THE IEICE TRANSACTIONS ON INFORMATION AND SYSTEMS (JAPANESE EDITION)

電子情報通信学会 論文誌 D 情報・システム

VOL. J104-D NO. 4
APRIL 2021

情報・システムソサイエティ

一般社団法人 電子情報通信学会
THE INFORMATION AND SYSTEMS SOCIETY
THE INSTITUTE OF ELECTRONICS, INFORMATION AND COMMUNICATION ENGINEERS

移動RGBDカメラによる複数移動物体の実時間追跡と3次元形状再構成

八馬　遼[†]　Christian Pirchheim[††]　斎藤　英雄[†]
Dieter Schmalstieg[††]

Real-Time Multiple Object Tracking and Reconstruction with a Moving RGBD Camera

Ryo HACHIUMA[†], Christian PIRCHHEIM[††], Hideo SAITO[†], and Dieter SCHMALSTIEG[††]

あらまし　移動カメラ画像列から環境の3次元幾何構造とカメラの移動軌跡と姿勢を同時に推定するSLAMのほとんどは，環境に動く物体が存在しないという仮定を置いているため，移動物体が環境にあるとシステムが破綻する．本研究では，移動物体が環境下に複数あった状態でもSLAMを行い，かつ各移動物体の追跡と3次元形状再構成を実現する方法を提案する．本手法では，RGB画像からの物体検出と距離画像からの幾何学的分割結果を組み合わせ，画像を高速に個々の物体ごとに分割することで，複数の移動物体の3次元形状再構成と追跡を行う．また，物体検出により未検出だった移動物体をカメラ追跡から除外することで，高精度にカメラ追跡を行う．実験では，提案手法の有効性を検証するために，動的環境下でのカメラ追跡精度，移動物体の3次元形状再構成精度，実行時間を既存手法と比較する実験を行い，提案手法が動的環境下で高精度にカメラ追跡を行え，かつ高速に複数の移動物体を3次元形状再構成できることを確認した．

キーワード　SLAM，RGBD，物体検出，動的環境下，3次元形状再構成

1. まえがき

SLAM (Simultaneous Localization And Mapping) とは，RGB [1], [2] またはRGBD (RGB + 距離) [3]〜[5]の画像群を入力とし，カメラの位置と姿勢を推定しながら環境の3次元地図を再構成するタスクであり，自動運転や拡張現実感 (AR) の実現のための重要な技術のうちの一つである．ほとんどのSLAMシステムは環境に移動物体が存在しない，つまり環境は静的であるという仮定のもと成り立っている．

しかし，実環境においてこの仮定が成り立つことは少ない．例えば自動運転のアプリケーションを考えたときに，車載カメラには対向車であったり，自転車や歩行者など様々な物体が映ることが想定される．また，ARアプリケーションでも例えば人がインタラクティブに物体を動かしAR表示したいことも容易に想定される．もしそのような移動物体が環境内に存在する場合，システムは画像中に映った動きがカメラ自身によるものなのか，物体そのものが動いているのか判断がつかず，カメラの追跡に失敗してしまう．

移動物体を考慮したSLAMは目的別に主に3カテゴリーに分類される．まず一つ目は，複数の移動物体が存在するような動的環境下でカメラの位置姿勢推定と静的な背景の3次元再構成を目的としたSLAM [6]〜[8]であり，例えばロボットや自動運転における自己位置姿勢推定のアプリケーションを想定している．このSLAMでは，移動物体の画素を画像中からsegmentationし外れ値とすることで達成される．二つ目は，人や布など形状が変化する非剛体な物体が存在するような環境において，その物体形状とカメラ位置姿勢推定することを目的としたSLAMであり[9]〜[11]，人体の3次元形状計測などのアプリケーションを想定している．最後に，三つ目は複数の物体が独立に移動しているような動的環境下において，それぞれの物体の位置姿勢を推定，3次元形状を再構成しつつ，カメラの

[†] 慶應義塾大学，横浜市
Keio University, Yokohama-shi, 223–8522 Japan
[††] グラーツ工科大学，オーストリア
Graz University of Technology, Graz, Steiermark, Republic of Austria
DOI:10.14923/transinfj.2020PDP0019

位置姿勢推定と静的な背景の 3 次元再構成を目的とした SLAM である [12]〜[16]．この SLAM では，例えば AR において移動する複数の物体の上に文字や画像などを重畳表示するようなアプリケーションで用いられる．本研究では，このような AR などへの応用を考え，この三つ目の SLAM に着目する．

一方，近年，深層学習技術の発展により，シーン理解のために SLAM から得られる 3 次元地図に対して意味情報を付与する研究が盛んに行われている [17]〜[19]．Semantic segmentation 技術や Instance segmentation 技術は Convolutional Neural Network (CNN) を用いて精度よく行われる [20], [21] ため，センサから得られる画像に対して segmentation を施し，環境地図にその意味情報も付与することで，意味ラベルがついた 3 次元地図が生成される．

既存の複数移動物体の 3 次元形状再構成を目的とした SLAM の多く [13]〜[15] では，instance segmentation 手法である Mask R-CNN [21] を用いて RGBD 画像を物体ごとに segmentation し，検出された物体の画素を背景の SLAM から除外し，物体毎で地図を作成し追跡を行っていた．しかし，Mask R-CNN はその用いているネットワークの大きさゆえ実時間で動かない．ドローンやスマートフォンに実装することを考慮すると，より軽量なネットワークで動くことが望ましい．よって本研究では，より軽量な計算資源でかつ実時間で動作する動的 SLAM を目指す．また，Mask R-CNN が物体を検出できなかった場合，その物体を segmentation することができず，背景の 3 次元モデルに統合され，カメラの追跡精度に悪影響を及ぼしていた．よって，本研究では未検出の移動物体を別途 Motion segmentation を施すことで，カメラの位置姿勢推定精度向上を目指す．

