K(\mathcal{T}(\mathbf{p}_{k,j}), \mathbf{o}_{k,i}), \dots, K(\mathcal{T}(\mathbf{p}_{k,l}), \mathbf{o}_{k,i})] \quad (12)$$

$$\mathbf{t} = [d(\mathcal{T}(\mathbf{p}_{k,j})), \dots, d(\mathcal{T}(\mathbf{p}_{k,l}))] \quad (13)$$

共分散行列 \mathbf{C} の要素は、 δ_{ij} を Kronecker のデルタとすると式 (14) で定義できる。

$$\mathbf{C}(\mathcal{T}(\mathbf{p}_{k,j}), \mathcal{T}(\mathbf{p}_{k,l})) = K(\mathcal{T}(\mathbf{p}_{k,j}), \mathcal{T}(\mathbf{p}_{k,l})) + \beta^{-1} \delta_{jl} \quad (14)$$

ここで、 β は深度 $d(\mathcal{T}(\mathbf{p}_{k,j}))$ の観測精度を表す超パラメータである。最終的に特徴点の深度を条件付き分布 $p(d(\mathbf{o}_{k,i})|\mathbf{t})$ として得ることができて、特徴点における深度が取りうる値の範囲は、次に示す平均 $d(\mathbf{o}_{k,i})$ と分散 $\sigma^2(\mathbf{o}_{k,i})$ をもつガウス分布として推定される²³⁾。

$$d(\mathbf{o}_{k,i}) = \mathbf{K}^t \mathbf{C}^{-1} \mathbf{t} \quad (15)$$

$$\sigma^2(\mathbf{o}_{k,i}) = K(\mathbf{o}_{k,i}, \mathbf{o}_{k,i}) + \beta^{-1} - \mathbf{K}^T \mathbf{C}^{-1} \mathbf{K} \quad (16)$$

ガウス過程回帰を実際に計算する際に、最も大きな計算量を要するのは、式 (15) と式 (16) で共分散行列 \mathbf{C} の逆行列を計算する部分である。これは行列の大きさを N とした場合、ガウス消去法 (掃き出し法) で逆行列を計算すると $O(N^3)$ の計算量となる。ここで、共分散行列 \mathbf{C} の大きさ N は LiDAR 点群の大きさである。本手法では、特徴点 $\mathbf{o}_{k,i}$ に対して、近傍の LiDAR 点の部分点群をガウス過程回帰の入力とする。部分点群の大きさは LiDAR 点群の大きさに比べて小さく、その結果、共分散行列 \mathbf{C} の大きさを小さくできて計算量を削減することができる。

図 6 にガウス過程回帰で深度の平均と分散を推定した結果を示す。上段はガウス過程回帰の入力となる LiDAR 点群の画像座標と深度を表している。中段と下段はガウス過程回帰で推定した、特徴点の深度の平均と分散を表している。LiDAR 点が近くにある特徴点の深度推定の分散は小さく、LiDAR 点が遠くない特徴点の深度推定の分散は大きいことがわかる。

5. 実験

提案手法の odometry 推定精度の評価するために実験を行った。本章では、その実験方法と評価結果を報告する。

評価用のデータセットとして、Velodyne とカメラを搭載した車両で様々な場所を走行して計測された、KITTI データセット²⁴⁾の中から、odometry の真値が付随している odometry データセットを用いた。odometry データセットは、主に住宅地や高速道路の走行シーンから構成されている。本論文では住宅地 (Residential) を走行したデータ (2011_10_03_drive_27) と高速道路 (Freeway) を走行したデータ (2011_10_03_drive_42) で評価を行った。また、odometry 評価では積算誤差よりも時刻毎の誤差を評価するため、それぞれの最初の 100 秒間を用いた。



(a) LiDAR points overlaid on an image (Color: distance [m])



(b) Depth at image feature points (Color: distance [m])



(c) Variance of depth at image feature points (Color: variance [m²])

Fig. 6 LiDAR points, depth and variance estimated from Gaussian process regression at image feature points

