

環境地図のスケールに合わせた Visual Odometry

山口 真 弘[†] 森 尚 平^{††} 斎 藤 英 雄[†](正会員)
 谷内田 尚 司^{†††} 柴 田 剛 志^{†††}

[†]慶應義塾大学, ^{††}グラーツ工科大学, ^{†††}日本電気株式会社

Visual Odometry Adjusted to the Scale of a Global Map

Masahiro YAMAGUCHI[†], Shohei MORI^{††}, Hideo SAITO[†](*Member*),
 Shoji YACHIDA^{†††}, Takashi SHIBATA^{†††}

[†] Keio University, ^{††} Graz University of Technology, ^{†††} NEC Corporation

あらまし 本手法では, Structure from Motion (SfM) を用いて事前に作成した環境地図に合わせた Visual Odometry (VO: 画像による自己位置推定手法) によるカメラ軌跡の推定手法とその利用方法を提案する. 提案手法では, VO の一つとして Direct Sparse Odometry (DSO) を取り上げ, この DSO の出力を用いて環境地図上でのカメラ位置姿勢の再計算を行う. その際, これらの関係づけられたカメラ位置姿勢を用いてポーズグラフを作成し最適化することで, DSO で推定される全フレームの環境地図上への位置合わせを実現する. そのため, 本手法は単眼のカラーカメラのみで利用でき, 学習データも不要であるという利点を有する. 提案手法により, 事前観測して作成した環境地図と現在ユーザが観測する光景との差から変更のあった箇所を物体単位で認識するシステムを実装し, 実験室環境でリアルタイム動作が可能であることを確認した. さらに, 既存のデータセットを用いた精度評価の実験を行った結果, 提案手法では DSO のみを用いた場合に比べ位置姿勢推定の精度が向上した.

キーワード: カメラ位置姿勢推定, ビジュアルオドメトリ, 3次元再構築, グラフ最適化, 個別物体領域分割

<Summary> We propose a method for estimating camera trajectories optimally aligned with prepared environment maps and its application. We implement Direct Sparse Odometry (DSO) as real-time Visual Odometry (VO) and Structure from Motion (SfM) to build environment maps. In this process, a pose graph is created and optimized using related camera poses to calculate the alignment of all frames from DSO towards the environment map with the scales matched. Therefore, the proposed method runs solely with a monocular color camera and does not require any training datasets. We demonstrate that the proposed method is able to be used as an object-based recognition system based on the difference between a map as a prior information and the current scene observations by the user. We tested this real-time application in a laboratory environment. In addition, we conducted an experiment to evaluate the accuracy of the proposed method using an existing dataset. The results showed that the proposed method improved the accuracy of pose estimation compared to the case using only DSO.

Keywords: camera pose estimation, visual odometry, structure from motion, graph optimization, instance segmentation

1. はじめに

画像のみを用いてカメラ位置姿勢と環境地図を実時間で逐次推定する Visual Odometry (VO: 画像による自己位置推定手法) は自動運転や拡張現実感 (AR) の基幹技術として考えられている¹⁾. 奥行推定のできない通常のカメラによる VO では, カメラ位置姿勢と環境地図間の相対的スケールしか推

定できないため, VO により得られる環境地図のスケールは, VO 初期化の際のカメラの動かし方やその方法に依存して実行ごとに異なり, 実世界の絶対スケールを得ることはできない. このため, 事前に作成した高精度な環境地図や地図上の位置に紐づけられた情報を, VO 実行時に得られる環境地図と正しいスケールで対応付けられないという問題があった. 慣性計測装置 (IMU) や地球上での絶対位置が得られる

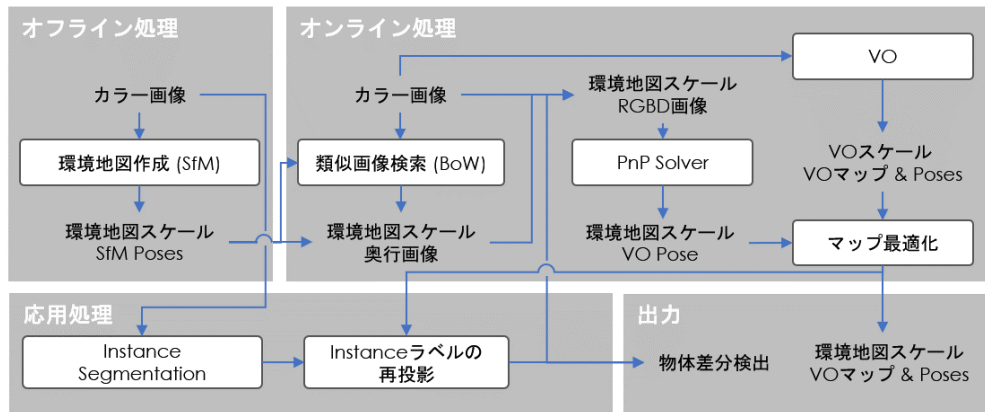


図1 システム概要
Fig. 1 System overview

Global Positioning System (GPS)などを組み合わせたセンサフュージョンは、環境地図上で実世界のスケールに基づいてカメラの位置姿勢を推定する手法である^{2)~4)}。センサを増やす方法に対して、単眼のカラーカメラの入力画像からConvolutional Neural Network (CNN)を用いて距離画像を推定する手法^{5),6)}は、学習方法によって実世界のスケールで距離画像を推定することができる。しかし、教師付き学習に基づく仕組み上、大量の学習データが必要であり、テスト環境が学習データと類似していなければならないなどの制約がある。また従来より、事前構築された環境地図上でカメラ位置姿勢を推定する手法が多数提案されている^{7),8)}。これらの手法は、フレームごとにデータベースとマッチングを取ってカメラ位置姿勢を求めるのみであるため、連続したフレームの幾何学的関連性を考慮したものにはなっていない。

本論文では、事前準備した環境地図上で、単眼のカラーカメラから得られた画像シーケンスによって全てのフレームでカメラの位置姿勢推定を行うことを目的としている。そのため、単眼カラーカメラ入力画像列にVO手法の一つであるDirect Sparse Odometry (DSO)⁹⁾を適用して、得られるスケール未知の環境地図と事前に作成した環境地図間で奥行値の比較により、DSOによりスケール未知で推定されるカメラ位置姿勢を事前作成の環境地図のスケールに合致させて推定する手法を提案する。さらに、環境地図上でDSOの位置姿勢を取得することによる3次元構造を用いた物体差分検出の手法を提案する。

これによって以下のような利点が挙げられる。まず、(1) VOを用いることで環境地図情報が一時的に得られない場合でもカメラの位置情報を得ることができる上、(2) 環境地図を参照できる場合はそこから得られる高精度な奥行情報を用いてドリフト誤差を解消できる。よって、(3) 提案手法を用いてDSOによって求めた位置姿勢を環境地図上で補正することで、既存のDSOのみを用いた場合に比べて精度良くすべてのカメラ位置姿勢を環境地図上で推定することができる。また、(4) 環境地図上でDSOのカメラ位置姿勢を推定することに

よって環境地図上に紐づけをしておいた情報へのアクセスが可能になる。その利用例として個別物体領域分割 (Instance Segmentation) を用いて、環境地図が持つ物体識別ラベルとDSOの入力画像から得られた物体識別ラベルの比較による物体の差分検出を実現できる。