以上のような背景に基づき，本研究では，毎フレーム instance segmentation を行いつつ軽量な計算資源で実時間に動作し，かつ未検出の物体をカメラの位置姿勢推定から除外する dynamic RGBD SLAM を提案する．まず実時間で Instance segmentation を行うために，提案手法では既存手法のボトルネックであった Mask R-CNN を用いず，実時間で動作する物体検出手法の YOLOv3 [22] を用いる．YOLOv3 は高速に動作し (30Hz)，かつ精度よく物体が検出できる（MS COCO データセットにおいて 40mAP）ことが知られている．しかし，物体検出タスクでは RGB 画像を入力とし物体のおおよその位置を表す矩形のみしか得られないため，背景領域も物体として再構成されてしまう．よって，本研究では距離画像に対して幾何学的な (geometric) segmentation [23] を施すことで画像を凸面ごとに分割し，物体検出の結果と組み合わせることで個々の物体ごとに画像を segmentation (instance segmentation) する．また，motion segmentation でもこの幾何学的 segmentation 結果を活用し，カメラの位置姿勢推定をした際の誤差画像と組み合わせることで未検出だった移動物体をカメラの位置姿勢推定から除外する．

実験では，まず TUM RGBD Dataset [24] を用いて動的環境下でのカメラの位置姿勢推定精度を評価する．次に合成データセットを用いて移動物体の 3 次元形状再構成精度を評価し，最後に本手法の実行速度を評価する．本手法の貢献点は，既存手法より軽量な計算資源でかつ実時間で動作し，motion segmentation により未検出の物体をカメラ追跡から除外することで追跡精度を向上させた dynamic RGBD SLAM を提案したことである．

2. 関 連 研 究

本章では，RGBD 画像を入力とし動的環境下で密に 3 次元再構成をする既存の SLAM (dynamic SLAM) と提案手法との差異を明らかにし，比較することで本論文の位置づけを明らかにする．まず本研究では，dynamic SLAM のうち非剛体物体の 3 次元形状再構成することを目的とした研究 [9]〜[11] とは異なり，人などの動的な非剛体物体は追跡，3 次元形状再構成から除外する．また，小澤らの研究 [25] では，RGBD センサから幾何学的に距離画像を分割し，各領域で位置姿勢推定を行うことで，物体の 3 次元形状を復元していた．しかし，SLAM を行っていないため，背景の 3 次元地図を復元することができず，また凸状の物体しか形状の再構成ができなかった．

関連手法との差異を項目ごとに表 1 にまとめる．ElasticFusion [4] は環境が静的であると仮定し，RGBD 画像を入力とした SLAM であり，動的環境下で動作する SLAM ではないが，参考のために表に示している．StaticFusion [6] は，カメラ位置姿勢推定と移動物体の segmentation を同時に行うことで，移動物体を位置姿勢推定から除外しながら，SLAM を行っていた．この論文では，移動物体の追跡，3 次元形状再構成を目的としておらず外れ値として画像から除外していた．Co-Fusion [12] では，カメラの位置姿勢を推定した際

表 1　関 連 研 究

関連研究	移動物体の 3 次元形状再構成	Instance segmentation	Motion segmentation	実時間性
ElasticFusion [4]				✓
StaticFusion [6]			✓	✓
Co-Fusion [12]	✓		✓	✓
MaskFusion [13]	✓	Mask R-CNN (キーフレーム)		✓
MID-Fusion [14]	✓	Mask R-CNN (オフライン)	✓	
EM-Fusion [15]	✓	Mask R-CNN (オフライン)		
紺野ら [27]	✓	Mask R-CNN (オフライン)		
提案手法	✓	YOLOv3 + geom.segm.	✓	✓

の誤差画像に対して Dense CRF [26] を適用することで，画像を個々の移動物体領域に分割し，個々の物体での追跡と 3 次元形状再構成を行っていた．この研究では，3 次元環境地図に対して意味情報を付加しておらず，作成される地図には色情報しか付与されていない．紺野らの研究 [27] では，移動物体は音を発しているという仮定に基づき音を発している物体の位置をマイクから推論することで，各移動物体で 3 次元形状再構成を行っていた．しかし，各物体が移動する際に発される音をセンシングするためには高性能なマイクが必要であり論文では仮想環境でのみ実験を行っていたため，ノイズが多く含まれる実環境で応用することは困難である．

MaskFusion [13], MID-Fusion [14], EM-Fusion [15] は，全て Mask R-CNN を用いて，画像を個々の物体に分割し (instance segmentation)，個々の物体に追跡をすることで，物体の形状再構成と動的環境下での SLAM を実現させていた．実時間での動作性が求められる SLAM にとって，1 枚当たりの画像を処理するのに数秒かかってしまう Mask R-CNN を適用することは難しい．MaskFusion では，キーフレームにのみ適用していたため，高速に動作するカメラや物体の動きには対応せず，MID-Fusion と EM-Fusion では，あらかじめオフラインに毎フレームに対して Mask R-CNN を適用していたため，オンラインでは動作しない．本研究では，既存手法の実行時間のボトルネックであった Mask R-CNN により画像を個々の物体領域に分割するのではなく，RGB 画像からの物体検出と距離画像の幾何学的 segmentation を組み合わせることにより実現する．

また，MaskFusion では，移動物体の分割は Mask R-CNN のみを用いて行っていたため，Mask R-CNN が検出できなかった物体，例えばモデルの推定誤差や物体のカテゴリーが学習されたデータセットに含まれていない，は静的な背景領域として統合される．本研究では，検出されなかった移動物体が環境の 3 次元地図

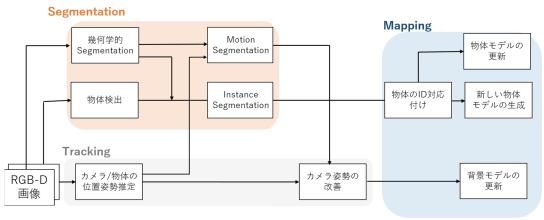

図 1　提案手法の流れ

に統合され，カメラの位置姿勢推定に悪影響を及ぼすことを防ぐために，それらの画素を motion segmentation によって分割する．その際，motion segmentation を適用した既存研究 [14] とは異なり，幾何学情報を考慮した motion segmentation を行うことで，部分的にしか segmentation されない問題を解決する．

本論文では，DetectFusion [28] に基づき，複数の移動物体がある環境下でカメラの自己位置推定・静的な環境の 3 次元形状再構成，そして各移動物体の位置姿勢推定と 3 次元形状再構成を行う手法を提案する．

