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

再学習不要なRoll方向回転に頑健な単眼depth推定の精度改善手法*

齋藤　祐貴† 八馬　遼† 山口　真弘† 斎藤　英雄†

Training-Free Approach to Improve the Accuracy of Monocular Depth Estimation with In-Plane Rotation*

Yuki SAITO†, Ryo HACHIUMA†, Masahiro YAMAGUCHI†, and Hideo SAITO†

あらまし　近年盛んに研究されている畳込みニューラルネットワークを用いた単眼depth推定手法は，学習時にroll回転をする画像を用いて学習していないため，大幅なroll回転を含んだシーンに対するdepthの推定精度が低下する問題がある．本論文では単眼RGB-SLAMにより計算されるカメラ姿勢を利用した単眼depth推定度を改善する手法を提案する．本手法では，単眼RGB-SLAMにより推定されたカメラの姿勢を利用して，画像の縦軸とシーンの重力方向を一致させるようにRGB画像をroll回転させることにより，単眼depth推定アルゴリズムにより得られるdepth推定の精度向上実現するものである．本手法はニューラルネットワークを再学習させる必要がなく，既存の学習済みモデルを利用可能という利点がある．提案手法の有効性を検証するために2種類の異なるデータセットによる精度評価実験を行い，ニューラルネットワークを再学習させた手法と比較して，本手法が優れていることを定性的，定量的に確認した．

キーワード　単眼depth推定，SLAM，畳み込みニューラルネットワーク，拡張現実感

1. まえがき

RGB画像から物体までの距離 (depth) を推測する単眼depth推定はAugmented Reality (AR) [1] やロボットアプリケーション [2] 等に幅広く応用される技術である．近年，畳み込みニューラルネットワーク (CNN) を用いることによって大量のデータから入力となるRGB画像からのマッピングを学習させることで精度の高いdepth画像を推定可能になっている [3], [4].

CNNを用いた単眼depth推定の多くでは膨大な画像を用いてニューラルネットワークを学習させるが，学習画像の多くは被写体の重力場方向がおおよそ真下を向いている場合が多い．カメラのxyz各軸周りの回転をpitch, yaw, rollと呼び，一般に公開されているデータセット [5], [6] ではカメラがpitch回転やyaw回転をしているデータは含まれるが，カメラが大幅にroll回転するデータはほとんど含まれていない．そのためCNNを用いた単眼depth推定の従来研究では入力画像中に映る被写体の重力場方向がおおよそ真下を向いているという仮定が暗に存在する．例えば，床や道路などカメラからの距離が比較的近くなる物体は画像中の下部に映り，天井や空などカメラからの距離が比較的遠くなる物体は画像中の上部に映る傾向がある．こうしたデータを用いて学習させることにより，画像下部は値が小さく，画像上部は値が大きいdepthが推定される傾向が強くなり，depthを推定する際に入力画像中の重力場方向は強く推定精度に影響を与えると考えられる [7]．したがってカメラが大幅なroll回転をする入力画像をネットワークに入力すると，従来の単眼depth推定手法では学習時に大幅なroll回転を含んだデータを学習していないため推定精度が急激に低下する問題が起きる [8].

スマートフォンやドローンを用いたARアプリケーションではユーザが任意の角度にカメラを回転させることが可能であり，大幅なroll回転を含んだシーンが単眼depth推定の入力となる場合がある．すると，こうしたroll回転シーンに対するdepthの推定精度は被写体の重力場方向が画像座標系のy軸と平行であるときに比べ急激に低下し，アプリケーションへの支障が

† 慶應義塾大学理工学部情報工学科，横浜市
Department of Information and Computer Science, Keio University, 4–1–1 Hiyoshi, Kohoku-ku, Yokohama-shi, 223–8521 Japan
* 本論文は，システム開発論文である．
DOI:10.14923/transinfj.2020PDP0010

ground truth　　従来手法　　提案手法

図 1　CNN-MonoFusion [1] を用いた Saito らのデータセット [8] seq1 の 3 次元復元結果

考えられる．図 1 に roll 回転のみで構成されるビデオシーケンスを単眼 depth 推定の従来手法を用いて 3 次元復元した例を示す．従来手法は床などシーンの幾何構造が全体的に確認しにくい復元結果となり，平面検出等 AR への応用は難しいと考えられる．

そこで本論文では，単眼 RGB-SLAM で計算されるカメラ姿勢を取り入れた単眼 depth 推定の精度改善手法を提案する．RGB-SLAM は RGB 画像シーケンスのみを用いて，xyz の平行移動成分と xyz 軸の回転成分から構成される 6 自由度のカメラ姿勢を推定し，カメラの roll 方向回転角を抽出できる．提案手法ではこれを用いて入力画像にアフィン変換を適用し，カメラがあたかも roll 方向に回転していないような画像を生成する．変換後の画像を depth 推定ネットワークに入力し，その出力結果に再度，逆方向の回換をするアフィン変換を適用することで，入力画像と同解像度の depth 画像が推定可能となる．また本論文では，本手法の有効性を確認するために行った実験について述べ，本手法が CNN を再学習させたモデルの推定結果と比べ定性的，定量的にほぼ同精度の精度改善を行えることをことを示す．また本手法は従来手法と比べ，単眼 RGB-SLAM と単眼 depth 推定を用いたシーンの三次元復元結果で定性的な優位性を発揮することを示す．