2. 関連手法

2.1 VOとSLAM

VO^{9),10)}はカメラ画像を用いてカメラの自己位置推定をするための枠組みである。入力画像シーケンスから画像間のマッチングと3次元復元を同時に行い、カメラの自己位置とそれまでの軌跡を推定する。VOは入力画像から直近の数フレームを用いてカメラの位置姿勢と環境の3次元構造を最適化する仕組みになっているので、ドリフト誤差が生じるとその誤差を解消する方法はなく、常に誤差が蓄積していく構造となっている。それに対してSimultaneous Localization and Mapping (SLAM)^{2),3),11)}ではこのようなドリフト誤差を軽減するための仕組みとして局所最適化やバンドル調整、Loop Closureといった機能が盛り込まれている¹⁾。

こういった最適化の仕組みにかかわらず、VOやSLAMでは連続した画像シーケンスの中で直前に推定した3次元構造に対する相対的なカメラ位置姿勢を推定することしかできない。よって、得られる環境地図やカメラ位置姿勢は実世界のスケールに基づいておらず、使用者のカメラの動かし方や被写体に応じて異なるスケールで出力される。そのため、実世界のスケールで環境地図に固定されている3次元コンテンツを、VOやSLAMから得られた位置姿勢によりAR表示するなどの用途には利用できなかった。

Visual-Inertial Odometry (VIO)はこのような問題を解決する一つの方法とであり、IMUによって求められたスケールによって、カメラの自己位置を実世界のスケールに合わせることができる。VIOによって事前取得した環境地図に現在の軌跡を登録していく手法は近年提案されている^{12),13)}。またGPSを用いてSLAMと実世界のスケールとの対応関係を

得る方法も提案されている¹⁴⁾。

このような複数センサを組み合わせた手法に対して、提案手法は、センサを組み合わせることなく、単眼のカラーカメラのみを用いて事前取得した環境地図に VO 手法の一つである DSO から得られた軌跡を同一座標系、同一スケールで登録することが可能である。これによって事前に取得した環境地図に付随している物体識別ラベルを参照して、DSO によって推定された 3 次元構造から物体の差分検出が可能になる。

2.2 事前準備した環境地図上でカメラ位置推定

事前に取得した環境地図に対応したカメラ位置姿勢を求める方法として、環境地図内のランドマークと入力画像の特徴点を紐づけることでカメラの自己位置姿勢を推定する手法がある^{7),15),16)}。これらの手法では環境地図のランドマークが画像内に写っていることが前提となっているため、登録したランドマークが写っていない画像に対して位置姿勢を推定することはできない。

また、PTAMM は複数回の試行で得られた複数のローカルマップを統合する手法として提案された¹⁷⁾。しかしこの手法は机の周りのような小規模なシーンを想定して作られている。また、これらの VO では、環境地図を作成した手法と同一の手法を用いて、自己位置姿勢推定を行いながら、再度トラッキングをするのが一般的である。そのため、事前取得した環境地図を構成する特徴量と VO のトラッキングに用いられる特徴量が同一であることが前提となっている。また VO に用いた手法と同様の手法を用いて環境地図を作成する際、環境地図と VO の両方で、同じ箇所ではトラッキングに失敗する恐れがある。

従来では、事前準備した環境地図上でカメラ位置姿勢を求める際は、環境地図と照合されたフレームのみでの位置姿勢を求める手法が着目されてきた⁷⁾。それに対して提案手法では、ランドマークが観測できないフレームでも、VO によってマッチングが得られた位置からの相対的な位置姿勢を取得することができる。これによって、すべての入力画像シーケンスでカメラの自己位置姿勢を推定することができる。また最先端の VO を用いることで、机周りのような小規模なシーンにとどまらず KITTI データセット¹⁸⁾のような屋外の大規模なシーンでも利用が可能である。さらに事前準備する環境地図と VO で異なる 3 次元点の作成方法を利用するため、VO でトラッキングに失敗するような箇所を環境地図で補間することができる。さらに、スケールを合わせて推定を行うことで、実行ごとの利用を目的とした VO では実現できなかった、環境地図による 3 次元構造と VO によるカメラ位置姿勢を利用した、物体の差分検出を可能にする。

3. 提案手法

本手法は、事前に取得した画像シーケンスから Structure from Motion (SfM) を用いて全体最適化を行った 3 次元環境

地図を作成するオフライン処理と、入力画像に対して、VO の一つである DSO を用いて環境地図上でカメラの自己位置推定を行うオンライン処理から成る。オンライン処理を行う際には、環境地図のスケールから DSO のスケールを補正し、座標系の統合を逐次行うことで DSO で得られた情報を環境地図上に反映する。さらに オンライン処理において、環境地図と現在取得した画像との個別物体領域分割の結果を比較することで、新たに環境地図に加えられたものと、環境地図から取り除かれたものを 3 次元構造を用いて差分検出することが可能になる。

3.1 システム概要

図 1 に提案手法の概要を示す。単眼カラーカメラ入力画像列からカメラの位置姿勢をオンラインで推定する手法として VO と vSLAM が広く用いられている。本手法では環境地図を基にオンラインで推定中の軌跡の補正を行うため、逐次入力される画像列のみでバンドル調整や Loop Closure などの大域最適化を行う vSLAM のような枠組みを用いる必要がない。よって、VO をオンラインでの軌跡推定に用いる。また、本手法では VO を単体で動かすよりも高精度にカメラ位置姿勢を推定することを目的とするため、環境地図の作成には VO よりも精度よくカメラ位置姿勢を推定できる方法を用いる。そのため同じくカラーカメラ画像群を入力とする SfM を環境地図の作成に利用している。SfM はすべてのフレームを使って全体最適化しているため、VO や SLAM と比較して高精度にカメラの自己位置推定と 3 次元形状の復元を行うことができる。そこで、オフラインでは SfM を用いて事前に環境地図 G を作成する。 G にはスケール s^{SfM} で作成した距離画像に加えて、カメラ位置姿勢、Bags of Binary Words (BoW)¹⁹⁾ のデータベース、個別物体領域分割の結果が含まれている。

オンラインでは入力画像に対して DSO を用いてカメラの位置姿勢推定を行う。その際の DSO のスケールは s^{VO} で表される。本手法では事前に作成した環境地図上に DSO の軌跡を紐づける。つまり、環境地図に対して DSO の入力画像の自己位置姿勢推定をする必要がある。関連した技術に、vSLAM に用いられている Loop Closure があり、これは連続していない時刻に観測された画像を BoW による類似画像検索を使って位置合わせする手法として広く用いられている^{2),3)}。本手法の目的である、連続していない時刻に観測された画像を既存の地図に合わせて自己位置姿勢推定することは、Loop Closure の目的と合致しているため、本手法では Loop Closure を参考にして、BoW を用いて自己位置姿勢推定を行う。その結果をもとに画像間の特徴点マッチングと Perspective-n-Point(PnP) 問題を解くことで環境地図上でカメラ位置姿勢を推定している。その際、DSO の距離画像と環境地図の 3 次元構造を比較することによって s^{VO} と s^{SfM} の比率を計算し、また類似フレームが見つからなかつ

