NAIST-IS-MT1551079

修士論文

ディープニューラルネットワークを用いた カメラの相対運動推定

橋岡 佳輝

2017 年 3 月 16 日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

橋岡 佳輝

審査委員：

横矢 直和 教授	（主指導教員）
加藤 博一 教授	（副指導教員）
佐藤 智和 准教授	（副指導教員）
中島 悠太 客員准教授	（副指導教員 / 大阪大学）

ディープニューラルネットワークを用いた カメラの相対運動推定*

橋岡 佳輝

内容梗概

車の安全運転支援や自動運転などの様々なアプリケーションにおいて，複数枚の画像からカメラ運動を推定する手法が利用されている．多くの手法では画像上の特徴点の対応関係からカメラ運動とシーン構造を推定する．しかし，空や道路などのように画像上に特徴の少ないシーンでは，特徴点を検出できず推定に失敗することがある．また，人工的な建造物のように類似した構造が連続する場合も，特徴点の対応付けの誤りにより推定に失敗する可能性がある．このような理由から，より多様なシーンにおいてカメラの相対運動とシーンの 3 次元構造を頑健に推定可能な手法が求められている．

本研究では，ディープニューラルネットワークを用いて 2 枚の入力画像からカメラの相対運動推定する手法を提案する．ディープニューラルネットワークでは大量の画像データを用いることで，多様なシーンについて学習することができる．これにより，例えば海や道路のような従来のアプローチでは運動推定が難しい場面においても画像上の色の分布などからシーンに応じて大まかな相対運動の変化量が予測できると考えられる．学習にはカメラの相対運動の 3 次元空間での回転成分と並進成分を表す 6 自由度のパラメータを訓練データとして与える．層の深いネットワークを学習する際，モデルの初期値をランダムに与えると上手く学習が行われない場合がある．そこで，モデルにより良い初期値を与えるために，予め別のタスクについて事前学習をしたパラメータを利用する．本研究では事前学

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, NAIST-IS-MT1551079, 2017 年 3 月 16 日.

習としてデプスマップの推定を行う．2 枚の入力画像からデプスマップを出力するようモデルを学習し，作成したモデルをカメラの相対運動推定のために再学習する．加えて，本研究で想定されるアプリケーションの一つとして車載カメラの映像やドローンの空撮映像からのカメラ運動推定があげられる．これらの映像において連続するフレーム間の運動量には相関がある．このような時系列データのモデル化に Long Short-Term Memory(LSTM) を用いた手法が数多く提案されている．そこで，カメラの相対運動推定の精度向上のため，LSTM を用いた時系列データにおける過去の情報を考慮した推定手法を提案する．また，学習や評価の際には大量のデータが必要となるが，実シーンで計測されたデータには限りがある．そこで，コンピュータグラフィックスを利用したシミュレーションによる学習用のデータセットを作成する．

評価実験では，学習したネットワークによるカメラの相対運動推定の結果を示す．また，LSTM の有無による推定結果を比較することで推定の頑健性を検証し，提案手法の有効性を示す．

キーワード

カメラの相対運動推定, ディープニューラルネットワーク, シーン構造の推定, two-view structure from motion

Relative camera motion estimation using a deep neural network*

Yoshiki Hashioka

Abstract

Camera motion estimation from images is an essential technique for a wide range of applications such as driving support system and self-driving cars. Most existing methods estimate camera motion or 3D scene structure from keypoints correspondences between images. However, these keypoint-based methods do not work in scenes with fewer textures, such as sky or roads, due to the lack of keypoints correspondences. Scenes with repetitive textures also cause incorrect correspondences of keypoints, which results in estimation errors.

To address these problems, we propose a camera motion estimation method using deep neural networks (DNNs). Recent studies using DNNs enjoy large scale dataset, and they revealed the strong capability of DNNs to generalize to various data. We train our motion estimation model on a large dataset so that our method can handle challenging scenes including seas or roads. Our model is trained to predict 6-DOF camera motion, which represent a rotaion and a translation. Since it is hard to train a large DNNs, we trained our model on the task of depth estimation from images. After pretraining, we fine-tuned our model to predict relative camera motion from image pairs. Our possible applications includes motion estimation from videos captured with car-mounted cameras or UAVs. Since such videos rarely have drastic changes in motion, we

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1551079, March 16, 2017.

utilize Long Short Term Memory (LSTM) to model the time series data. Our model with LSTM incorporate previous motion for estimation on each time step, which improve robustness of prediction. For training our DNN, a large amount of image pairs with groundtruth motion parameters and depth maps are required, but such a dataset is not available. Therefore, we build a dataset for training our DNN by computer graphics simulation.