2. 関 連 研 究

2.1　roll 方向回転を考慮した推定

カメラの roll 方向回転によって単眼 depth 推定の精度が低下する問題に着目して，Saito らは単眼 RGB-SLAM によるカメラ姿勢を考慮した精度改善手法を提案した [8]．カメラが roll 回転のみを行う独自の評価用データセットを作成し，画像を CNN に直接入力する従来手法と比べ定性的かつ定量的に推定結果に優位性があることを示した．単眼 depth 推定において従来研究の多くは roll 回転を含まない画像を入力として想定しており，学習時にカメラが roll 回転をするデータを含めたり，学習画像をランダムな角度で回転させる Data Augmentation を適用したりする手法は推定精度の低下を招くため避けられてきた [4], [7]．ネットワークをこうした手法で再学習させることで大幅な roll 回転を含んだ入力に対して推定精度の向上は期待はできる．しかし，CNN の再学習による大幅な roll 回転を含んだ入力に対する有用性を検証した研究例はない．また再学習には GPU を用いる膨大な計算コストや計算時間を要し，従来研究が公開している既存の学習済みモデルを直接利用することは困難である．本論文ではランダムな回転を加える augmentation を適用して CNN を再学習させる手法との比較を実験で行い，本手法の有用性を再学習の観点から検討した．

また，本研究と類似したアプローチを図った研究例として，Toyoda らは人間が側転など激しい運動をするビデオ映像における CNN を用いた骨格推定の精度改善を行った [9]．被写体が上下逆になるようなシーンは実世界では稀であり，骨格推定用のデータセットにこうしたデータは含まれない．また，CNN の再学習は学習時間や計算コストを要する欠点がある．そこで，ビデオ映像の各フレームを様々な角度で回転させ，最も信頼値の高い推定結果を採用することで人間の骨格推定の精度を改善した．

本研究はネットワークを再学習させることなく精度改善を図った点で Toyoda らの研究と類似しており，Toyoda らの骨格推定のタスクに対して，本研究では単眼深度推定に着目する．単眼深度推定は AR [1] やロボットアプリケーション [2] 等のリアルタイム性の高いアプリケーションに広く応用されるため，本研究では RGB-SLAM のリアルタイムで計算されるカメラ姿勢を直接用いたシステムを提案する．

2.2　単眼 depth 推定を組み合わせた RGB-SLAM

RGB-SLAM は特徴点ベースの手法 [10], [11] や輝度値ベースの手法 [12], [13] 等が提案されてきたが，これらの手法はテクスチャが十分なシーンにおいても回転運動をするカメラ姿勢を精度よく推定できない欠点がある．

そこで密な点群を復元し，テクスチャがあまりないシーンにおいてもカメラ姿勢を正しく推定するため，RGB-SLAM と単眼 depth 推定を組み合わせた手法が提案されている．代表例が CNN-SLAM [14] であり，CNN を用いて推定された depth が輝度ベースの RGB-SLAM に組み込まれている．CNN-SLAM は roll 方向回転が起こるシーンでも正確にカメラ姿勢

と密な環境地図を作成可能だが，roll 方向回転時には KeyFrame が生成されず depth を推定できない．そのため CNN に roll 方向回転を含むシーンを入力すると，**1.** で示したような精度低下が起こると予想される．また，CNN-MonoFusion [1] は TUM RGB-D データセット [6] の *rpy* シーケンスの 3 次元復元結果を評価しているが，CNN の推定 depth の精度を定量的に評価していない．本論文では，学習データに起因する単眼 depth 推定の欠点に問題を設定し，本手法の有効性を定性的，定量的に評価する．

2.3 CNN を用いた roll 方向回転角度の推定

1 枚の RGB 画像のみからカメラの roll 方向回転角を推定する手法が提案されている．例えば，Fischer らは画像の roll 方向回転角を回帰問題により推論するネットワークを提案した [15]．Greg らは 1 枚の RGB 画像から pitch，roll 方向の回転角を推定する CNN を用いた手法を提案した [16]．しかし，これらの手法はおおよそのカメラ回転角しか求まらず精度が不十分である．また幾何的拘束条件も考慮せず，学習データに基づいた推定しか行えない．また，Xian らは局所的，大域的な特徴を 1 枚の RGB 画像から抽出することで 2 自由度のカメラ回転角を推定している [17]．しかし，この手法は ±20° 若しくは ±50° のような小さな回転角でしか有効性を検証していない．

また，Jaderberg らは CNN を用いた画像分類のタスクにおいて予測に必要な部分だけを取り出し，対象物の姿勢を修正することで予測精度を向上させるネットワークを提案した [18]．ネットワークの任意の位置に姿勢修正を行うネットワークを組み込むことで，アフィン変換等の幾何変換パラメータを適切に学習し，より入力画像のひずみや回転にも頑健なモデルを構築可能にした．