ここで、提案手法は Velodyne の 3 次元点群とグレースケールの前方画像データ cam0 (gray) を用いて odometry の推定を行う。

評価指標は、KITTI で定義された並進移動の誤差と回転角度の誤差^{24) 25)} にならない、それぞれを式 (17), (18) で定義する。

$$E_{\text{trans}} = \frac{1}{|\mathcal{F}|} \sum_{(m,n) \in \mathcal{F}} |\hat{\mathbf{t}}_m^n - \mathbf{t}_m^n| \quad (17)$$

$$E_{\text{rot}} = \frac{1}{|\mathcal{F}|} \sum_{(m,n) \in \mathcal{F}} |\angle \hat{\mathbf{R}}_m^n - \angle \mathbf{R}_m^n| \quad (18)$$

ここで、 \mathcal{F} は全フレーム、 $|\mathcal{F}|$ はカメラの総移動距離、 \mathbf{t}_m^n は odometry の並進移動量、 $\angle \mathbf{R}_m^n$ は odometry の回転角度を表す。また $\hat{\mathbf{t}}_m^n, \hat{\mathbf{R}}_m^n$ は、それぞれ odometry の並進移動量と回転角度の真値を表す。

提案手法の効果を確認するため、以下の手法で odometry 推定精度を比較した。

- GPD-VCSLO (提案) : ガウス過程回帰による深度推定と深度推定の信頼度に基づき目的関数の設定を切り替える、画像特徴点と LiDAR 点による odometry 推定手法
- CLD-VCSLO : LiDAR とカメラを組み合わせた推定手法。特徴点の最近傍の LiDAR 点から、その特徴点の深度を算出し、対応付けされた特徴点間の距離を式 (9) で算出する。これを Closest LiDAR point Depth and Visually Constrained Sparse LiDAR Odometry (CLD-VCSLO) と表記する。
- LOAM : LOAM¹⁾ の odometry 推定手法

提案手法は疎な LiDAR 点群でも高精度に odometry を推定することを目指している。KITTI データセットから疎な LiDAR 点群を模擬したデータを用いて各手法の odometry 推定手法を比較することで、提案手法の効果を確認する。KITTI データセットは、走査線数が 64 本の Velodyne 社製の HDL-64 で 3 次元点群を計測している。本来、走査線にはそれぞれに ID が

Table 2 Translation and rotation errors

Scene (Data)	LiDAR scan lines	Method	E_{trans} [%]	E_{rot} [deg/m]
Residential (2011_10.03_drive_27)	64	GPD-VCSLO (Proposed)	4.02	0.18
		CLD-VCSLO	4.62	0.19
		LOAM	9.10	0.34
	16	GPD-VCSLO (Proposed)	3.42	0.19
		CLD-VCSLO	3.68	0.19
	8	GPD-VCSLO (Proposed)	3.56	0.19
CLD-VCSLO		4.18	0.19	
Freeway (2011_10.03_drive_42)	64	GPD-VCSLO (Proposed)	2.83	0.08
		CLD-VCSLO	3.21	0.08
		LOAM	23.18	0.20
	16	GPD-VCSLO (Proposed)	5.09	0.10
		CLD-VCSLO	36.16	0.10
	8	GPD-VCSLO (Proposed)	41.46	0.10
CLD-VCSLO		41.94	0.10	

割り振られていて、各 LiDAR 点のデータに格納されている。しかし、KITTI データセットではこれが削除されているので、LiDAR 点の座標から、LiDAR 点のセンサ本体に対するピッチ角を算出して走査線 ID を推定する。これに基づいて走査線を間引き、走査線数が 16 本と 8 本の疎な LiDAR 点群を模擬して評価を行った。以降、“GPD-VCSLO 16” のように手法の後ろに走査線数を追記する。