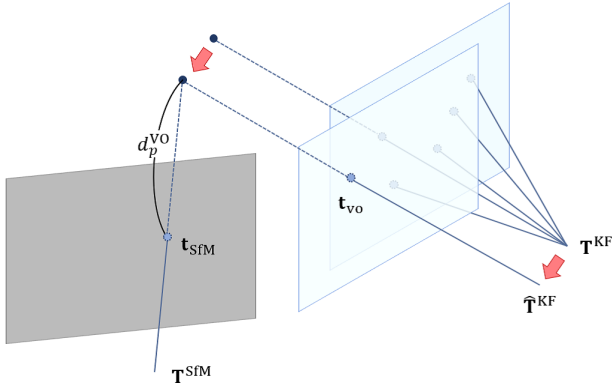


図2 VO座標系から環境地図座標系への座標系変換による奥行値の取得

Fig. 2 Obtaining depth values through coordinate system transformations from VO coordinate system to SfM coordinate system

たフレームに対しても、グラフ最適化を行うことによって、DSOによって計算されたカメラ位置姿勢を環境地図上で再計算し登録を行う。

手法の応用としてオンラインの入力画像に対して個別物体領域分割を行い、その結果をあらかじめ取得して環境地図が保持している個別物体領域分割の結果と比較することによって環境地図に対する物体識別ラベルの差分検出を行う。ここでは環境地図作成時と比較して新たに追加されたものと、取り除かれたものに関して検出を行う。

3.2 SfMを用いた環境地図の作成

オフラインでの処理として、事前取得しておいた M 枚の画像から SfM を用いて環境地図 G を作成する。ここで SfM を用いて作成された環境地図はカラー画像群 I^{SfM} 、距離画像群 D^{SfM} 、その時のカメラ位置姿勢 T^{SfM} によって構成されている。今後、 i 番目 ($< M$) におけるカラー画像、距離画像、カメラ位置姿勢はそれぞれ $I_i^{\text{SfM}} \in I^{\text{SfM}}$ 、 $D_i^{\text{SfM}} \in D^{\text{SfM}}$ 、 $T_i^{\text{SfM}} \in T^{\text{SfM}}$ と定義する。それに加えて DSO の入力に用いられるフレームに対する類似フレームを I^{SfM} のなかから検索して得るために、ORB 特徴量²⁰⁾ $F_i^{\text{SfM}} \in F^{\text{SfM}}$ による BoW データベースの作成を行う。

また環境地図と DSO への入力との差分検出を行うために I^{SfM} に対して個別物体領域分割を行い、各フレームの物体識別ラベルを得る。SfM によって推定された 3 次元点群をそれぞれ I^{SfM} に再投影して各 3 次元点を持つ物体識別ラベルの集計を行う。ここで一つの点が複数のラベルを持っている際には、最も票数の多いラベルをその点のラベルとする。また各点に対して、同一ラベルをしきい値以上の保有している点のみ有効だと判定し、 L^{SfM} として環境地図に保存する。本手法ではこのしきい値を 3 フレーム以上とした。よって本手法では事前準備した環境地図を $G \in \{I^{\text{SfM}}, D^{\text{SfM}}, T^{\text{SfM}}, F^{\text{SfM}}, L^{\text{SfM}}\}$ と定義する。

3.3 環境地図上での VO の位置姿勢推定

オンラインでは、事前取得した環境地図上でのカメラの位置姿勢推定と 3 次元構造の推定を VO を用いて行う。単眼のカラーカメラを用いた VO では、作成される 3 次元構造のスケール s^{VO} が初期化の際のステップ幅などに依存してランダムに定まる⁹⁾。そのため事前に作成した環境地図上で VO の位置姿勢を推定するためには、VO のスケールを環境地図のスケールに合わせたうえで、カメラ位置姿勢や 3 次元構造を補正する必要がある。そこで本手法では DSO によって得られた距離画像 D^{VO} と環境地図を構成している距離画像 D^{SfM} を用いて、各ピクセルの奥行値を比較することで環境地図と DSO のスケールの比率を取得する。

はじめに、DSO の入力画像に対して環境地図の入力画像 I^{SfM} から類似する画像の探索を行う。入力画像が DSO のキーフレーム $I^{\text{KF}} \subset I^{\text{VO}}$ として扱われた場合、そのフレームから得られた BoW ベクタ F^{KF} を用いて式 (1) を満たす環境地図内のフレームを取得する。

$$1 - \frac{1}{2} \left| \frac{F_i^{\text{SfM}}}{|F_i^{\text{SfM}}|} - \frac{F^{\text{KF}}}{|F^{\text{KF}}|} \right| > t_{\text{BoW}} \quad (1)$$

ここで t_{BoW} は BoW の類似画像検索結果のしきい値を表しており、 i は環境地図のフレームのインデックスを表している。

該当フレームが複数取得できた際には、スコアが高い順に上位数フレームを該当フレームとして使用することとする。該当したフレームに対して、 I_i^{SfM} と I^{KF} の間で ORB 特徴量を用いた特徴点マッチングを行うことによって 2D-2D の特徴点对応を得る。また I_i^{SfM} で得られた 2 次元特徴点と距離画像 D_i^{SfM} 、環境地図を作成する際に用いたカメラパラメータから世界座標系上での 3 次元点を取得する。 I^{KF} 上の一つの特徴点に対して、複数の I^{SfM} でマッチングする特徴点を得られた際、それぞれのマッチング点の世界座標系上での 3 次元位置を計算した後にそれらの点の重心の計算を行う。重心と各点の分散の計算を行い、値がしきい値以下であれば有効な 3 次元点として扱う。最後に I^{KF} から得られた特徴点と D^{SfM} から得られた 3 次元点から、3D-2D 対応によって PnP 問題を RANSAC を用いて解くことによって、環境地図に対応する I^{KF} での DSO のカメラ位置姿勢を得ることができる。

次に、2D-2D マッチングを得られた 2 次元点のうち、DSO から得られた距離画像 D^{VO} が値を持っている点を用いて、DSO と環境地図の相対スケール $s^{\text{VO} \rightarrow \text{SfM}}$ を得る。相対スケールの算出については後述する。事前マップを再利用することができる DSO を用いる際には、環境地図から得られた特徴点と DSO から得られる特徴点で同一のものをを用いている^{2), 17)}。しかし本手法では環境地図の作成に SIFT 特徴量を用いており、DSO では輝度値を直接参照している。そのため、環境地図と DSO から作成された 3 次元点群を同一のものとして扱うことができない。よって、このようなマッ