3. 提 案 手 法

提案手法の流れを図 1 に示す．提案手法は，カメラ，物体位置姿勢を推定する tracking モジュール，画像を個々の物体ごとに分割する segmentation モジュール，3 次元地図にフレーム情報を追加していく mapping モジュールから成る．Tracking モジュールでは，現在フレームにおけるカメラ，そして各物体の位置姿勢を推定する．segmentation モジュールでは，RGBD 画像に対して，各物体ごと，そして未検出だった移動物体に画像を分割する．そして，mapping モジュールでは，静的な背景の 3 次元点と各物体の 3 次元点をそれぞれの地図に統合していく．各々のモジュールについて詳しく説明する．

3.1 Tracking

カメラ，そして各物体（合わせてモデル M_m と定義する．m は m 番目のモデルの番号を表す）の位置姿勢 ψ_m を RGBD センサから取得された現在の輝度画

像 I_t，距離画像 D_t から求める．ここで，t はフレーム番号を表す．各物体は独立に動くと考えるため，各モデルの位置姿勢推定はモデルごとに行う．位置姿勢推定は下式の関数 E_m をモデルごとに最小化することにより求められる．

$$E_m = \min_{\psi_m}(E_m^{geo} + \lambda E_m^{pho}), \tag{1}$$

ここで，E_m^{geo} は幾何学的な誤差項，E_m^{pho} は輝度の誤差項であり，λ はその 2 項の重みを調整するハイパーパラメータである．ψ_m は位置姿勢パラメータであり，リー代数で表現されている．

まず 1 項目の幾何学的な誤差項では，現在の距離画像 D_t と m 番目の 3 次元地図を $t-1$ フレームに投影したときの距離画像 $\hat{D}_{t-1}$ 間の 3 次元点同士の距離が最も小さくなるような位置姿勢パラメータ ψ_m を求めることを目的としており，

$$E_m^{geo} = \sum_k ((\boldsymbol{v}^k - \exp(\psi_m)\boldsymbol{v}_t^k)\cdot \boldsymbol{n}^k)^2, \tag{2}$$

として表される．ここで，$\boldsymbol{v}_t^k$ は距離画像 D_t の k 番目の 3 次元点であり，$\boldsymbol{v}^k, \boldsymbol{n}^k$ はその 3 次元点に対応した距離画像 $\hat{D}_{t-1}$ の 3 次元点，法線ベクトルである．

また 2 項目の輝度の誤差項では，現在の輝度画像 I_t と m 番目の 3 次元モデルと $t-1$ フレームに投影したときの輝度画像 $\hat{I}_{t-1}$ 間の輝度情報が最も揃うような位置姿勢パラメータ ψ_m を求めることを目的としており，

$$E_m^{pho} = \sum_{\boldsymbol{u}\in\Omega} (I_t(\boldsymbol{u}) - \hat{I}_{t-1}(\pi(\exp(\psi_m)\pi^{-1}(\boldsymbol{u}))))^2, \tag{3}$$

と表される．ここで，π は透視投影変換であり，π^{-1} は距離画像の画素 $\boldsymbol{u}$ を 3 次元点に再投影する変換である．この式 (1) の最適化には，Gauss-Newton 法を用いる．

また，この際のカメラの位置姿勢推定には，移動物体領域や非剛体物体領域も含めて最適化が行われているため，segmentation モジュールで移動物体を segmentation 後，その移動物体だと判断された画素を除外し，再度カメラにのみ位置姿勢の推定を行う．

3.2 Segmentation

Segmentation モジュールでは，個々の物体に画像を分割する instance segmentation モジュールと，そのモジュールで検出されなかった移動物体を画像中から分割する motion segmentation モジュールから成る．

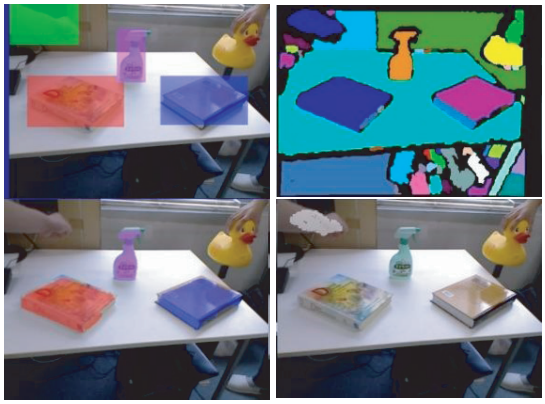

図 2　Instance segmentation の流れ（左上：Object detection，右上：幾何学的 segmentation，左下：instance segmentation 結果，右下：非剛体物体マスク結果）

3.2.1 Instance segmentation

Mask R-CNN [21] を用いて画像を物体ごとに分割していた既存の動的 SLAM とは異なり，本研究では，RGB 画像からの物体検出と距離画像からの幾何学的な segmentation を組み合わせることにより，instance segmentation を行う．

まず RGB 画像から YOLOv3 [22] を用いて物体検出を行う．物体検出を施すことで，学習されたデータセットに含まれる様々な物体の bounding box（矩形位置，大きさ）とその物体のカテゴリーを得ることができる（図 2 左上）．本研究では，MS COCO [29] により学習された重みを用いるため 81 カテゴリーの物体が検出される．しかし，物体検出のみでは，矩形領域内に背景領域が含まれており，物体のみの領域を抽出することはできない．

また，距離画像に立野らの幾何学的 segmentation [23] を行う．この幾何学的 segmentation では，距離画像の各画素において，法線情報と距離情報の非連続性を周囲の画素を用いて計算し，各画素が境界であるか否かを判定し 2 値画像を生成する．そして，その 2 値画像に対して連結成分を抽出することで，距離画像を面（凸面）領域に分割できる（図 2 右上）．

以上の物体検出，幾何学的 segmentation の結果を組みあわせ，物体ごとに物体カテゴリー情報とともに画像を分割する．検出された各物体の bounding box に対して幾何学的 segmentation の各面との Intersection-over-Union (IoU) を計算する．そして，その IoU がしきい値 (0.6) 以上の場合，その面に物体のラベルを割り当て，しきい値以下の場合，その面には背景ラベル