従来の end-to-end な CNN の学習による roll 方向回転角の推定手法とは異なり，本手法では幾何情報を考慮した回転角の推定手法として ORB-SLAM2 [19] を用いる．ORB-SLAM2 は高精度にカメラ姿勢を計算でき，比較的大きな roll 方向回転角も精度よく推定できる．

3. 提案手法

3.1 システムの概要

図 2 に本手法の概要図を示す．まず，はじめに RGB-SLAM を用いて入力 RGB 画像のカメラ姿勢を推定する．次に SLAM のカメラ姿勢から KeyFrame 画像を

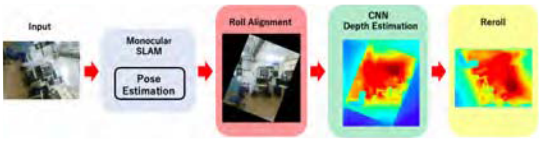


図 2　提案手法の概要図

回転させるアフィン変換 $F(\theta)$ を計算し，あたかも roll 回転していないような画像を生成する．そして，この画像を CNN に入力する．最後に出力の depth 画像に対して先ほどとは逆向きの回転を適用することで入力画像と同解像度の depth 画像を推定できる．

3.2 RGB-SLAM を用いたカメラ姿勢推定

単眼 RGB-SLAM は多視点幾何を用いて正確なカメラ姿勢を推定できる．カメラが回転運動をした際にも高精度なカメラ姿勢を推定するために，本手法は ORB-SLAM2 [19] を用いた．輝度ベースの RGB-SLAM [12], [13] に比べ，ORB-SLAM は計算コストが低い利点がある．

3.3 CNN を用いた単眼 depth 推定

depth 推定部には DenseDepth [4] と同じ構造のネットワークを用いた．このネットワークはエンコーダー部とデコーダー部から構成される．エンコーダー部には ImageNet [20] で事前学習された DenseNet-169 [21] が用いられ，入力画像は 1×94080 の特徴ベクトルへと変換される．デコーダー部には [25] のアップサンプリング層とエンコーダー部の同次元数の特徴を連結するスキップコネクションが用いられる．ネットワークの出力は入力画像の解像度の 1/2 なので，最近傍補間を用いて入力画像と同解像度に拡大する．また，学習時の内部パラメータと異なるカメラで撮影された入力画像の depth を絶対スケールで計算するために，ネットワークの出力 depth は以下の式 (1) を用いて SLAM のスケールに変換される．

$$D_{test} = \frac{f_{test}}{f_{tr}} D_{CNN} \tag{1}$$

ここで，D_{CNN} はネットワークの出力 depth，f_{test} は SLAM に用いられるカメラの焦点距離，f_{tr} は学習時に用いられるデータの焦点距離である．

3.4 回転角を用いた画像補正

depth 推定ネットワークに入力する前に RGB 画像にアフィン変換を適用し，被写体の重力場方向が画像座標系の y 軸と平行になるように画像を拡大，回転させる．この際 SLAM は初期フレームを基準としてカ

メラ姿勢を推定するため，今回 SLAM の初期フレームではカメラは一切 roll 方向回転をしていないと仮定する．

ORB-SLAM2 [19] を用いて推定されたカメラ姿勢 $\mathbf{T}_t^{cw} \in \mathbb{R}^{4\times4}$ は回転成分 $\mathbf{R}_t^{cw} \in \mathbb{R}^{3\times3}$ と平行移動成分 $\mathbf{s}_t^{cw} \in \mathbb{R}^3$ から構成される．この内，カメラ回転成分 $\mathbf{R}^{cw}$ は式 (2) のように xyz 3 軸周りの各成分へと分解できる．

$$\mathbf{R}^{cw} = \mathbf{R}^{cw}(\psi)\mathbf{R}^{cw}(\phi)\mathbf{R}^{cw}(\theta) \tag{2}$$

ここで，ψ は pitch，ϕ は yaw，そして θ は roll 方向の回転を表す．Roll 方向の回転に着目し，$\mathbf{R}^{cw}(\theta)$ は式 (3) のような 3×3 の行列で表現される．

$$\mathbf{R}^{cw}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

この θ を用いて，式 (4) に示す 2×3 の $F_t(\theta)$ アフィン変換を入力画像に適用する．

$$F_t(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & s_x \\ \sin\theta & \cos\theta & s_y \end{pmatrix} \tag{4}$$

s_x，s_y は元の RGB 画像の中心を変換後の画像の中心に平行移動させるベクトルである．

アフィン変換 $F_t(\theta)$ で画像を回転させる際にはバイリニア補間を適用し，元画像の pixel を完全に残し余白の pixel を最小限にするよう画像サイズを調整する．また，畳み込み演算の際に余白の pixel が影響を与えないようにするために，CNN の畳み込み層では余白の pixel 値を 0 に設定する．CNN の出力が得られたら，再度 depth 画像を逆方向に回転させるアフィン変換を適用し，元の RGB 画像と同解像度の depth 画像を得る．