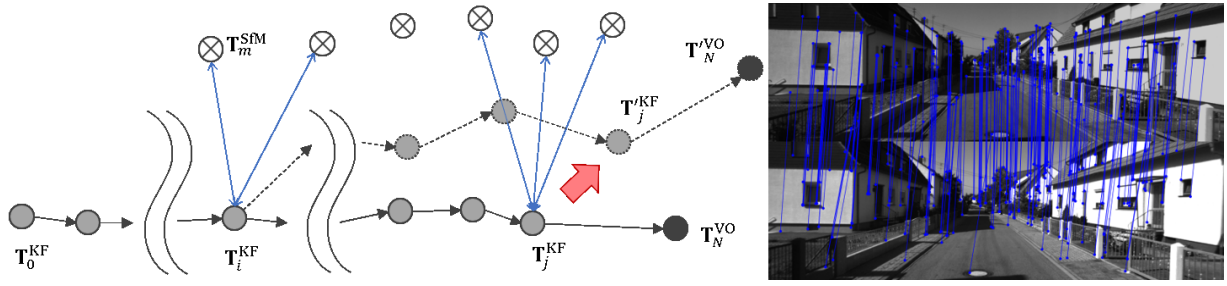


図3 環境地図を用いたグラフ最適化の概念図(左)と実際のマッチング結果(右)
Fig. 3 Graph optimization using an environment map (left) and a matching result (right)

グをベースとした距離画像による位置姿勢の再計算とスケールの補正が必要になる。

図2はVO座標系から環境地図座標系への座標変換を表している。ここでは、PnP問題を解くことで得られたカメラ位置姿勢を $\hat{\mathbf{T}}^{\text{KF}}$ 、DSOによって推定されたカメラ位置姿勢を \mathbf{T}^{KF} とする。DSOのカメラ座標系は、DSOから推定された距離画像 \mathbf{D}^{VO} の奥行値 d とDSOの画像座標 $\mathbf{W} = (u, v, 1)$ によって、 $\mathbf{t}_{\text{VO}} = d\mathbf{K}^{\text{VO}-1}\mathbf{W}$ と表すことができる。 \mathbf{K}^{VO} はDSOの入力画像を取得しているカメラの内部パラメータを表している。これを用いてSfMのカメラ座標系上への変換を式(2)を用いて表すことができる。

$$\begin{pmatrix} \mathbf{t}_{\text{SfM}} \\ 1 \end{pmatrix} \simeq \mathbf{T}^{\text{SfM}-1} \hat{\mathbf{T}}^{\text{KF}} \mathbf{T}^{\text{KF}-1} \begin{pmatrix} \mathbf{t}_{\text{VO}} \\ 1 \end{pmatrix} \quad (2)$$

ここで \mathbf{t}_{SfM} の第3成分は、SfM画像座標系における各ピクセルの奥行値を表しており、これを d_p^{VO} と表すこととする。対応するピクセルに対するSfMによって得られた奥行値を d_p^{SfM} とすると式(3)を用いて相対スケールを求めることができる。

$$s^{\text{VO} \rightarrow \text{SfM}} = \frac{1}{n} \sum_{p \in \Omega} \frac{d_p^{\text{SfM}}}{d_p^{\text{VO}}} \quad (3)$$

ただし、 $p \in \Omega$ は画像の2次元平面 Ω の中のピクセル p を表している。ここでピクセル p はPnP問題を解く際に2D-2Dマッチングを得ていてかつ \mathbf{D}^{VO} が値を持っているピクセルのみである。また条件を満たしているピクセル数の合計を n とする。DSOから得られる距離画像 \mathbf{D}^{VO} は距離センサなどで得られるものに比べて疎なものとなっている。DSOは入力画像にのみ依存して独立して動作をしているため、常にドリフト誤差が蓄積し続ける。そのため式(3)から得られる相対スケールの値はDSOの蓄積誤差によって変化するので一定ではない。よって、マッチングを行うたびに式(3)による計算を行い、相対スケールの値を更新する必要がある。

3.4 キーフレームにおける位置姿勢補正

ここでは、3.3節で $I^{\text{SfM}} \in \mathbf{I}^{\text{SfM}}$ とマッチングが得られず位置姿勢推定を行うことができなかったフレームに対して、 $s^{\text{VO} \rightarrow \text{SfM}}$ を用いて位置姿勢の補正を行う。位置姿勢の補正は、直近の3.3節の処理を行ったフレーム I_m^{KF} から現在の入

カフレーム I_n までに対する処理と、 I_m^{KF} とそれ以前に3.3節の処理を行ったフレーム I_{m-1}^{KF} との間のフレームに対する処理の2種類に分けられる。

前者に対して式(4)を適用しVO座標系から対応する環境地図座標系へと変換を行う。

$$\mathbf{T}_n^{\text{VO} \rightarrow \text{SfM}} := s^{\text{VO} \rightarrow \text{SfM}} \hat{\mathbf{T}}_m^{\text{KF}} \mathbf{T}_m^{\text{KF}-1} \mathbf{T}_n^{\text{VO}} \quad (4)$$

ここで、 $\mathbf{T}^{\text{VO} \rightarrow \text{SfM}}$ は、環境地図座標系でのカメラ位置姿勢、 \mathbf{T}^{VO} は、VO座標系でのカメラ位置姿勢を表している。またフレーム n で得られた奥行値に関しても $s^{\text{VO} \rightarrow \text{SfM}}$ によって環境地図のスケールへと補正を行う。

後者のフレーム群に対しては、グラフ最適化を用いたカメラ位置の補正を行っている。図3は提案手法でのポーズグラフの作成を表している。すべてのキーフレーム i のカメラ位置に対して、その直前 $i-1$ のカメラ位置とのエッジ $e_{i,i-1}$ の作成を行い以下の式のようなポーズグラフの作成を行う。

$$e_{i,i-1} = \left((\mathbf{T}_{i-1}^{\text{KF}})^{-1} \mathbf{T}_i^{\text{KF}} \right) \left((\hat{\mathbf{T}}_{i-1}^{\text{KF}})^{-1} \hat{\mathbf{T}}_i^{\text{KF}} \right) \quad (5)$$

マッチングが得られたフレームに関しては、 $\hat{\mathbf{T}}_i^{\text{KF}}$ はPnP問題を解くことによって得られた位置姿勢を表しており、 \mathbf{T}_i^{KF} は i 番目のDSOのカメラ位置姿勢を表している。g2oアルゴリズム²¹⁾を用いて次の式のポーズグラフを最適化する。ここでは $\mathbf{x} = (\mathbf{T}_1, \dots, \mathbf{T}_n)$ としたとき、 \mathbf{x} を最適化したカメラ位置姿勢である \mathbf{x}^* を求める。

$$\mathbf{F}(\mathbf{x}) = \sum_{\langle i \rangle \in N} e_{i,i-1}^T \Omega_{i,i-1} e_{i,i-1} \quad (6)$$

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \mathbf{F}(\mathbf{x}) \quad (7)$$

ここで N は最後にマッチングしたフレームを表しており、 $\Omega_{i,i-1}$ は情報行列を表している。式(7)をレーベンパーク・マーカート法を用いて解くことによってそれぞれのカメラ位置姿勢に対して最適化²¹⁾を行い、新たにカメラ位置姿勢群 \mathbf{x}^* を得る。