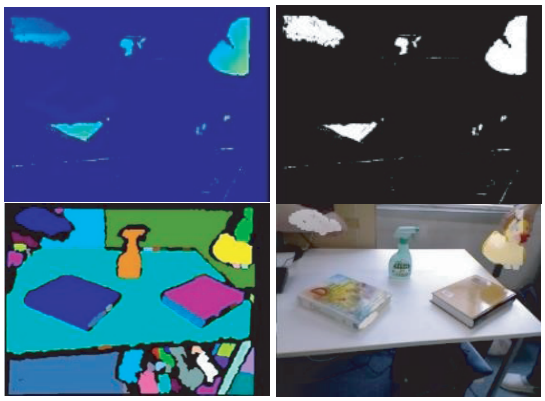

図 3　Motion segmentation の流れ（左上：誤差画像，右上：2 値化された誤差画像，左下：幾何学的 segmentation 結果，右下：移動物体のマスク結果）

を割り当てる．その結果，画像を物体ごとに分割することができる（図 2 左下）．

また，本研究では非剛体物体の 3 次元形状再構成をすることは目的としていないため，非剛体物体は 3 次元形状再構成せず，その物体ラベルが割り振られた画素はカメラの追跡から除去する．そのために，あらかじめカテゴリーごとに剛体物体か非剛体物体かを定めておき例えば person カテゴリーなどの物体領域は非剛体物体マスクとする（図 2 右下）．

3.2.2　Motion segmentation

Instance segmentation は既存手法では Mask R-CNN を用いて，本研究では YOLO の bounding box から行っていた．そのため例えばそれらの手法を施した結果，カテゴリーが学習されたデータセットに含まれていないような物体の場合や，またその手法の精度の問題から全ての移動物体を検出できない場合がある．その移動物体は画像から分割されず静的であるとみなされ，背景領域に統合されるため，カメラの位置姿勢推定に悪影響を及ぼしていた．

よって，instance segmentation とは別途移動物体領域を画像中から分割する motion segmentation を施す．そのために，tracking モジュールにおいてカメラの位置姿勢推定をした際の式 (2) を用いる．式 (2) では各画素ごとに前フレームとの 3 次元点の距離を計算しているが，もし物体が動いていた場合，推定されたカメラの位置姿勢を用いて 3 次元点を変換すると，前フレームの対応する 3 次元点の距離が大きくなってしまう（図 3 左上）．そして，その誤差画像に対して K-means ($K = 2$) を施すことで誤差画像を 2 値画像にし，動的な画素と静的な画素に画像を分割する（図 3 右上）．

しかし，二つの連続したフレーム間の物体の動きは小さいため，この 2 値化された画像は移動物体の一部分しかマスクすることができず，全ての移動物体の画素が含まれていない．この理由から，この 2 値化されたマスク画像と幾何学的 segmentation の結果を統合し最終的な移動物体の segmentation 画像を得る．ここで，幾何学的 segmentation により同じラベルが割り振られた画素は同じ物体に属すると仮定している．その統合には，instance segmentation と同じく IoU を用いた（図 3 右下）．

3.3　Mapping

Mapping モジュールでは，現在のフレーム情報を 3 次元地図に登録していく．本研究では，静的な環境地図，そして各検出された物体ごとに独立の 3 次元地図を作成する．3 次元地図の表現方法としては，ElasticFusion [4] と同様に Surfel を用いる．Surfel には，3 次元位置，法線，色情報，半径，重みの他にカテゴリーラベルが付与されている．現在のフレームの各画素を対応した 3 次元地図に登録する．その際，物体ごとに画像が分割されているため，まず背景画素は環境地図に登録し地図を更新する．各物体は，現在のフレームと 3 次元地図に登録されている物体のインスタンス ID の対応をとるために，各物体の 3 次元地図を現在のフレーム上に推定された位置姿勢を用いて投影し，segmentation された画像と重なり具合を計算することで，各物体 3 次元地図と現在フレームの個々の物体の対応付けを行う．また，重なり具合から対応する 3 次元地図が存在しない場合は，新しい物体であるとみなし，新しく環境地図を作成する．

4.　実　　験

本論文では，動的環境下で複数移動物体の追跡と 3 次元形状再構成，カメラの位置姿勢推定の精度を評価するために，様々な定性的，定量的実験を行った．**4.1** では，動的環境下でのカメラの追跡精度を評価し，**4.2** では，移動物体の 3 次元形状再構成精度を評価，そして **4.3** では，提案システムの実行時間を評価した．また，実験には，CPU: Intel Corei7-6950X 3.0GHz, GPU: GeForce GTX 1080Ti, RAM: 64GB, OS: Ubuntu 16.04 を用いた．そして，YOLOv3 の重みは，MS COCO dataset を用いて学習されたものを用いた(注1)．また，

(注1)：https://pjreddie.com/darknet/yolo/

表 2　絶対軌跡誤差による TUM RGBD Dataset を用いた動的環境下での既存研究とのカメラ位置姿勢推定精度比較

Setting	シーケンス	ATE RMSE (cm) ↓					
		VO-SF	ElasticFusion	StaticFusion	Co-Fusion	MaskFusion	提案手法
Slightly Dynamic	f3s_static	2.9	**0.9**	1.3	1.1	2.1	1.5
	f3s_xyz	11.1	**2.6**	4.0	2.7	3.1	5.2
	f3s_halfsphere	18.0	13.8	4.0	**3.6**	5.2	4.1
Highly Dynamic	f3w_static	32.7	6.2	**1.4**	55.1	3.5	2.8
	f3w_xyz	87.4	21.6	12.7	69.6	10.4	**8.5**
	f3w_halfsphere	73.9	20.9	39.1	80.3	10.6	**7.2**

入力する画像サイズはどの実験においても 640×480 とし，式 (1) における λ は 10.0 とした．本手法で必要なハイパーパラメータは全て Kinect v1 を用いて撮影された自作データセットで実験的に決定し，そのパラメータを全ての実験において用いた．

4.1　カメラ追跡誤差の定量的評価

まず，提案手法のカメラ追跡精度を評価し，関連研究と精度を比較することで本研究の優位性を示す．追跡精度評価のために SLAM の評価で広く用いられている TUM RGBD Dataset [24] を用いた．このデータセットには，Kinect v1 で撮影された六つの RGBD 画像シーケンスが含まれており，各シーケンスでは，人がテーブルに座っているほとんど動きのない動的環境（slightly dynamic 環境）と周りを歩いて回っている動きの大きい動的環境（highly dynamic 環境）を動いているカメラで撮影している．また，シーケンスは 3 種類のカメラ軌道 (static, xyz, spherical) がある．static ではカメラが微小に，xyz ではカメラが xyz 軸方向に，spherical ではカメラが半球を描くように動いている．