4. 実　　験

4.1 評価用データセット

評価用データセットとして TUM RGB-D データセット [6] と Saito らのデータセット [8] の二つを用いた．TUM RGB-D データセット [6] は Kinect V1 を用いて屋内環境で撮影されており，この内カメラが pitch-yaw-roll の順で回転運動する 3rpy シーケンスを用いた．また，Saito らのデータセット [8] は Kinect V2 を用いて屋内環境で撮影されており，$-180 < \theta < 180$ のカメラの roll 運動のみで構成される 6 シーケンスを用いた．両データセットとも RGB 画像と真値の depth 画像の解像度は 640×480 である．

4.2 評 価 実 験

まず depth 推定ネットワークに **3.** の DenseDepth [4] を用いて定性的，定量的評価を行った．実験環境は Intel Core i7-7700 CPU と Nvidia GTX 1080Ti GPU 搭載のデスクトップ PC を用いた．本手法に対するベースライン手法として CNN に RGB 画像を直接入力して depth を推定する手法を設定した．

ネットワークの学習には NYU Depth V2 データセット [5] を用いた．このデータセットは異なる屋内空間で Kinect V1 を用いて撮影され，解像度 640×480 の RGB 画像と真値の depth 画像 120000 枚を含んでいる．この内 50688 枚を用いて，同じ絵柄に関して $-180 < \theta < 180$ の角度でランダムな roll 回転を加えた画像と加えていない画像の 2 セットを用意し，本手法とベースライン手法それぞれにおいて学習させた．roll 回転を加えた画像で学習させたモデルを Augmentation を付与したモデル，roll 回転を加えていない画像で学習させたモデルを Augmentation を付与していないモデルと設定し，両者を用いて本手法とベースライン手法の精度を比較した．

depth 画像中で値が欠けている画素は [22] の手法を用いて値を埋めた．ネットワークの出力 depth は入力画像の縦横半分の解像度なので，元画像と同じ解像度へと最近傍補間を用いて拡大した．重みの初期値として，エンコーダー部は ImageNet [20] を用いて事前学習させた DenseNet-169 [21] の重みを用い，デコーダー部は [23] の手法でランダムに初期化した重みを用いた．Optimizer には ADAM [24] を用い，学習率を 0.0001，$\beta_1 = 0.9$，$\beta_2 = 0.999$ に設定した．バッチサイズは 16，エポック数は 20 に設定し，Nvidia Quadro GV100 (32GB メモリ) を用いて学習を行った．

4.3 3 次元復元結果を用いた評価

本手法とベースライン手法の定性評価として CNN-MonoFusion [1] を用いて 3 次元復元結果を比較した．CNN-MonoFusion は ORB-SLAM2 [19] の RGB-D モードを用いてカメラ姿勢の推定を行い，これと単眼 depth 推定ネットワークの推定 depth を組み合わせることで密な 3 次元点群を復元できる．

今回，この depth 推定ネットワークの前後で画像にアフィン変換を適用する本手法と，画像を直接ネットワークに入力するベースライン手法との比較を行った．

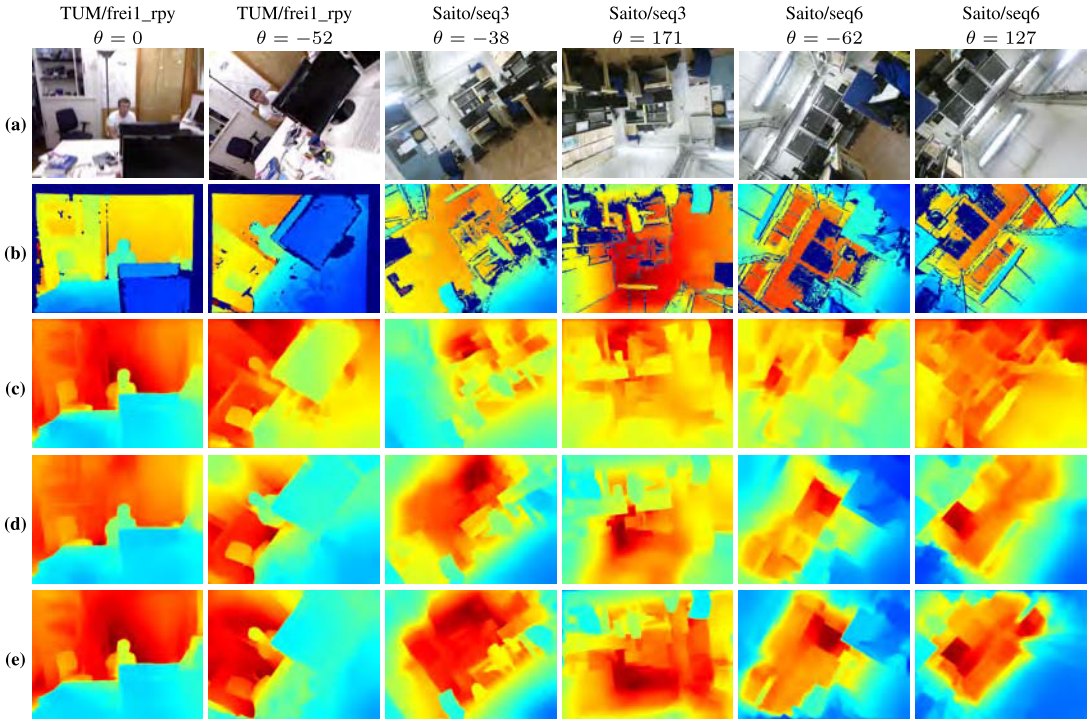


図 3　TUM RGB-D データセットと Saito らのデータセットにおける depth の推定結果に対する定性的評価．**(a)** 入力画像，**(b)** 真値，**(c)** 学習時に augmentation を加えていないベースライン手法，**(d)** 学習時に augmentation を加えたベースライン手法，**(e)** 本手法の結果．