3.5 環境地図を利用した物体差分検出

環境地図内のラベル \mathbf{L}^{SfM} と3.4節によって補正されたカ

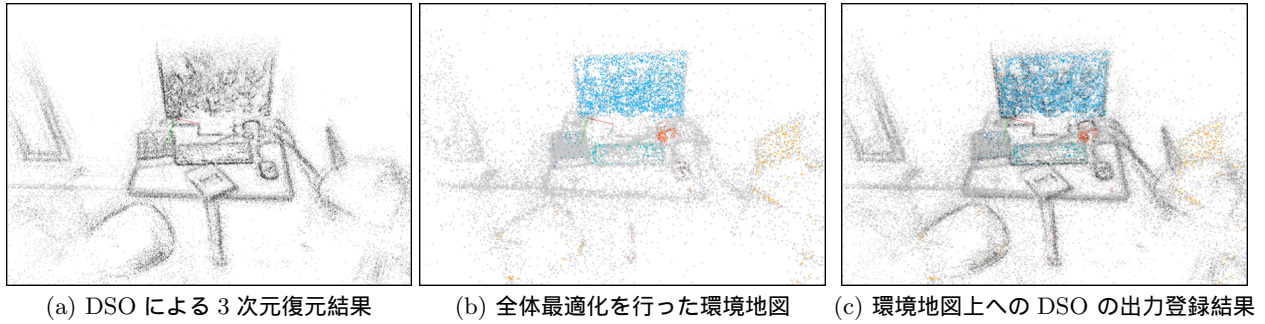


図4 提案手法による3次元復元結果

Fig. 4 3D reconstruction results of proposed method

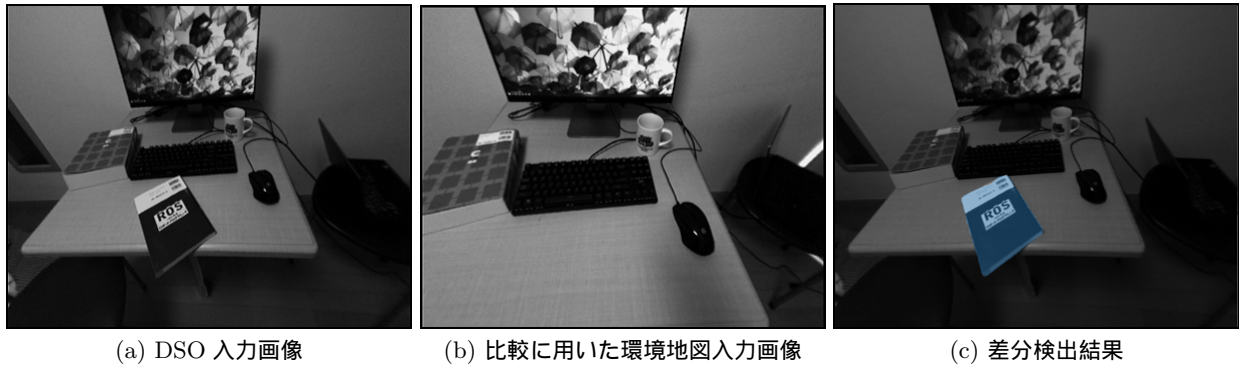


図5 提案手法とDSOとの差分

Fig. 5 Difference between the proposed method and DSO

メラ位置姿勢結果を用いて、3次元構造を利用した物体の差分検出を行う。DSOのキーフレーム $I^{KF} \subset I^{VO}$ に対して環境地図を構成するフレーム I^{SfM} の中から対象フレームとの距離のL2ノルムと画角の角度差がしきい値未満のフレーム $I^{Comp} \subset I^{SfM}$ を取得する。環境地図を作成する際に I^{Comp} を用いて作成された点群に注目し、それらの点群をフレーム I^{KF} 上に再投影する。また I^{KF} に対して個別物体領域分割を行い、各画素に対して個別物体領域分割ラベルを取得する。再投影されたラベルと取得したラベルとの差分を取得して、差分値がしきい値以上の際には新たに加えられたラベル、または取り除かれたラベルと判断する。これによって新たに環境に出現した物体や環境から取り除かれた物体の検出を行う。

4. 評価

本手法の有効性を示すために、定性的な評価として、自作したデータとパブリックデータセットにおいてあらかじめ作成した環境地図上でVOが動作することを示す。定量的な評価として、パブリックデータセットを用いてカメラ位置姿勢推定の精度を測定し、本手法によってVOの精度がSfMの精度に近づくことを示す。この実験結果を通して提案手法がVOの問題点となる蓄積誤差とドリフト誤差を軽減できていることを示す。また各処理に要する時間の測定を行った。最後に、個別物体領域分割を用いた差分検出に関して自作のデータセットを作成し、あらかじめ作成した環境地図に対して追加された物体と取り除かれた物体の検出を行った。

4.1 実装

本手法ではCOLMAP²²⁾を用いてSfMを行い環境地図Gを作成した。COLMAPの距離画像はフォトメトリックとジオメトリックの2通りの方法で作成されているが本手法ではジオメトリック手法によって得られた距離画像を用いている²³⁾。それに加えて各フレームに対してORB特徴量の計算を行い、DBoW3を用いて画像データベースの作成を行った。さらに各フレームに対してYolact²⁴⁾を用いて個別物体領域分割を行いフレームごとに物体識別ラベルを作成した。撮影した際のカメラ位置姿勢に対するドリフト誤差を軽減するために撮影画像は画像シーケンスの中から五枚に一枚の量で間引いてCOLMAPの入力とした。

本手法では、3.3節で説明したように、距離画像を介して環境地図に対するVOのスケールを算出している。そのため、他の手法と比較して密に奥行推定を行うことができるDSOをVOの基礎として選択した。これを以下の2点において拡張した。まず、BoWを用いたマッチングのためのスレッド、ポーズグラフ最適化のためのスレッド、差分検出のためのスレッドを追加した。次に、ポーズグラフ最適化のためにDirect Sparse Odometry with a Loop Closure (LDSO)³⁾のループクローザーの箇所と類似したフレームワークを用いて、g2oによるグラフ最適化を行っている。

4.2 環境地図上へのVOの投影結果

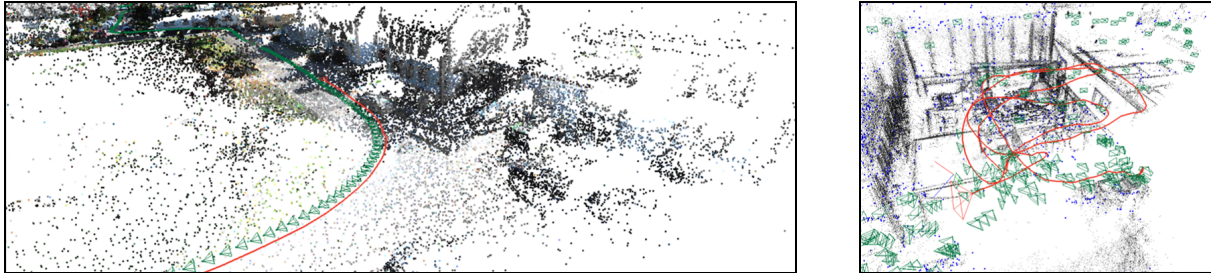
図4から図7は本手法を用いて環境地図上にVOの推定結果を投影したものである。図4は屋内の自作データに対して処



(a) LIDAR による 3 次元点群を入力画像に投影した結果 (b) (a) にバイラテラルフィルタリングして生成した距離画像

図 6 KITTI データセットの LIDAR 点群からの距離画像作成結果

Fig. 6 Output of depth image from LIDAR point cloud of KITTI dataset



(a) KITTI データセットにおける LIDAR で作成した環境地図での提案手法の出力結果 (b) EuRoC データセットにおける連続していない画像群に対する提案手法の出力結果