本論文では，幾つか既存の動的 SLAM，VO-SF [30], StaticFusion (SF) [6], Co-Fusion (CF) [12], MaskFusion (MF) [13]，そして静的 SLAM (ElasticFusion (EF) [4]) と比較する．本研究では，実時間で動作し，密に 3 次元再構成をする手法であるため，実時間では動作しない MID-Fusion [14]，EM-Fusion [15] や，疎に再構成をする DynaSLAM [7] とは比較しない．

カメラ追跡の評価指標には，絶対軌跡誤差 (Absolute Trajectory Error) [24] を用いた．絶対軌跡誤差はフレーム番号 1 から n までの真値のカメラ軌跡 $\mathbf{Q}_{1:n}$ と推定された軌跡 $\mathbf{P}_{1:n}$ の差から計算される．真値の軌跡と推定された軌跡は任意の座標系で定義されているため，誤差を計算する際にはその二つの軌跡を揃える必要がある．よって，Horn らの手法 [31] を用いて推定された軌跡 $\mathbf{P}_{1:n}$ から真値の軌跡 $\mathbf{Q}_{1:n}$ へ変換する剛体変換 $\mathbf{S}$ を求める．そして，各タイムステップ i において剛体変換の誤差行列を $\mathbf{F}_i = \mathbf{Q}_i^{-1}\mathbf{S}\mathbf{P}_i^{-1}$ 計算する．そして，絶対軌跡誤差は次のように算出される，

$$Error = (\frac{1}{n}\sum_{n=1}^{n} ||trans(\mathbf{F}_i)||^2)^{1/2}, \tag{4}$$

ここで $trans(\mathbf{F})$ は剛体変換行列 F の並進ベクトル成分を表す．

表 2 に精度評価をまとめる．値が小さいほど，精度よくカメラの位置姿勢を推定できたことを示す．表より，提案手法は既存の動的 SLAM と同等，若しくはより精度が高いことがわかる．特に，highly dynamic 環境のシーケンスである f3w_xyz と f3w_halfsphere では既存の動的 SLAM より精度よくカメラ追跡が行えた．その理由として，本手法では既存手法とは異なり，instance segmentation だけでなく motion segmentation も行ったことで，未検出の移動物体をカメラ追跡に含めなかったためだと考えられる．

図 4 に，instance segmentation と motion segmentation の結果を載せる．各シーケンスごとに載せたフレーム番号を括弧の中に示している．図では，カメラ追跡から除外された画素を白で表示している．上段に示している slightly dynamic 環境のシーケンスでは，人はほとんど動いていないため，motion segmentation ではマスクされていないが，人が物体として検出された場合，正しく人領域がマスクされていることがわかる．しかし，例えば f3s_xyz のシーケンスの右の人に見られるように，物体検出器は常に正しく物体の位置を予測できないため，物体が未検出の場合，マスクすることができない．一方，下段に示している highly dynamic 環境のシーケンスでは，人がシーン中を歩いているため，motion segmentation 結果において正しくマスクされていることがわかる．f3w_xyz シーケンスにおいて，instance segmentation では人は検出し除外されているものの，人が動かしている椅子は未検出だったことがわかる．一方，motion segmentation では，人と椅

図 4　各シーケンスにおける Segmentation 結果（左：instance segmentation，右：motion segmentation）

表 3　TUM RGBD Dataset を用いた動的環境下でのベースライン手法とのカメラ位置姿勢推定精度比較 (IS: Instance segmentation, MS: Motion segmentation)

Setting	シーケンス	ATE RMSE (cm) ↓			
		Baseline 1 (IS:✓, MS:✕)	Baseline 2 (IS:✕, MS:✓)	Baseline 3 (IS:✓, MS:△)	提案手法
Slightly Dynamic	f3s_static	1.5	**1.3**	1.6	1.5
	f3s_xyz	4.9	**4.0**	5.0	4.9
	f3s_halfsphere	**4.0**	**4.0**	4.4	**4.0**
Highly Dynamic	f3w_static	3.7	4.0	3.2	**2.8**
	f3w_xyz	10.0	10.6	9.9	**8.5**
	f3w_halfsphere	10.2	10.8	7.8	**7.2**

子ともに segmentation された．このように，instance segmentation だけでなく，motion segmentation も施すことでカメラ追跡から未検出な移動物体を除外しカメラ追跡の精度を向上させることができた．

次に，本手法のモジュールである，instance segmentation，motion segmentation がどのようにカメラ追跡精度を向上させたかを比較，検証する．そのために，表 3 に幾つかのベースライン手法との比較をまとめる．ベースライン 1 では，segmentation モジュールにおいて，instance segmentation のみを用いた手法，ベースライン 2 では，motion segmentation のみを用いた手法，ベースライン 3 では，instance segmentation，motion segmentation どちらも用いるが，motion segmentation において，geometric segmentation の結果を適用せず，ICP の残差のみを K-means で 2 値化したものである．

提案手法と Baseline 1 を比較することで，motion segmentation の効果を，提案手法と Baseline 2 を比較することで，instance segmentation の効果を，提案手法と Baseline 3 と比較することで，motion segmentation において，幾何学的 segmentation と組み合わせる効果を検証する．Slightly dynamic 環境の 3 シーケンスでは，人が動いていないため何も segmentation されず最も精度が高いという結果になった．しかし，highly dynamic 環境の 3 シーケンスでは instance segmentation, motion segmentation どちらも行い，かつ提案の幾何学的 segmentation と組み合わせる motion segmentation を施す手法が最も精度が良いことが検証された．

また，複数のシーケンスにおいて，最終的に 3 次元再構成されたモデルを図 5 に示す．図により，複数のシーケンスにおいてシーンの 3 次元形状が復元できていることがわかる．Highly dynamic 環境のシーケンス (f3w_static, f3w_xyz) では，机の周りを歩いている 2 人が instance/motion segmentation により，ほとんど 3 次元再構成されていないことがわかる．一方，slightly dynamaic 環境のシーケンス (f3s_static, f3s_xyz) では背景の 3 次元モデルに加えてシーン中に座っている人も再構成されている．これは，highly dynamic 環境のシーケンスでは，物体検出器が物体（人）を未検出だったとしても，motion segmentation により再構成から除外されているが，slightly dynamic 環境のシーケンスでは，物体検出器が物体を未検出だった場合，人が動いていないため背景として 3 次元再構成されてしまうた