depth 推定ネットワーク部とそのモデルには **4.2** で述べた DenseDepth [4] と一切 Augmentation を付与せずに NYU Depth V2 データセット [5] で学習させたモデルを用いた．ネットワークの出力 depth は入力画像の縦横半分の解像度なので，元画像と同じ解像度へと最近傍補間を用いて拡大した．

また，本手法とベースライン手法を用いた場合のシステムの処理時間を比較した．カメラ姿勢推定 (Tracking)，回転補正 (Roll Alignment)，Depth 推定 (Depth Estimation)，点群マッピング (Point Cloud Fusion) の各スレッドに要する処理時間を計測し，システム全体に要する処理時間を算出した．両手法ともにネットワークの重みは roll 回転による augmentation を加えていないモデルを用い，解像度を 480 × 270，カメラ速度を 10Hz，ORB 特徴点を 1 フレームあたり 2000 点抽出する設定で ORB-SLAM2 [19] を動作させた．

3 次元復元と処理時間計測の実験環境はともに Intel Core i7-7700 CPU と Nvidia GTX 1080Ti GPU 搭載のデスクトップ PC を用いて行った．

5. 考　　察

5.1 定性的評価

図 3 に TUM RGB-D データセットと Saito らのデータセットで実験した推定 depth の定性的結果を示す．1 行目から順に，入力画像，真値，学習時に augmentation を加えていないベースライン手法，学習時に augmentation を加えたベースライン手法，本手法の結果を示す．赤色 depth がより大きい depth 値を示し，青色がより小さい depth 値を示す．

$\theta = 0$ 付近では 3 手法とも真値に近い推定をしているが，θ が大きくなるにつれ学習時に augmentation を加えていないベースライン手法と比べて学習時に augmentation を加えたベースライン手法と学習時に augmentation を加えていない本手法はより真値に近い推定をしていることが分かる．また，この両者の推定結果は類似していることが分かる．

5.2 定量的評価

定量評価には絶対相対誤差 (Abs_Rel) と平均平方 2

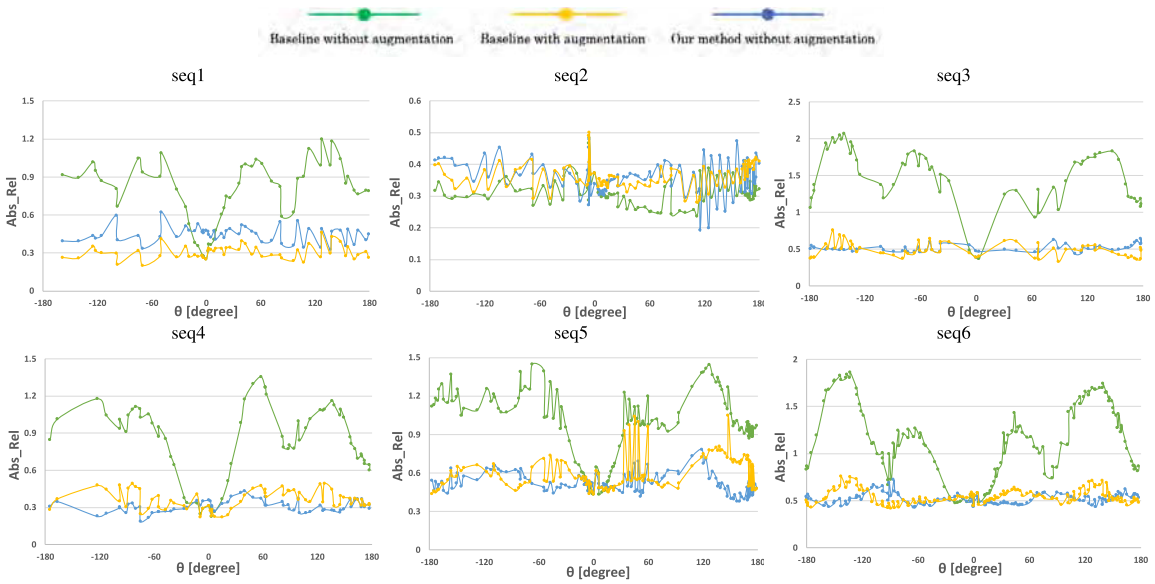


図 4　Saito らのデータセットにおける回転角度と絶対相対誤差 (Abs_Rel) の関係

表 1　Saito らのデータセットにおける平均誤差率

	Abs_Rel ↓			RMSE ↓		
	Baseline[1]	Baseline[2]	Ours[1]	Baseline[1]	Baseline[2]	Ours[1]
seq1	0.7656	0.4478	**0.3286**	0.9490	1.0913	**0.8927**
seq2	**0.3109**	0.4770	0.3622	0.8001	0.7588	**0.6097**
seq3	1.4721	0.5133	**0.4851**	2.4185	**1.5583**	1.6000
seq4	0.8344	**0.3071**	0.3612	1.4151	**0.8826**	0.9809
seq5	0.9770	**0.5185**	0.6012	1.6754	**1.2328**	1.3378
seq6	1.1214	**0.5176**	0.5406	2.2568	1.7223	**1.7027**
ave	0.9136	0.4635	**0.4465**	1.5858	1.2077	**1.1873**