図 7 提案手法による軌跡の出力結果

Fig. 7 Output results of the trajectory by the proposed method

理を行った結果で、環境地図上 (図 4(b)) での DSO (図 4(a)) の実行結果 (図 4(c)) を示している。環境地図は物体識別ラベルによって色が付随しており、DSO への入力画像に対する個別物体領域分割の結果と環境地図との差分から環境における物体の差分検出を行っている (図 5)。図 5 では画像 (a) の DSO 入力画像に対して、環境地図入力画像として用いられた画像 (b) と比較すると、画像 (c) において本が新たに環境に現れた物体として検出されていることを示している。図 7 はそれぞれ KITTI データセット¹⁸⁾と EuRoC MAV データセット²⁵⁾に対して処理を行った結果である。KITTI データセットは車載のステレオカメラで街を走行した大規模なデータセットである。KITTI データセットには真値を持つ訓練用のシーケンス (00 ~ 10) と真値を持たないテスト用のシーケンス (11 ~ 21) が含まれている。EuRoC MAV データセットではステレオ画像、IMU データ、カメラの軌跡の真値を含む 11 の画像シーケンスが提供されている。画像シーケンスは異なる 3 種類の環境で撮影されており、Machine Hall が 5 種類のシーケンス (MH01 ~ MH05)、Vehicon Room 1 が 3 種類 (V101 ~ V103)、Vehicon Room2 が 3 種類 (V201 ~ V203) それぞれ提供されている。

KITTI データセットでは、提案手法のより広い汎用性を示すために SfM により事前に生成した環境地図を用いる代わりに、LIDAR と GPS によって測定された結果を環境地図に用いた。LIDAR によって計測された環境の 3 次元点群を入力画像に投影 (図 6(a)) し、バイラテラルフィルタ²⁶⁾を用いて距離画像 (図 6(b)) を作成した。この距離画像と GPS のデータから環境地図を作成した。ステレオ画像のシーケンスであるので左カメラの画像を環境地図の入力に、右カメラ

の画像を DSO の入力として提案手法を実行した。その結果として図 7(a) が得られた。また、EuRoC MAV データセットでは、Machine Hall の 5 種類の画像シーケンスからランダムに画像を 350 枚選び、連続していない画像群を入力として SfM を用いて環境地図を作成した。図 7(b) はその結果を表している。図 7(a),(b) ではそれぞれ緑の四角錐が環境地図のカメラの位置姿勢を、赤線が提案手法の軌跡を表している。

また提案手法の各処理機能が要する処理時間の測定を行い、表 1 にまとめた。本手法は三つのスレッドから構成されている。DSO の結果を受け取りその他二つのスレッドに出力を割り振るメインスレッド、3.3 節と 3.4 節の処理を行うマッチングスレッド、出力を表示するビューアスレッドから成る。1 フレームを処理するのに要する時間はメインループの処理速度と等価である。メインループが処理を行う上で、類似画像の抽出に成功したときマッチング処理が開始され、表 1 のマッチングにまとめられた処理が行われる。MH01 を環境地図にして MH02 を DSO の入力として全 3040 フレームに対して測定を行った。3040 フレームの中で 331 回マッチング処理が起動している。今回の実験では Intel Core i7-7700、3.60GHz のプロセッサとメモリ 32GB、NVIDIA GeForce GTX 1060 のグラフィックカードから成るデスクトップパソコンを用いている。本手法ではデータ量が増えた際、BoW の類似画像検索にかかる時間、点群のレンダリング、オフラインでの環境地図の作成とその読み込みにかかる時間が増える。そこで KITTI データセットのシーケンス 00、4541 枚の画像に本手法を適用した。ステレオカメラの右側の映像で環境地図を作成し、左側の映像をオンラインでの入力とした。その結果、トラッキングの処理速度の平均は 1 フレーム当た

り $87.8 \pm 38.8 \text{ msec}$ であった．処理時間に加えてこのシーケンスでは環境地図として 11.9GB のデータをロードしている．より多くの画像を扱って環境地図を作成する際は，マッチングスレッドの完了速度が遅くなるので，3.3 節の処理に関して，より時間がかかることが想定される．また点群のレンダリングを行わないことや，参照する環境地図を場所によってロードして，切り替えるといった実装をしてメモリの使用量を抑える必要があると考えられる．

4.3 Sim(3) 補正を用いた APE 結果

EuRoC MAV データセットを用いて提案手法の精度を検証した．それぞれの撮影環境に対して，一つのシーケンスを環境地図の作成に，その他のシーケンスを VO の入力として使用して実験を行った．いずれの画像シーケンスにおいてもステレオ画像の右側の画像のみを用いて環境地図の作成，VO への入力を行った．

本手法の有効性を示すために Absolute Pose Error (APE) の測定をそれぞれのシーケンスでの実行結果に対して行った．ここでは，入力画像に対して推定された位置姿勢と真値の位置姿勢の比較を行い，回転行列と並行移動ベクトルの平均二乗誤差を結果として表している．APE の評価は最適化を行う DSO のキーフレームに対してのみ行った．本手法，DSO，Colmap の位置姿勢推定結果に対して，梅山補正²⁷⁾による Sim(3) 補正をおこなって位置姿勢とスケールの最適化を行った．

表 1 提案手法の各機能処理時間
Table 1 Processing time of each component

スレッド	処理	処理回数 (回)	時間 (msec)
メイン	メインループ	3040	122.76
マッチング	特徴点マッチング	331	203.59
	カメラ位置推定	313	70.39
	グラフ最適化	10	2.23

表 2 は EuRoC MAV データセットにおける Machine Hall と Vehicon Rooms 1, 2 の結果を表している．表中の × 印は十分なマッチングを取得できずシステムが起動できなかったことを表している．結果として，提案手法では環境地図とのマッチングによってドリフト誤差が軽減されるため，全てのシーケンスにおいて DSO よりも高い精度でカメラの位置姿勢推定を行うことができた．提案手法は類似画像検索から得られるマッチング結果に依存して，環境地図上での DSO のカメラ位置を推定している．そのため類似した視点から撮影された画像が得られない場合，期待した結果を得ることができない．表 2 の Vehicon Room2 の結果において，V201 と V203 の組み合わせは互いにマッチングを得ることができなかったのもシステムが起動しなかった．また Machine Hall に関して，MH01 と MH02，MH04 と MH05 はそれぞれ似たような視点からの撮影を行っているため，互いに他方を環境地図にしたときの結果が最もよくなっていることがわかる．なお，表 2 における (Colmap) は Colmap によって推定されたカメラ位置姿勢と真値との APE 測定結果を表している．

また図 8 は，MH01 を環境地図にして，本手法を用いて五つの Machine Hall シーケンスすべてに対して位置合わせとスケール合わせを行い，同一のスケールで表した結果である．各色ごとに一つのシーケンスの軌跡推定結果を表しており，破線は真値による軌跡を梅山補正によって重畳したものである．このように同様の環境地図を用いて提案手法の軌跡推定を行うことによって，異なる時系列で撮影されたシーケンスを同一の環境地図上に，Sim(3) 補正を行うことなく投影できていることがわかる．