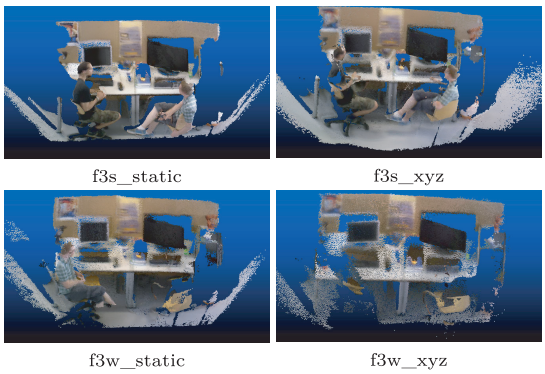

図 5　3 次元再構成されたモデル

表 4　移動物体の 3 次元形状再構成精度評価比較

評価指標	Co-Fusion	MaskFusion	提案手法
精度 ↑	0.792	**0.896**	0.810
再現率 ↑	0.525	0.457	**0.591**

めである．

4.2　3 次元形状再構成の定量的評価

次に，移動物体の 3 次元形状再構成精度を評価し，既存の Co-Fusion [12] と Mask-Fusion [13] と比較する．Co-Fusion で用いられた合成データセット (*room4-noise*) を用いた．そのデータセットでは，car カテゴリーの物体がシーン中を動いており，その物体に対応した CAD モデルが配布されているため，3 次元形状再構成の精度が評価できる．SLAM を用いて 3 次元形状再構成されたモデルの各点から CAD モデルの最近傍点までのユークリッド距離がしきい値以内の点の割合を精度，その逆を再現率とし，評価を行った．その結果を表 4 にまとめる．表 4 より，MaskFusion は精度よく再構成ができたものの，提案手法のほうが再現率が高かった．

また，実際に 3 次元形状再構成されたモデルを図 6 にまとめる．各点の誤差を色で表示しており，青色であれば誤差が小さく，赤いほど誤差が高いことを表している．図中の赤い枠で囲まれた領域に着目すると，提案手法により再構成されたモデルはより多くの点を再構成しており，表 4 に示しているように再現率が高くなっている．これは，本手法は物体検出により得られたおおまかな矩形と距離画像の geometric segmentation を組み合わせているため，他手法と比較し，より物体の境界を正しく segmentation できていることが原因の一つだと考えられる．一方，図中のオレンジ色の枠に着目してみると，提案手法により再構成されたモデル

表 5　ハイパーパラメータ λ の変化による 3 次元形状再構成精度評価比較

評価指標	6.0	7.0	8.0	9.0	10.0	11.0	12.0
精度 ↑	0.814	0.793	0.795	0.810	0.810	0.812	**0.823**
再現率 ↑	0.592	**0.595**	0.594	0.594	0.591	0.591	0.590

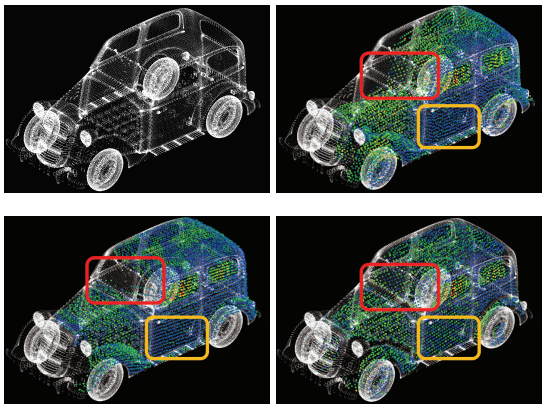

図 6　移動物体の 3 次元形状再構成結果（左上：真値の CAD モデル，右上：Co-Fusion，左下：MaskFusion，右下：提案手法）

は他手法と比較し，青色の点が少ない，つまり誤差が大きく精度が低いことがわかる．これは，MaskFusion で用いていた Mask R-CNN の精度が本手法で用いた YOLOv3 より高くシーケンス中のより多くのフレームで物体を検出でき，Mapping モジュールにおいて，より地図の更新が行われ，真値の 3D モデルに近いモデルが再構成できたためだと考えられる．

そして，3 次元再構成の精度は物体の位置姿勢推定の追跡精度に依るため，式 (1) におけるハイパーパラメータ λ を変化させて，再構成精度がどのように変化したかを検証する．この際，カメラの位置姿勢による影響をなくすために，カメラ位置性推定の際の λ は固定した．その結果を表 5 にまとめる．表より，精度が最も高かったときは，$\lambda = 12.0$ の場合で，再現率が最も高くなったときは $\lambda = 7.0$ のときであった．しかし，λ の変化による 3 次元再構成の精度の影響は低く，λ の変化に頑健な推定となっていることがわかる．

4.3　動作時間の定量的評価

最後に，提案手法の実行時間を計測し，実時間で動作することを検証する．提案手法は複数の要素から成るため，各要素の実行時間を表 6 にまとめる．ここで，Preprocess とは，カメラから得られた RGBD 画像に対し，以降の計算で必要な法線の推定，距離画像のバイラテラルフィルタを用いた平滑化などが含まれ

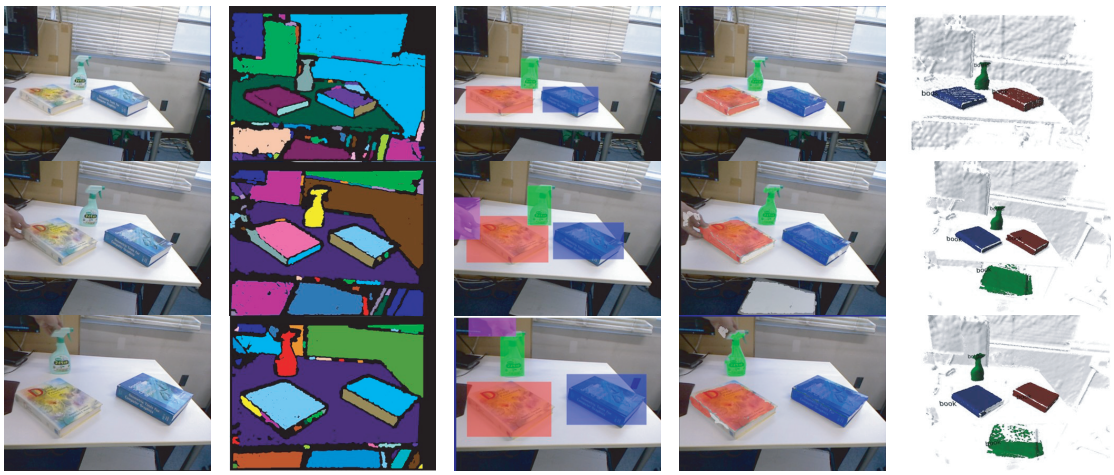

図 7　提案手法の定性的評価（左から，RGB 画像，Geometric segmentaion 結果，Object detection 結果，Object mask generation 結果，再構成された 3 次元モデル）