また、提案手法で式 (10) の重み行列は実験的に式 (19), (20) のように設定した。

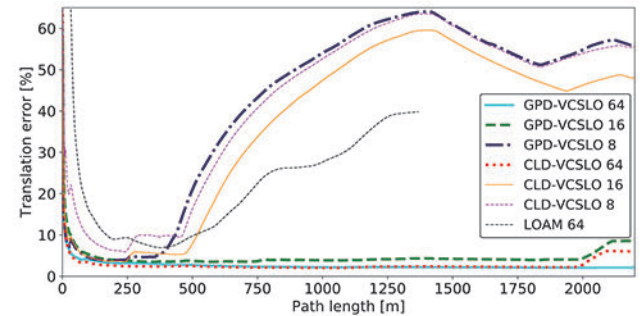
$$W_w = \begin{bmatrix} 2.5^2 & 0.0 & 0.0 \\ 0.0 & 2.5^2 & 0.0 \\ 0.0 & 0.0 & 2.5^2 \end{bmatrix} \quad (19)$$

$$W_o = \begin{bmatrix} 10.0^2 & 0.0 \\ 0.0 & 10.0^2 \end{bmatrix} \quad (20)$$

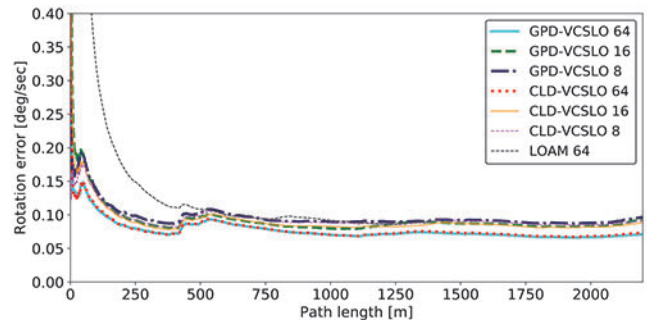
表 2 に評価結果を示す。太字の数字は最も精度が高かった手法の精度を表す。この結果、全てのデータで提案手法により odometry 推定精度が向上することを確認した。走査線数が 16 本と 8 本の疎な LiDAR 点群では、LOAM は初期位置から進むことができず、odometry の推定ができなかった。

住宅地 (Residual) のデータで評価した結果、odometry の回転角度の推定精度は、GPD-VCSLO 64 (提案) で最も高くなった。走査線数が疎になっても回転角度の推定精度に大きな低下は見られず、提案手法の効果を確認できた。一方で、odometry の並進移動量の推定精度は、GPD-VCSLO 16 (提案) で最も高くなった。これは、Velodyne の走査形状が原因で誤対応付けされた LiDAR 点か、画像特徴点に比べて目的関数に多く入力されたためと考えられる。走査線数が 8 本の odometry 推定精度は、提案手法と比較手法のいずれの手法でも、16 本の場合よりも推定精度が低下した。これは、LiDAR 点群の数が減ることで、特徴点の深度推定の誤差が大きくなったためと考えられる。提案手法では深度推定の信頼度を用いて目的関数を切り替えているため、比較手法よりも odometry 推定の精度低下を防ぐことができた。

次に高速道路 (Freeway) のデータで評価した結果では、odometry の並進移動量の推定精度は、GPD-VCSLO 64 (提案) で最も高くなり、odometry の回転角度の推定精度は、GPD-VCSLO 64 (提案) と CLD-VCSLO 64 が最も高くなった。疎な LiDAR を模擬した評価では、GPD-VCSLO 16 (提案) で、最も高い odometry 推定精度が得られた。CLD-VCSLO 16, CLD-VCSLO 8 と GPD-VCSLO 8 は並進移動量の推定精度が大きく低下した。図 7 に、高速道路のデータで評価した結果を示す。横軸に



(a) Translation error



(b) Rotation error

Fig. 7 Translation and Rotation errors in the Freeway data evaluation

走行距離、縦軸に式 (17) と式 (18) で定義した評価指標を描画した。これを見ると、並進移動量の推定精度は、走行距離 250 m から大きく低下していることがわかった。一方、GPD-VCSLO 16 は、他の手法に比べて、並進移動量の推定精度の大きな低下は見られなかった。推定精度が大きく低下している理由については次章で詳細に考察する。