[1] model without augmentation
[2] model with augmentation

表 2　TUM RGB-D データセットにおける平均誤差率

	Abs_Rel ↓			RMSE ↓		
	Baseline[1]	Baseline[2]	Ours[1]	Baseline[1]	Baseline[2]	Ours[1]
frei1_rpy	0.4190	0.4644	**0.4174**	**0.3973**	0.4444	0.4016
frei2_rpy	1.7474	**1.6323**	1.7033	3.5741	**3.5050**	3.5576
frei3_rpy	0.3121	0.3635	**0.3017**	0.8855	1.1413	**0.8614**
ave	0.8261	0.8201	**0.8075**	1.6190	1.6969	**1.6069**

[1] model without augmentation
[2] model with augmentation

乗誤差 (RMSE) の二つの指標を用いた．まず図 4 にシーケンスごとの結果 (Abs_Rel) を示す．横軸は roll 方向の回転角を示し，縦軸は誤差値を示す．全てのシーケンスにおいて $\theta = 0$ 付近の誤差値は 3 手法とも大差はない．しかし，$\theta < -30, 30 < \theta$ では augmentation を加えていないベースライン手法の誤差率は増加する一方，augmentation を加えたベースライン手法と augmentation を加えていない本手法は変化の少ない一定の誤差率を保持している．また表 1 に Saito らのデータセットの合計した平均誤差を示す．Roll 回転による augmentation を加えていないモデルを用いた本手法は augmentation を加えたモデルと加えていないモデルを用いたベースライン手法と比べより誤差率の低い結果が得られ，シーケンス全体では定性的な優位性を確認できる．

上記結果より，ランダムな roll 回転による augmentation を学習時に適用することで，大幅な roll 回転シーンを含んだ画像への推定精度を改善できるが，augmentation を学習時に適用しない場合にも本手法を用いることでほぼ同精度に推定精度を改善できると分かる．単眼 depth 推定の従来研究において，一般に公開されている既存の学習済みモデルは学習時に様々な回転角で roll 回転させる augmentation を適用していない．そのため本手法は augmentation を行う手法と比較して，大幅な roll 回転を含んだシーンに対して再学習のコストをかけることなく精度改善を行えるという点で優位性をもつといえる．

次に表 2 に TUM データセットの平均誤差を，図 5 にシーケンスごとの結果 (Abs_Rel) を示す．TUM データセットでは 3 手法とも誤差値に大差が見られないことが分かる．この原因の一つ目として TUM データセットの frei2_rpy，frei3_rpy では Kinect V1 の depth

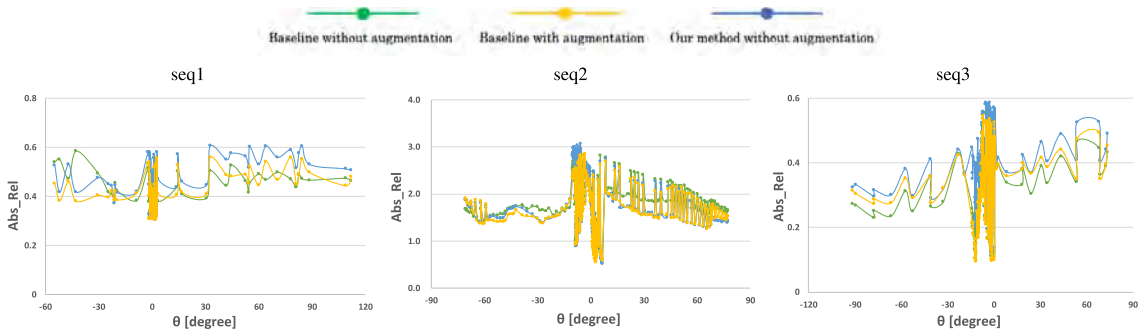


図 5　TUM RGB-D データセットにおける回転角度と絶対相対誤差 (Abs_Rel) の関係

表 3　TUM RGB-D/frei1_rpy における各モジュールの処理時間 (ms)