表 6　提案手法の実行時間．∗ がついた要素は並列に計算されている．

要素	実行時間 [ms]
Preprocess	5.40
位置姿勢推定	3.80 / model
幾何学的 segmentation ∗	6.16
物体検出 ∗	19.1
Motion Segmentation	2.72
Instance mask	0.57
カメラ姿勢改善	6.30
Mapping	2.05 / model
Total	46.1 + 5.16 / model

る．また，物体の位置姿勢や Mapping は個々の物体ごとに独立に行われるため，環境中に物体が多くなるほど計算時間が増える．そして，物体検出と幾何学的 segmentation はそれぞれの結果に依存しないため並列に行った．

表より，移動物体がない場合 46ms で，物体が 1 個存在すると 51ms で動作することが確認された．MaskFusion は平均して 33ms の実行時間を二つの Nvidia TITAN X GPU を用いて，また 12 フレームごとにのみ instance segmentation をすることで成し遂げていた．一方，提案手法では一つの Nvidia GTX 1080Ti を用いて，かつ instance segmentation を毎フレーム行いながら 46ms で動作するため，準リアルタイム (22FPS) でありながら，軽量な計算資源で動作していることがわかる．

4.4　提案システムの定性的評価

複数の物体が動いているシーンを撮影し，定性的に提案手法の有効性を評価する．シーン中には，三つの物体があり，人が物体を動かしている様子を動いている様子を Kinect で撮影している．図 7 にその結果をまとめる．図中の，幾何学的 segmentation の結果は各 segment ごとに色分けされて図示，Object detection の結果は物体の bounding box を色分けして表示してある．

図には，撮影したシーケンスの 244, 991, 1448 フレーム目を示している．991 フレーム目では，Book カテゴリーの物体が移動されており，1448 フレーム目では，Bottle カテゴリーの物体が移動されている．Object mask generation 結果により，正しく画像を物体ごとに分割できたことがわかる．また，その結果，移動物体の 3 次元形状も再構成され，かつ静的な背景も 3 次元再構成されていることが一番右の列の画像からわかる．

また，991 フレーム目，1448 フレーム目の再構成された 3 次元モデルの結果では，テーブル下の物体の 3 次元モデルが誤って生成されている．これは，244 フレーム目から 991 フレーム目の間に物体が Object detection モジュールで book カテゴリーの物体だと誤検出されてしまい，3 次元モデルが再構成されてしまっている．このように，物体検出器が物体を誤検出してしまうと背景にもかかわらず物体モデルだとして背景の 3 次元モデルとは別に作成されてしまい，その物体の画素はカメラの位置姿勢推定から除外され，カメラ追跡精度の低下を及ぼしてしまう．また，物体検出器が未検出を起こしてしまうと，motion segmentation によりカメラ位置姿勢推定には悪影響を及ぼさないものの，その物体の 3 次元再構成はされなくなってしまう．

このように，本手法は物体検出器の結果に大きく依存しているところが限界点として挙げられる．よって，今後の課題としてこのような物体検出器の誤検出や未検出により頑健な SLAM システムが求められる．

5. むすび

本論文では，動的な環境下で各移動物体の追跡と 3 次元形状再構成と SLAM（カメラの位置姿勢推定，静的な環境の 3 次元再構成）を行う手法を提案した．本研究の貢献点は，既存の動的 SLAM とは異なり，軽量なデバイスだけでも実時間で動作し，かつ正確にカメラ位置姿勢推定をすることである．

既存手法で計算時間のボトルネックとなっていた instance segmentation を，RGB 画像からの物体検出と距離画像からの幾何学的 segmentation に置き換えることにより速度を高速化した．また，既存手法では移動物体が未検出の場合カメラの位置姿勢推定に悪影響を及ぼしていたため，別途 motion segmentation を施すことでカメラ位置姿勢推定の精度向上を図った．

提案手法の有効性を確認するために実験を行い，本手法の動的環境下でのカメラの追跡精度，移動物体の 3 次元形状再構成精度，実行速度を評価した．その結果，カメラの追跡精度においては，既存手法と同等，シーケンスによっては既存手法より精度よく追跡が行えた．また，毎フレーム instance segmentation を施しながら，22fps で動作することを確認した．

謝辞　本研究は，JST CREST JPMJCR19F3 の支援を受けたものである．

文　献

[1] R. Mur-Artal and J.D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” IEEE Trans. Robotics, vol.33, no.5, pp.1255–1262, 2017.

[2] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol.40, no.3, pp.611–625, 2018.

[3] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, “KinectFusion: Real-time dense surface mapping and tracking,” IEEE Int. Symp. Mixed and Augmented Reality, pp.127–136, Oct. 2011.

[4] T. Whelan, R.F. Salas-Moreno, B. Glocker, A.J. Davison, and S. Leutenegger, “ElasticFusion: Real-time dense SLAM and light source estimation,” The International Journal of Robotics Research, vol.35, no.14, pp.1697–1716, 2016.

[5] T. Schops, T. Sattler, and M. Pollefeys, “Bad slam: Bundle adjusted direct rgb-d slam,” IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp.134–144, June 2019.

[6] R. Scona, M. Jaimez, Y.R. Petillot, M. Fallon, and D. Cremers, “StaticFusion: Background reconstruction for dense RGB-D SLAM in dynamic environments,” IEEE Int. Conf. Robotics and Automation, pp.1–9, May 2018.