In our evaluation experiment, we show our DNN-based model can predict camera motions. Also, we verify the robustness of estimation by comparing the results by our model and baseline model without LSTM or not, and show the effectiveness of the proposed method.

Keywords:

ego-motion estimation, deep learning, scene estimation, two-view structure from motion

目 次

図 目 次

表 目 次

1. はじめに

車の安全運転支援や自動運転などの様々なアプリケーションにおいて，複数枚の画像からカメラ運動を推定する Visual-SLAM [1, 2, 3, 4] や Structure from Motion [5, 6, 7, 8, 9] と呼ばれる三次元復元手法が利用されている．これらの手法では撮影されたシーンの構造を推定すると同時に基準となる座標系におけるカメラの位置姿勢を推定する．この Visual-SLAM の初期化や Structure from Motion の処理の一部に 2 枚の画像間での相対運動の推定が行われており，重要な要素技術となっている．

カメラの相対運動を推定する多くの手法では，画像上から検出した特徴点を相対運動を推定する 2 枚の画像間で対応付け，対応付けられた特徴点の幾何学的な関係性についての拘束式を解くことでカメラの相対運動を推定する．しかし，画像にノイズが含まれる場合などでは，特徴点の対応付けの誤りなどにより正しい相対運動が推定できない場合がある．そうした問題に対し，外れ値の影響を低減する手法として RANSAC [10] や M-estimator を用いながら再投影誤差を最小化する手法が一般に使われる．再投影誤差とは，カメラの位置姿勢と特徴点の対応から推定した三次元点を画像上に投影し，投影された点と対応付けられた特徴点との距離の誤差である．再投影誤差を最小化することで，より正確な相対運動のパラメータを得ることができる．

RANSAC では対応付けられた特徴点の組の中からランダムに選択した一部の特徴点でカメラの相対運動を推定し，再投影誤差を計算する．これを繰り返し，再投影誤差が最も小さくなる推定結果を採用する．十分な回数ランダムに特徴点の組を選択することによりノイズの影響の少ない特徴点のみで推定することができ，外れ値の影響を低減できる．

M-estimator は，再投影誤差を最小化する際に外れ値の影響を小さくするよう重みづけすることで外れ値の影響を低減する．最小二乗法によって再投影誤差を最小化する際，誤差が大きくなる程最小化する関数への影響が大きくなるため外れ値の影響を強く受け全体の推定が失敗することがある．そこで，M-estimator では一定以上の誤差の影響が小さくなるような評価関数を利用することで，外れ値の影響を低減している．

(a) テクスチャが繰り返されるシーンの例

(b) 特徴の少ないシーンの例

図 1: カメラの相対運動の推定が困難であるシーンの例

しかし，図 1(a) のような同じ形状の窓が並ぶシーンなど，似た構造やテクスチャが繰り返されるシーンでは特徴点の配置が似るため，再投影誤差が小さくなり誤った組が対応付けられることがある．この問題に対し，Fredriksson らの手法 [11] では特徴点の対応付けと推定を最適化しながら繰り返すことで頑健な推定手法を提案している．通常の手法では特徴点同士を 1 対 1 で対応付けるが，Fredrikusson らの手法では検出された特徴点の各点に複数の対応関係を持たせる．その中から特徴点の対応の選択とカメラの相対運動の推定を繰り返し，再投影誤差を最小化することでより再投影誤差の小さい特徴点の対応関係を選択し，頑健な推定を実現する．しかし特徴点ベースの手法では，図 1(b) のように空や道路のような画像上に特徴の少ないシーンでは，特徴点を検出できず推定に失敗することがある．

一方，このような特徴点によらない相対運動推定手法として，ディープニューラルネットワークを用いた手法の開発が進められている．例えば、Handa ら [12] はディープニューラルネットワークを用いてコンピュータビジョン分野の幾何に関する諸問題を取り扱うためのライブラリを公開している．ここでは画像の幾何変換や透視投影，オプティカルフローや視差画像を計算する関数に加え，またディープニューラルネットワークの学習に必要な誤差逆伝播に使用される勾配を計算する関数が実装されている．また，Agrawal ら [13] はディープニューラルネットワークのシーン認識や物体認識のようなタスクの事前学習に，カメラの相対運動推定のタスクが有効であることを示した．従来，シーン認識や物体認識の