	Baseline			Ours			
	Tracking	Depth Estimation	Point Cloud Fusion	Tracking	Depth Estimation	Roll Alignment	Point Cloud Fusion
frames	677	677	677	677	677	677	677
min	16.76 ± 0.69	32.62 ± 0.36	1.03 ± 0.04	18.81 ± 1.16	32.21 ± 0.65	1.53 ± 0.05	14.57 ± 1.97
max	76.40 ± 1.69	47.41 ± 0.47	1165.09 ± 2.88	90.78 ± 10.54	1093.45 ± 26.89	6.30 ± 2.58	514.44 ± 45.76
median	38.21 ± 0.61	36.66 ± 0.36	37.69 ± 0.59	37.59 ± 1.24	36.99 ± 0.79	1.81 ± 0.04	36.87 ± 1.4
mean	37.96 ± 0.66	37.73 ± 0.33	49.49 ± 0.74	37.82 ± 1.71	43.03 ± 0.83	2.09 ± 0.13	42.49 ± 1.64
total time per frame	86.58 ± 0.68			93.3 ± 2.76			

図 6　Roll 回転を含まないシーンの三次元復元結果

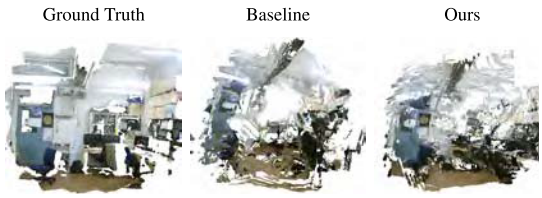


図 7　Saito らのデータセット seq3 の三次元復元結果

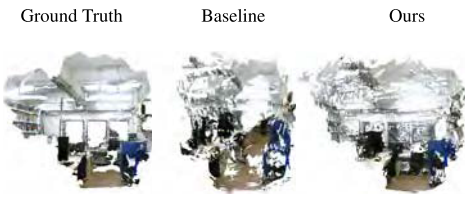


図 8　Saito らのデータセット seq6 の三次元復元結果

取得範囲を超えた depth をもつシーンがあり，depth の真値が 0 であるデータを多数含んでいるためである．これらの pixel 値は定量的評価で評価に含めないため，提案手法の有効性が確認できなかった事が挙げられる．また原因の二つ目として TUM データセットではカメラの roll 方向回転角が比較的大きい際，フレーム枚数が少ない傾向があるためである．フレーム枚数にばらつきがある中で平均値を比較したことで公平な評価ができなかった事も挙げられる．

5.3　3 次元復元結果による評価

まず図 6 に roll 回転を含まないシーンを 3 次元復元した結果を示す．ベースライン手法，本手法ともに床や壁など屋内環境の幾何構造を明瞭に確認できる復元ができていることが分かる．大幅な roll 回転を含まない入力に対して，アフィン変換を適用する本手法とベースライン手法では CNN に入力する画像に大きな変化がないため，共に ground truth に近い復元ができたと考えられる．

次に図 7，8 に roll 回転を含む Saito らのデータセット [8] を復元した結果を示す．ベースライン手法は大幅なカメラの roll 回転によって床の一部などにずれが生じているが，本手法ではシーン全体の構造をより明瞭に復元できていることが分かる．ベースライン手法と比べて 3 次元復元に用いる depth の推定精度が向上した事によって，復元結果に定性的な違いが生じたと考えられる．AR アプリケーションなどにおいてユーザが任意の角度にデバイスを操作させることによって，大幅な roll 回転が引き起こされる場合にも本手法はよ

り頑健なシステムを実現できると考えられる．

また表 3 に CNN-MonoFusion [1] を用いた三次元復元システムの各モジュールに要する処理時間を示す．

提案手法は回転補正処理と depth 推定の入力画像サイズがアフィン変換で拡大されることによって，ベースライン手法より処理時間が増加すると分かる．しかし，システム全体として 1 フレームあたりに要する時間はベースライン手法が 11.5Hz，提案手法が 10.7Hz と両手法ともに 10Hz 程度で動作し，これはカメラ速度 10Hz と比較してリアルタイム性が十分であると考えられる．また，提案システムでは入力される全フレームに対して単眼 depth 推定を適用するため，このネットワークの処理時間がボトルネックになる．近年は [26], [27] 等の 30Hz 以上で動作するよりリアルタイム性の高い単眼 depth 推定手法が提案されており，こうしたネットワークを適用することでシステム全体の処理速度も更に向上できると考える．

6. むすび

本研究では単眼 RGB-SLAM によって計算されるカメラ姿勢を用いて，roll 回転を含むシーンに対する単眼 depth 推定の精度改善をする手法を提案した．単眼 depth 推定では一般的に大幅な roll 回転を含んだシーンや roll 回転に関する data augmentation を適用したデータを用いて CNN を学習させない傾向がある．しかし，本手法はこうした roll 回転したデータを学習していないモデルを用いても，CNN の学習時に augmentation を加えて roll 回転を含んだデータを再学習させたモデルとほぼ定量的，定性的に同程度の精度改善を行えることを確認した．本手法を用いることによって CNN の再学習にかかる計算時間や計算コストを削減することが可能となり，学習済みのモデルを直接利用した精度改善が可能となる．更に，本手法は三次元復元等の AR アプリケーションにも容易に応用でき，従来手法と比べて大幅な roll 回転を含んだシーンでもより高精度なシステムを実現できることを確認した．

単眼 RGB-SLAM を用いる際の初期フレームが roll 回転していないという仮定を取り除く点は今後の課題となる．また本手法はアフィン回転を適用することで CNN の入力画像に余分な画素が含まれ，depth 推定の精度に影響を与える可能性がある．この部分を考慮した depth 推定のネットワーク構造や畳み込みの演算手法を提案することも今後の課題である．

文　献

[1] J. Wang, H. Liu, L. Cong, Z. Xiahou, and L. Wang, “CNN-MonoFusion: Online monocular dense reconstruction using learned depth from single view,” IEEE Int. Symp. Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), pp.57–62, IEEE, Munich, 2018.