以上の結果、画像特徴点と LiDAR 点を合わせて odometry を推定する手法は、odometry の推定精度向上に有効で、特徴点の深度の推定精度に応じて目的関数を切り替えることが、odometry の推定精度向上に有効であることが示された。

6. 考 察

5 章における評価の結果、走査線数が 16 本と 8 本の疎な LiDAR 点群では、高速道路 (Freeway) のデータの評価結果は住宅地 (Residential) のデータに比べて並進移動量の誤差が大きかったことがわかった。高速道路のデータの中で誤差が大きかったシーンを図 8 に示す。高速道路を走行していて、かつ先行車が走行しているシーンは、先行車から得られる対応点が移動量推定の外れ値になるため、特に難しいシーンであると考



Fig. 8 Example of a challenging situation

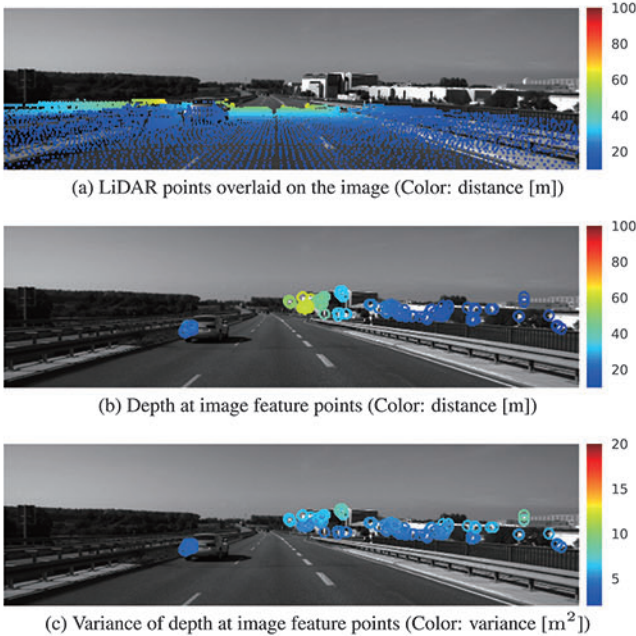


Fig. 9 LiDAR points, depth and variance estimated from Gaussian process regression at image feature points in the challenging situation

えられる。このシーンにおいて、ガウス過程回帰による特徴点の深度と分散の推定結果を図9に示す。図9(a)から、道路より外側の構造物はLiDARで検知可能な距離より遠方であるため、LiDARの反射点を得ることができないことがわかる。路面や周囲のガードレールからLiDARの反射点を多く得ることができるが、路面は平面のため、また、ガードレールは周期的構造をもつため、2フレーム間でLiDAR点を正しく対応付けることは難しい。図9(b)からは、ガードレールや路面の特徴点是对応付けを得ることができていないことがわかる。画像特徴点はガードレール等の周期的構造から得られる特徴点同士を正しく対応付けることは難しい。道路外の建物から得られる特徴点是对応付けができていないが、LiDAR点群が近くないため、深度推定の誤差が大きい。従来手法ではこれが原因でodometryの推定精度が低下していると考えられる。提案手法では深度推定の信頼度が低い点を除去しているため、従来手法に比べて精度が向上したと考えられる。

一方、走査線数が8本の疎なLiDAR点群では、提案手法も同じシーンでodometryの推定精度が大きく低下した。遠方の建物から得られる特徴点は、深度推定の信頼度が低く提案手法では除去される。一方、先行車から得られる特徴点の周囲にはLiDAR点が多く存在して、深度推定の信頼度は高くなる。提案手法は、このシーンで、先行車の特徴点は深度を用いて3次元座標系で特徴点間の距離を算出する。一方、遠方の建物は画

像座標系で特徴点間の距離を算出する。その結果、奥行き方向の移動量を推定するのに重要な、深度推定の信頼度が高い特徴点は先行車から得られた特徴点だけになるため、odometryの推定精度が低下したと考えられる。これを防ぐためには、先行車の特徴点を除去する必要がある。移動物の除去については、オプティカルフローを利用する手法²⁶⁾や、セマンティックセグメンテーションを利用する手法^{27) 28)}が提案されており、odometry推定の精度向上の効果が確認されている。