さらに環境地図を保存して，保存した環境地図上で自己位置姿勢推定を行いながら SLAM を実行する機能を持っている OpenVSLAM²⁸⁾との比較検討を行った．KITTI データ

表 2 EuRoC データセットにおける APE 測定結果 (RMSE)

Table 2 APE in the EuRoC dataset (RMSE)

シーケンス	提案手法					DSO	(Colmap)
環境地図	MH01	MH02	MH03	MH04	MH05	—	—
VO	—	0.046	0.069	0.058	0.069	0.280	(0.031)
MH01	—	0.029	—	0.049	0.074	0.270	(0.017)
MH02	0.124	0.173	—	0.254	0.185	0.630	(0.036)
MH03	0.249	0.258	0.344	—	0.068	0.572	(0.049)
MH04	0.104	0.150	0.272	0.085	—	0.509	(0.045)
環境地図	V101	V102	V103	—	—	—	—
VO	—	0.148	0.200	—	—	0.344	(0.141)
V101	0.080	—	0.094	—	—	0.631	(0.049)
V102	0.830	1.242	—	—	—	1.009	(0.063)
環境地図	V201	V202	V203	—	—	—	—
VO	—	0.105	×	—	—	0.262	(0.030)
V201	0.142	—	0.052	—	—	0.589	(0.025)
V202	×	1.544	—	—	—	1.611	(0.029)

セットのシーケンス 00 に対して、自己位置姿勢推定のための評価を行うために、OpenVSLAM のループ機能が動作しない最初の 1000 フレームに関して実験を行った。ステレオカメラの左側カメラ画像を環境地図の入力、右側画像を DSO の入力とした。またその際、環境地図に用いる画像群を全フレームから 3 フレームに 1 枚、5 フレームに 1 枚、7 フレームに 1 枚と間引いて実験を行った。この結果を表 3 に示す。OpenVSLAM では環境地図を SLAM を使って作成しているためフレームを間引いてしまうと環境地図を作成することができず測定を行うことができなかった。それに対して提案手法ではフレームを間引いた際にも環境地図を作成することが

できていることがわかる。提案手法ではこのように連続でないフレームを環境地図として扱うときに有効であることがわかる。

4.4 物体の差分検出実験

自作のデータセットに対して環境地図を作成して本手法を適用した。その後、環境に新たに物体を加えて、環境に加わった物体が検出できるかを評価した。図 9 は検出結果を表しており、画像 (a) が DSO への入力画像、画像 (b) が環境地図で参照されたフレームの画像である。それに対して画像 (c) は物体の差分検出結果であり、上段では環境に新たに追加されたノートパソコンが、下段では環境から取り除かれたキーボードが検出されている。

5. む す び

本論文では SfM を用いて事前に作成した環境地図に合わせた VO によるカメラ軌跡の推定手法とその利用方法に関して提案した。提案手法は、SfM を用いて作成した環境地図を BoW データベースによって参照することにより、VO の一つである DSO によって推定されたカメラ位置姿勢の補正を行う。そして、補正されたカメラ位置姿勢を含めた DSO の軌跡からポーズグラフを作成し、グラフ最適化を行うことによって DSO によって求められた軌跡を補正する。そのため、本手法は単眼カラーカメラのみで利用でき、学習データも不要であるという利点を有する。また補正された環境地図上で軌跡をもとに、3 次元構造を用いた物体の差分検出を行った。

提案手法の有効性を示すために自作した複数のシーケンスで環境地図に合わせたスケールでの DSO の実行結果を示した。また定量的な評価として、既存のデータセットを用いて、DSO のみを用いた場合に比べて提案手法による補正を行っ

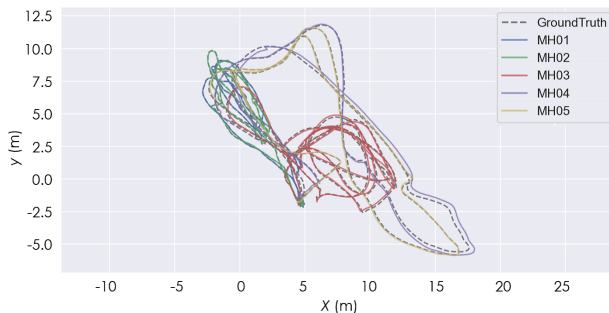


図 8 EuRoC Machine Hall データセットの全シーケンスに対して、同一の環境地図を用いて軌跡推定を行った結果

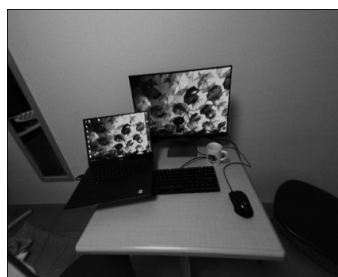
Fig. 8 Results of trajectory estimation using the same environment map for all sequences of the EuRoC Machine Hall dataset

表 3 OpenVSLAM との APE 測定比較結果 (RMSE)

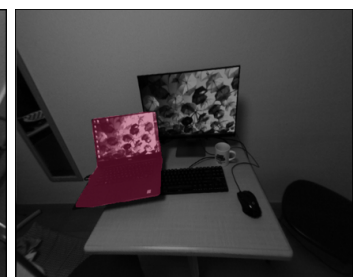
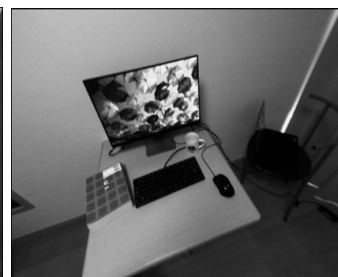
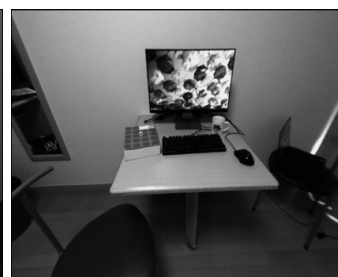
Table 3 Comparison of APE measurements with OpenVSLAM (RMSE)

手法	1	1/3	1/5	1/7
提案手法	6.91	2.71	1.31	×
OpenVSLAM	8.05	×	×	×

(i) 環境に新たに追加されたノートパソコンの検出



(ii) 環境から取り除かれたキーボードの検出



(a) VO 画像

(b) 環境地図画像

(c) 検出結果

図 9 個別物体領域分割による差分検出結果

Fig. 9 Change detection by Instance Segmentation

たほうが高い精度の位置姿勢推定を行うことができることを示した。また、差分検出の具体例として、物体識別ラベルの差分を用いて環境に新たに追加された物体や環境から取り除かれた物体の検出に本手法が有用であることを示した。