[7] B. Bescos, J.M. Facil, J. Civera, and J. Neira, “DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes,” IEEE Robotics and Automation Letters, vol.3, no.4, pp.4076–4083, Oct. 2018.

[8] L. Xiao, J. Wang, X. Qiu, Z. Rong, and X. Zou, “Dynamic-slam: Semantic monocular visual localization and mapping based on deep learning in dynamic environment,” Robotics and Autonomous Systems, vol.117, pp.1–16, 2019.

[9] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S.R. Fanello, A. Kowdle, S.O. Escolano, C. Rhemann, D. Kim, J. Taylor, P. Kohli, V. Tankovich, and S. Izadi, “Fusion4D: Real-time Performance Capture of Challenging Scenes,” ACM Trans. Graphics, vol.35, no.4, pp.114:1–114:13, July 2016.

[10] M. Dou, P. Davidson, S.R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi, “Motion2Fusion: Real-time Volumetric Performance Capture,” ACM Trans. Graphics, vol.36, no.6, pp.246:1–246:16, Nov. 2017.

[11] R.A. Newcombe, D. Fox, and S.M. Seitz, “DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time,” IEEE Conf. Computer Vision and Pattern Recognition, pp.343–352, 2015.

[12] M. Rünz and L. Agapito, “Co-Fusion: Real-time Segmentation, Tracking and Fusion of Multiple Objects,” IEEE Int. Conf. Robotics and Automation, pp.4471–4478, May 2017.

[13] M. Rünz, M. Buffier, and L. Agapito, “MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects,” IEEE Int. Symp. Mixed and Augmented Reality, pp.10–20, Oct. 2018.

[14] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, “Mid-fusion: Octree-based object-level multi-instance dynamic slam,” 2019 Int. Conf. Robotics and Automation (ICRA), pp.5231–5237, 2019.

[15] M. Strecke and J. Stueckler, “Em-fusion: Dynamic object-level slam with probabilistic data association,” 2019 IEEE/CVF Int. Conf. Computer Vision (ICCV), pp.5864–5873, 2019.

[16] M.A. Alcorn, Q. Li, Z. Gong, C. Wang, L. Mai, W.-S. Ku, and A. Nguyen, “Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects,” IEEE Conf. Computer Vision and Pattern Recognition, pp.4840–4849, 2019.

[17] J. McCormac, A. Handa, A.J. Davison, and S. Leutenegger, “Semanticfusion: Dense 3d semantic mapping with convolutional neural networks,” IEEE Int. Conf. Robotics and Automation, pp.4628–4635, 2017.

[18] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, “Fusion++: Volumetric Object-Level SLAM,” Int. Conf. 3D Vision, pp.32–41, Sept. 2018.

[19] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, “Panopticfusion: Online volumetric semantic mapping at the level of stuff and things,” 2019 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), pp.4205–4212, 2019.

[20] A. Valada, R. Mohan, and W. Burgard, “Self-supervised model adaptation for multimodal semantic segmentation,” Int. J. Com-

puter Vision (IJCV), pp.1239–1285, July 2019.
[21] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” IEEE Int. Conf. Computer Vision, pp.2980–2988, Oct. 2017.
[22] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” CoRR, abs/1804.02767, 2018.
[23] K. Tateno, F. Tombari, and N. Navab, “Real-time and scalable incremental segmentation on dense SLAM,” IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp.4465–4472, Sept. 2015.
[24] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A Benchmark for the Evaluation of RGB-D SLAM Systems,” IEEE/RSJ Int. Conf. Intelligent Robot Systems, pp.573–580, Oct. 2012.
[25] 小澤岳大，中島由勝，斎藤英雄，“AR のための複数自由移動剛体の三次元再構成，” 情処学研報，vol.215, pp.1–3, Jan. 2019.
[26] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” Advances in Neural Information Processing Systems 24, eds. J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, pp.109–117, Curran Associates, Inc., 2011. http://papers.nips.cc/paper/4296-efficient-inference-in-fully-connected-crfs-with-gaussian-edge-potentials.pdf
[27] 紺野隆志，西田健次，糸山克寿，中臺一博，“視聴覚統合による動的環境下における 3 次元再構成の提案，” 人工知能学会 AI チャレンジ研究会，vol.55, pp.33–40, Nov. 2020.
[28] R. Hachiuma, C. Pirchheim, D. Schmalstieg, and H. Saito, “Detectfusion: Detecting and segmenting both known and unknown dynamic objects in real-time SLAM,” The British Machine Vision Conference (BMVC), pp.1–12, 2019.
[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, “Microsoft COCO: Common objects in context,” European Conf. Computer Vision, eds. D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, pp.740–755, 2014.
[30] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, “Fast odometry and scene flow from rgb-d cameras based on geometric clustering,” 2017 IEEE Int. Conf. Robotics and Automation (ICRA), pp.3992–3999, May 2017.
[31] B.K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” Josa A, vol.4, no.4, pp.629–642, 1987.

（2020 年 5 月 28 日受付，9 月 29 日再受付，
2021 年 1 月 6 日早期公開）

八馬　遼

2016 慶應義塾大学理工学部情報工学科卒．2020 年現在，慶應義塾大学理工学研究科後期博士課程に在学し，コンピュータビジョンとその応用に関する研究に従事．

Christian Pirchheim

2005 グラーツ工科大学 Dipl.-Ing (M.Sc.) 課程了．2015 グラーツ工科大学 Dr.techn. (Ph.D.) 課程了．現在，グラーツ工科大学の senior researcher と software engineer であり，Extended Reality for Industry 4.0 に所属．実時間コンピュータビジョン，ユーザインタフェース，Extended Reality のためのソフトウェア設計に従事．彼の研究は，幾つかの査読付き出版物や特許につながっています．特に ISMAR 2015 では Best paper を受賞．

斎藤　英雄　（正員）

1987 慶應義塾大学理工学部電気工学科卒．1992 同大学院理工学研究科電気工学専攻博士課程了．その後，同大学助手，専任講師，助教授を経て 2006 より教授．博士 (工学)．この間，1997 から 99 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．

Dieter Schmalstieg

ウィーン工科大学で博士課程了．現在では，オーストリアのグラーツ工科大学の教授．拡張・仮想現実感，コンピュータグラフィックス，ヒューマンコンピュータインタラクションの研究に従事．