高速道路のシーンでは、先行車以外の特徴点が少ないことも原因と考えられる。これを解決するには、路面やガードレールから特徴点を抽出して正しく対応付けすることが必要がある。自動車が直進する2フレームの画像間では、路面上の局所領域が正方形から台形に変換される。これは射影変換になるためORB特徴量²⁹⁾が不変にならない。このため、路面の特徴点是对応付けが難しい。路面上の特徴点の対応付けを増やすためには、射影変換を考慮した特徴点の抽出と対応付け³⁰⁾が有効と考える。また、ガードレールは、周期的構造により正しく対応付けすることが難しい。ガードレールの特徴点を、正しく対応付けするためには、対応付けの候補となる特徴点を事前に絞り込むことが有効と考えられる。提案手法は、ガウス過程回帰を用いて特徴点の深度と合わせて、深度の推定値が取りうる範囲を分散として推定することができる。この分散を用いて対応付け候補の特徴点を事前に絞り込むことで、周期的構造をもつ構造物であっても、精度が高い対応付けを実現できると考えられる。

更には、特徴点を用いずに画素値を直接用いる手法^{3) 18) 31)}を組み合わせると、より精度が向上することが期待される。

7. 結 言

本論文では、疎なLiDAR点群と画像特徴点を組み合わせて、高精度なodometry推定手法を提案した。提案手法は、特徴点の深度推定の信頼度に応じて、2フレーム間の画像で対応付けされた特徴点間の距離の算出方法を切り替える点に特徴がある。また、特徴点の深度と推定の信頼度について、疎なLiDAR点群を入力として、ガウス過程回帰を用いて推定する手法を提案した。

提案手法の有効性を確認するため、KITTIのodometryデータセットでodometry推定精度の評価を行った。走査線数が16本と8本のLiDAR点群を模擬したデータで評価を実施したところ、提案手法は疎なLiDAR点群でも高精度なodometryを推定できることを示した。

評価の結果、高速道路で先行車に追従するシーンで、odometry推定の精度が低下する課題があることがわかった。推定精度が低下する原因を考察し、さらなる精度向上を目指すためには、先行車の特徴点を除去する必要があることがわかった。また、路面やガードレールから抽出された特徴点を正しく対応付けることも効果があると考えられる。正しい対応付けを増やすためには、射影変換を考慮した特徴点の対応付け^{30) 32)}や、odometryと深度推定の結果から対応付け候補となる特徴点を絞りこむ手法が有効と考えられる。

一方、局所特徴量を用いずに、画像の画素値を直接活用する手法も提案されており^{3) 18) 31)}、これらの手法と組み合わせることで、odometry推定の精度向上が期待される。

SLAMはodometry推定の後に、localizationとmappingを行い、地図生成と自己位置推定を同時に行う技術である。Local-

ization は odometry を初期値として最適化問題を解くことで推定が行われる。本論文で提案した手法は、高精度の odometry 推定が可能で、localization の精度向上に貢献する。また、提案した最適化手法は localization の推定に応用することが可能で、localization と mapping の精度向上にも有効であると考えられる。今後は localization と mapping の研究と合わせて SLAM 技術の研究を進めていく。