参考文献

- 1) T. Taketomi, H. Uchiyama, S. Ikeda: “Visual SLAM Algorithms: a Survey from 2010 to 2016”, IPSJ Trans. on Computer Vision and Applications, Vol. 9, No. 1, pp.16 (2017).
- 2) R. Mur-Artal, J. D. Tardós: “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras”, IEEE Trans. on Robotics, Vol. 33, No. 5, pp.1255–1262 (2017).
- 3) X. Gao, R. Wang, N. Demmel, D. Cremers: “LDSO: Direct Sparse Odometry with Loop Closure”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018), pp.2198–2204 (2018).
- 4) C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza: “IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation”, Proc. of Robotics Science and Systems, Vol. 13, pp.136 (2015).
- 5) K. Tateno, F. Tombari, I. Laina, N. Navab: “CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction”, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), pp.6243–6252 (2017).
- 6) N. Yang, R. Wang, J. Stuckler, D. Cremers: “Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry”, Proc. of European Conference on Computer Vision (ECCV 2018), pp.817–833 (2018).
- 7) T. Taketomi, T. Sato, N. Yokoya: “Real-Time and Accurate Extrinsic Camera Parameter Estimation using Feature Landmark Database for Augmented Reality”, Computers & Graphics, Vol. 35, No. 4, pp.768–777 (2011).
- 8) J. Ventura, T. Höllerer: “Wide-Area Scene Mapping for Mobile Visual Tracking”, Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012), pp.3–12 (2012).
- 9) J. Engel, V. Koltun, D. Cremers: “Direct Sparse Odometry”, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 40, No. 3, pp.611–625 (2017).
- 10) C. Forster, M. Pizzoli, D. Scaramuzza: “SVO: Fast Semi-Direct Monocular Visual Odometry”, Proc. of IEEE International Conference on Robotics and Automation (ICRA 2014), pp.15–22 (2014).
- 11) J. Engel, T. Schöps, D. Cremers: “LSD-SLAM: Large-Scale Direct Monocular SLAM”, Proc. of European Conference on Computer Vision (ECCV 2014), pp.834–849 (2014).
- 12) T. Qin, P. Li, S. Shen: “Relocalization, Global Optimization and Map Merging for Monocular Visual-Inertial SLAM”, Proc. of IEEE International Conference on Robotics and Automation (ICRA 2018), pp.1197–1204 (2018).
- 13) T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, R. Siegwart: “Maplab: An Open Framework for Research in Visual-Inertial Mapping and Localization”, IEEE Robotics and Automation Letters, Vol. 3, No. 3, pp.1418–1425 (2018).
- 14) J. Ventura, C. Arth, G. Reitmayr, D. Schmalstieg: “Global Localization from Monocular SLAM on a Mobile Phone”, IEEE Trans. on Visualization and Computer Graphics, Vol. 20, No. 4, pp.531–539 (2014).
- 15) J. Ventura, T. Höllerer: “Wide-Area Scene Mapping for Mobile Visual Tracking”, Proc. of IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012), pp.3–12 (2012).
- 16) H. Kume, A. Suppé, T. Kanade: “Vehicle Localization along a Previously Driven Route Using an Image Database”, Proc. of IAPR Conference on Machine Vision Applications (MVA 2013), pp.177–180 (2013).
- 17) R. Castle, G. Klein, D. W. Murray: “Video-Rate Localization in Multiple Maps for Wearable Augmented Reality”, Proc. of IEEE International Symposium on Wearable Computers, pp.15–22 (2008).
- 18) A. Geiger, P. Lenz, R. Urtasun: “Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite”, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), pp.3354–3361 (2012).
- 19) D. Gálvez-López, J. D. Tardós: “Bags of Binary Words for Fast Place Recognition in Image Sequences”, IEEE Trans. on Robotics, Vol. 28, No. 5, pp.1188–1197 (2012).
- 20) E. Rublee, V. Rabaud, K. Konolige, G. R. Bradski: “ORB: an Efficient Alternative to SIFT or SURF”, Proc. of IEEE International Conference on Computer Vision (ICCV 2011), pp.2564–2571 (2011).
- 21) R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard: “g2o: A General Framework for Graph Optimization”, Proc. of IEEE International Conference on Robotics and Automation (ICRA 2011), pp.3607–3613 (2011).
- 22) J. L. Schönberger, and J. M. Frahm: “Structure-from-Motion Revisited”, Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp.4104–4113 (2016).
- 23) J. L. Schönberger, E. Zheng, J. M. Frahm, M. Pollefeys: “Pixelwise View Selection for Unstructured Multi-View Stereo”, Proc. of European Conference on Computer Vision (ECCV 2016), pp.501–518 (2016).
- 24) D. Bolya, C. Zhou, F. Xiao, Y. J. Lee: “YOLACT: Real-time Instance Segmentation”, Proc. of IEEE International Conference on Computer Vision (ICCV 2019), pp.9157–9166 (2019).
- 25) M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, R. Siegwart: “The EuRoC Micro Aerial Vehicle Datasets”, The International Journal of Robotics Research, Vol. 35, No. 10, pp.1157–1163 (2016).
- 26) C. Premevida, L. Garrote, A. Asvadi, A. P. Ribeiro, U. Nunes: “High-Resolution LIDAR-Based Depth Mapping Using Bilateral Filter”, Proc. of IEEE International Conference on Intelligent Transportation Systems (ITSC 2016), pp.2469–2474 (2016).
- 27) S. Umeyama: “Least-Squares Estimation of Transformation Parameters between Two Point Patterns”, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 13, No. 4, pp.376–380 (1991).
- 28) S. Sumikura, M. Shibuya, K. Sakurada: “OpenVSLAM: a Versatile Visual SLAM Framework”, Proc. of The 27th ACM International Conference on Multimedia, pp.2292–2295 (2019).

(2020年5月29日 受付)

(2021年1月5日 再受付)



山口 真 弘

2016 年 慶應義塾大学理工学部情報工学科卒業 .
2021 年 同大学院理工学研究科開放環境科学専攻
博士課程修了 . 2021 年 日本電気株式会社入社 .
博士 (工学) . コンピュータビジョンやその応用
に関する研究に従事 .



森 尚 平

2016 年 立命館大学大学院情報理工学研究科博士
課程修了 . その後 , 慶應義塾大学理工学部にて
日本学術振興会特別研究員 (PD) . 現在 , 2018
年よりグラーツ工科大学 ICG ポストドクトラル
フェロー . 博士 (工学) . 拡張現実感及びヒュー
マンインタフェースに関する研究に従事 .



斎 藤 英 雄 (正会員)

1987 年 慶應義塾大学理工学部電気工学科卒業 .
1992 年 同大学院理工学研究科電気工学専攻博士
課程修了 . その後 , 同大学助手 , 専任講師 , 助教
授を経て 2006 年 より教授 . 博士 (工学) . この
間 , 1997 年から 99 年 までカーネギーメロン大
学ロボティクス研究所訪問研究員 . コンピュータ
ビジョンとその応用に関する研究に従事 .



谷内田 尚 司

1988 年 日本電気ホームエレクトロニクス株式会
社入社 , 2000 年 日本電気株式会社入社 , 画像処
理回路設計技術及び研究開発に従事 , 現在同社 ,
バイオメトリクス研究所 , 主任 .



柴 田 剛 志

2008 年 日本電気株式会社入社 . 2017 年 東京工
業大学大学院 , 理工学研究科博士課程修了 (工学
博士) . 2020 年 日本電信電話株式会社入社 . 画
像認識 , 画像処理及びコンピュータービジョン
の研究開発に従事 . 現在 , 日本電信電話株式会
社 , コミュニケーション科学基礎研究所 , 主任研
究員 .