ようなタスクの学習には人の手でラベルづけされた大量の訓練データが必要となるが，人の手での訓練データの作成には膨大な時間とコストがかかる．一方で，カメラの相対運動は従来の位置姿勢推定手法を用いることで半自動的にデータを収集することができる．そのため Agrawal らはシーン認識や物体認識に必要な画像特徴量とカメラの相対運動推定に必要な視覚表現の一部は共通するというアイデアから，カメラの相対運動の推定がシーン認識や物体認識などの他のタスクにおけるディープニューラルネットワークの事前学習に有効であることを示した．

本研究ではディープニューラルネットワークを用いて 2 枚の画像からカメラの相対運動として 3 次元空間での回転成分と並進成分を表す 6 自由度のパラメータを推定する手法を提案する．ディープニューラルネットワークで構築されたモデルを大量の画像データを用いて学習することで，従来の特徴点に基づく手法では考慮されてこなかった画像上の特徴が得られることが期待できる．これにより，例えば海や道路のような特徴の乏しいシーンにおいても画像上の色の分布などから大まかな相対運動の変化量が予測できると考えられる．

本研究で使用するネットワークはパラメータ数が膨大であるため，学習が困難である．そこで本研究では事前学習としてデプスマップを出力するようモデルを学習し，その後カメラの相対運動推定のために再学習する．加えて，動画像を扱う際により精度を向上させるため，カメラの相対運動推定モデルに Long Short-Term Memory (LSTM) を取り入れる．これにより，時系列データにおける過去の情報を考慮したカメラの相対運動を推定する手法を提案する．

以下，2 章では関連研究及び本研究の位置づけについて述べる．3 章ではディープニューラルネットワークを用いたカメラの相対運動の推定方法について述べる．4 章ではシミュレーションデータを用いた学習と評価実験の詳細について述べ，結果を示す．5 章では本論文のまとめと今後の展望について述べる．

2. 従来研究および本研究の位置付けと方針

本章では従来から研究されているカメラの相対運動の推定手法について記述し，続いてディープニューラルネットワークを用いた関連手法について述べる．最後に，本研究の位置づけと方針についてまとめる．

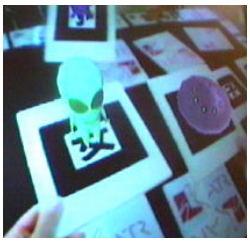
2.1 カメラ運動推定の従来手法

カメラ運動推定手法は，事前知識を用いる手法 [14, 15, 16, 17, 18] と事前知識を用いない手法 [1, 2, 3, 4, 5, 6, 7, 8, 9, 19, 20] に大別される．また，事前知識を用いない手法は複数枚の画像を対象にした手法と 2 枚の画像のみを対象とした手法に分けられる．このとき，事前知識を用いる手法ではマーカなどによって設定される座標系での位置姿勢，3 枚以上の画像を対象にした手法では多くの場合ある画像のカメラ座標系における位置姿勢を推定する．一方で本研究で対象とする 2 枚の画像のみを対象とした手法は 2 枚の画像間での一方の画像のカメラ座標系における他方のカメラの位置姿勢（相対運動）を推定する．以下，各手法の特徴について述べる．

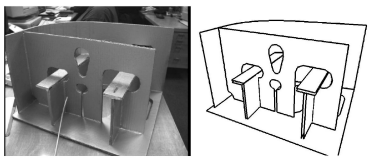
2.1.1 事前知識を用いる既存研究

既存研究として，マーカ [14, 15] ，CAD モデル [16] ，3 次元点群 [17, 18] などを事前知識として利用するカメラ運動の推定手法がある．マーカを用いる手法では，シーン内に画像上で検出が容易なマーカを配置し，マーカ上の 3 次元位置と画像上での二次元位置を対応付けることで，マーカに対するカメラの相対位置姿勢を推定する（図 2(a)）．この手法は拡張現実感技術のようにリアルタイム性が要求されるアプリケーションで広く用いられている．しかし，事前にマーカの配置が必要となるため利用可能な環境が限定されており，また拡張現実感技術のようなアプリケーションでは，画像上にマーカが写り込むために見栄えが悪くなるという問題がある．