謝 辞

本研究の一部は、科学研究費補助金 JP17H00745 による。

参 考 文 献

- 1) J. Zhang and S. Singh: LOAM: LiDAR Odometry and Mapping in Real-time, Proc. Robotics: Science and Systems Conference, X (2014).
- 2) 竹内栄二郎, 坪内孝司: 移動ロボットによる環境地図生成のための 3 次元化 NDT を用いた高速 3 次元スキャンマッチング, 第 11 回ロボティクスシンポジウム講演予稿集 (2006) 252.
- 3) J. Engel and D. Cremers: LSD-SLAM: Large-scale Direct Monocular SLAM, Proc. 13th European Conf. on Computer Vision, 2 (2014) 834.
- 4) R. Mur-Artal, J.M.M. Montiel and J.D. Tardós: ORB-SLAM: A Versatile and Accurate Monocular SLAM System, IEEE Trans. Robot., 31, 5 (2015) 1147.
- 5) T. Whelan, H. Johannsson, M. Kaess, J. Leonard and J. McDonald: Robust Real-time Visual Odometry for Dense RGB-D Mapping, Proc. 2013 IEEE Int. Conf. on Robotics and Automation (2013) 5724.
- 6) Y. Shin and A. Kim: Sparse Depth Enhanced Direct Thermal-Infrared SLAM Beyond the Visible Spectrum, IEEE Robot. Autom. Lett., 4, 3 (2019) 2918.
- 7) P. Biber and W. Strasser: The Normal Distributions Transform: A New Approach to LASER Scan Matching, Proc. 2003 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 3 (2003) 2743
- 8) H.P. Moravec: Sensor Fusion in Certainty Grids for Mobile Robots, AI Mag., 9, 2 (1988) 61.
- 9) S. Thrun, W. Burgard and D. Fox: A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots, Mach. Learn., 31, 1-3 (1998) 29.
- 10) S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani and H. Durrant-Whyte: Simultaneous Localization and Mapping with Sparse Extended Information Filters, Int. J. Robot. Res., 23, 7-8 (2004) 693.
- 11) G. Grisetti, C. Stachniss and W. Burgard: Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling, Proc. 2005 IEEE Int. Conf. on Robotics and Automation (2005) 2432.
- 12) W. Zhen, Y. Hu, J. Liu and S. Scherer: A Joint Optimization Approach of LiDAR-Camera Fusion for Accurate Dense 3-D Reconstructions, IEEE Robot. Autom. Lett., 4, 4 (2019) 3585.
- 13) J. Zhang and S. Singh: Visual-LiDAR Odometry and Mapping: Low-drift, Robust, and Fast, Proc. 2015 IEEE Int. Conf. on Robotics and Automation (2015) 2174.
- 14) M. Velas, M. Spänzel and A. Herout: Collar Line Segments for Fast Odometry Estimation from Velodyne Point Clouds, Proc. 2016 IEEE Int. Conf. on Robotics and Automation (2016) 4486.
- 15) T. Shan and B. Englot: LeGO-LOAM: Lightweight and Ground-optimized LiDAR Odometry and Mapping on Variable Terrain, Proc. 2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2018) 4758.
- 16) P.J. Besl and N.D. McKay: A Method for Registration of 3D Shapes, IEEE Trans. Pattern Anal. Mach. Intell., 14, 2 (1992) 239.
- 17) Velodyne, <https://velodynelidar.com/>
- 18) J. Engel, V. Koltun and D. Cremers: Direct Sparse Odometry, IEEE Trans. on Pattern Anal. Mach. Intell., 40, 3 (2018) 611.
- 19) H. Strasdat, J. Montiel and A. Davison: Scale Drift-aware Large Scale Monocular SLAM, Proc. Robotics: Science and Systems, IV (2010).
- 20) R. Wang, M. Schwörer and D. Cremers: Stereo DSO: Large-scale Direct Sparse Visual Odometry with Stereo Cameras, Proc. 18th IEEE Int. Conf. on Computer Vision (2017) 3923.
- 21) J. Graeter, A. Wilczynski and M. Lauer: LIMO: LiDAR-Monocular Visual Odometry, Proc. 2018 IEEE/RSJ Int. Conference on Intelligent Robots and Systems (2018) 7872.
- 22) H. Bay, T. Tuytelaars and L. Van Gool: SURF: Speeded Up Robust Features, Proc. 9th European Conf. on Computer Vision, 1 (2006) 404.
- 23) C.M. ビンヨップ: パターン認識と機械学習 (2012).
- 24) A. Geiger, P. Lenz and R. Urtasun: Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite, Proc. 2012 IEEE Conf. on Computer Vision and Pattern Recognition (2012) 3354.
- 25) R. Kuemmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stach-