[2] A. Marcu, D. Costea, V. Licaret, M. Pirvu, E. Slusanschi, and M. Leordeanu, SafeUAV: Learning to estimate depth and safe landing areas for UAVs from synthetic data, L. Leal-Taixé and S. Roth, eds., ECCV 2018 Workshops, LNCS, vol.11130, pp.43–58, Springer, Cham, 2019.

[3] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” Int. Conf. 3D Vision (3DV), pp.11–20, IEEE, 2016.

[4] A. Ibraheem and W. Peter, “High quality monocular depth estimation via transfer learning,” arXiv e-prints, abs/1812.11941, 2018.

[5] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, Indoor Segmentation and Support Inference from RGBD Images, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds., ECCV 2012, LNCS, vol.7576, pp.746–760, Springer, Heidelberg, 2012.

[6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of RGB-D SLAM systems,” IEEE Int. Conf. Intelligent Robot Systems, pp.573–580, IEEE, 2012.

[7] L. Mi, H. Wang, Y. Tian, and N. Shavit, “Training-free uncertainty estimation for neural networks,” arXiv preprint arXiv:1910.04858, 2019.

[8] S. Yuki, H. Ryo, Y. Masahiro, and S. Hideo, “In-plane rotation-aware monocular-depth estimation using SLAM,” Proc. International Workshop on Frontiers of Computer Vision, pp.305–317, 2020.

[9] K. Toyoda, M. Kono, and J. Rekimoto, “Post-data augmentation to improve deep pose estimation of extreme and wild motions,” arXiv preprint arXiv:1902.04250, 2019.

[10] G. Klein and D. Murray, “Parallel tracking and mapping for small AR workspaces,” Proc. IEEE and ACM Int. Symp. Mixed and Augmented Reality, pp.225–234, IEEE, 2007.

[11] R. Mur-Artal, J.M.M. Montiel, and J.D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system,” IEEE Trans. Robotics, vol.31, no.5, pp.1147–1163, 2015.

[12] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., ECCV 2014, Part II, LNCS, vol.8690, pp.834–849, Springer, Heidelberg, 2014.

[13] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol.40, no.3, pp.611–625, 2017.

[14] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.6243–6252, IEEE, 2017.

[15] P. Fischer, A. Dosovitskiy, and T. Brox, “Image orientation estimation with convolutional networks,” J. Gall, P. Gehler, and B. Leibe, eds., GCPR 2015, LNCS, vol.9358, pp.368–378, Springer, Cham, 2015.

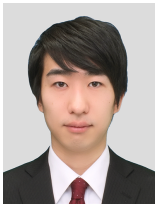
[16] G. Olmschenk, H. Tang, and Z. Zhu, “Pitch and roll camera ori-

entation from a single 2D image using convolutional neural networks," 2017 14th Conference on Computer and Robot Vision, pp.261–268, IEEE, 2015.
[17] W. Xian, Z. Li, M. Fisher, J. Eisenmann, E. Shechtman, and N. Snavely, "UprightNet: Geometry-aware camera orientation estimation from single images," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.9974–9983, IEEE, 2019.
[18] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," Advances in neural information processing systems, pp.2017–2025, 2015.
[19] 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, 2015.
[20] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.248–255, 2009.
[21] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger, "Densely connected convolutional networks," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.4700–4708, IEEE, 2017.
[22] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," ACM Trans. Graphics, vol.23, no.3, pp.689–694, 2004.
[23] X. Glorot and B. Yoshua, "Understanding the difficulty of training deep feedforward neural networks," Proc. Int. Conf. Artificial Intelligence and Statistics, pp.249–256, 2010.
[24] D. Kingma, and J. Ba, "Adam: A method for stochastic optimization," Proc. Int. Conf. Learning Representations, 2015.
[25] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," Proc. Int. Conf. Machine Learning, pp.2965–2974, 2018.
[26] A. Atapour-Abarghouei and T.P. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.2800–2810, IEEE, 2018.
[27] D. Wofk, F. Ma, T.J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," Int. Conf. Robotics and Automation, pp.6101–6108, IEEE, 2019.

（2020 年 5 月 26 日受付，9 月 13 日再受付，
2021 年 1 月 6 日早期公開）

齋藤　祐貴

2020 慶應義塾大学理工学部情報工学科卒．2020 現在，慶應義塾大学理工学研究科前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．



八馬　遼

2016 慶應義塾大学理工学部情報工学科卒．2020 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．

山口　真弘

2016 慶應義塾大学理工学部情報工学科卒．2020 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．

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 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．