- niss and A. Kleiner: On Measuring the Accuracy of SLAM Algorithms, Auton. Robots, 27, 4 (2009) 387.
- 26) J. Cheng, Y. Sun and M. Q.-H. Meng: Improving Monocular Visual SLAM in Dynamic Environments: An Optical-flow-based Approach, Adv. Robot., 33, 12 (2019) 576.
- 27) C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei and Q. Fei: DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments, Proc. 2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2018) 1168.
- 28) X. Chen, A. Milioto, E. Palazzolo, P. Figue, J. Behley and C. Stachniss: SuMa++: Efficient LiDAR-based Semantic SLAM, Proc. 2019 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (2019) 4530.
- 29) E. Rublee, V. Rabaud, K. Konolige and G. Bradski: ORB: An Efficient Alternative to SIFT or SURF, Proc. 12th IEEE Int. Conf. on Computer Vision (2011) 2564.
- 30) G.-R. Cai, P.-M. Jodoin, S.-Z. Li, Y.-D. Wu, S.-Z. Su and Z.-K. Huang: Perspective-SIFT: An Efficient Tool for Low-altitude Remote Sensing Image Registration, Signal Process., 93, 11 (2013) 3088.
- 31) Y. Shin, Y.S. Park and A. Kim: Direct Visual SLAM Using Sparse Depth for Camera-LiDAR System, Proc. 2018 IEEE Int. Conf. on Robotics and Automation (2018) 5144.
- 32) J.-M. Morel and G. Yu: ASIFT: A New Framework for Fully Affine Invariant Image Comparison, SIAM J. Imaging Sci., 2, 2 (2009) 438.

付 録

本論文で使用する基本的な座標変換について式を定義する。

カメラ座標系の座標変換

地図の 1 点を原点とした 3 次元の座標系を地図座標系と呼び、カメラの光学中心を原点とし、Z 軸をカメラの光軸方向に一致させた座標系をカメラ座標系と呼ぶ。地図座標系において、カメラポーズ F_m は地図の原点からの移動量 T_m として定義できる。移動量 T_m は回転 R_m と並進 t_m で構成され、例えば、カメラポーズ F_m を原点としたカメラ座標系で得られる点 $p_{m,i}$ を地図座標系に変換する式は次のように定義できる。

$$T_m p_{m,i} = R_m p_{m,i} + t_m \quad (21)$$

また、カメラポーズ F_m とカメラポーズ F_n の相対移動量は式 (22) で定義される。

$$T_m^n = T_n^{-1} T_m \quad (22)$$

これを用いて、カメラポーズ F_m の座標系で得られる点 $p_{m,i}$ をカメラポーズ F_n の座標系に変換するには、 $T_m^n p_{m,i}$ とすればよい。

カメラ座標系から画像座標系への投影変換

画像中心を原点とした 2 次元の座標系を画像座標系と呼ぶ。ここでは、カメラポーズ F_m のカメラ座標系の点 $p_{m,i}$ を画像 I_m 座標系へ変換する $T(p_{m,i})$ を定義する。カメラの内部パラメータ

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (23)$$

を用いると、 $p_{m,i} = (X_C, Y_C, Z_C)$ を画像座標系に変換した点は式 (24) のように定義される。

$$T(p_{m,i}) = \begin{pmatrix} f_x \frac{X_C}{Z_C} + c_x \\ f_y \frac{Y_C}{Z_C} + c_y \end{pmatrix} \quad (24)$$

逆に、画像座標系 (u, v) からカメラ座標系への変換は、深度 $d(u, v)$ を用いて $T^{-1}((u, v), d(u, v))$ と表し、これは式 (25) のように定義される。

$$T^{-1}((u, v), d(u, v)) = \begin{pmatrix} d(u, v) \left(\frac{u}{f_x} - c_x \right) \\ d(u, v) \left(\frac{v}{f_y} - c_y \right) \\ d(u, v) \end{pmatrix} \quad (25)$$