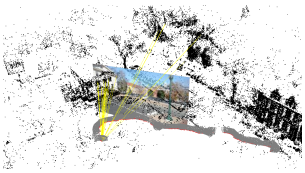
CAD モデルを用いる手法は，現実物体またはシーンの 3 次元 CAD モデルを事前に作成し画像上のエッジ等の特徴を CAD モデルに対応付けることで推定を行



(a) マーカを用いた手法



(b) CAD モデルを用いた手法



(c) 3 次元点群を用いた手法

図 2: 事前知識を用いる手法

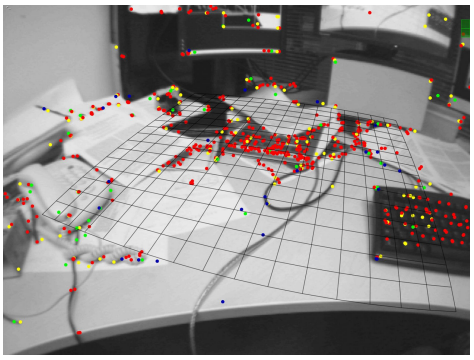
う（図 2(b)）．この手法ではマーカを用いることなくカメラ運動の推定が可能となるが，利用できるのは CAD モデルとして復元可能なシーンに限られ広域で複雑な屋外環境での推定には適さない．また CAD モデルの作成には専門的な知識や技能が必要となる．

3 次元点群を用いた手法は，Structure from Motion [5, 6, 7, 8, 9] などによって事前に 3 次元点群データベースを作成する．データベースに特徴点の見えについての記述子を記憶させておくことで，3 次元点群と画像上から検出した特徴点を対応付けカメラ運動を推定する．こちらもマーカを用いることなく高精度にカメラ運動を推定可能となる（図 2(c)）．しかし，画像上からの記述子抽出や高次元の記述子による特徴点の対応付けを行うために計算コストが高くなる．また，3 次元点群データベースの各点毎に記述子を保存するために膨大なデータ量が必要となるといった問題点がある．

2.1.2 事前知識を用いない手法

3 枚以上の画像を対象にした手法：事前知識を用いない推定手法の中でも 3 枚以上の画像からカメラの相対運動を推定する手法として Structure from Motion [5, 6, 7, 8, 9] や Visual-SLAM [1, 2, 3, 4] がある．Structure from Motion は複数枚の画像から撮影されたシーンの構造とカメラの位置姿勢を推定する手法である（図 3(a)）．推定には撮影された画像上の特徴点を検出し，検出した特徴点を画像間で対応付け，対応付けられた特徴点から得られる幾何学的な関係式から，カメラの相対運動とシーン構造を推定をする．その後，推定したカメラの相対運動と

(a) Structure from Motion の例



(b) Visual-SLAM の例

図 3: 複数枚の画像を対象にした手法

シーン構造を初期値として，推定に使用した全画像でカメラの位置姿勢を最適化することで高精度なカメラの相対運動とシーン構造を推定する．

特徴点の検出には SIFT [21] 検出器などが利用される．これは DoG(Difference of Gaussian) 画像を用いて特徴点を検出する手法である．DoG 画像は複数のスケールで作成した平滑化画像の差分をとることで画像の勾配を計算し，勾配の極値から特徴点となる点を検出する．

検出された特徴点の対応付けには，画像の局所領域の Sum of Squared Difference に基づく方法と SIFT 記述子のコサイン類似度を用いる方法などが利用される．画像の局所領域の Sum of Squared Difference に基づく方法では，特徴点周辺の局所領域で同士で画素の差の二乗和を計算し，二乗和が最小となる特徴点同士を対応付ける方法である．SIFT 記述子のコサイン類似度を用いる方法では，検出された特徴点の周辺領域で勾配のヒストグラムから最も勾配の大きい方向を検出し，エッジの方向を示す特徴ベクトルを計算する．検出した特徴点の中から特徴ベクトルのコサイン類似度が最も近いもの選ぶことで特徴点の対応付けをする．

次に，対応付けられた特徴点の組から，8 点アルゴリズム [19] や 5 点アルゴリズム [20] といった手法を用いることでカメラ運動を推定する．8 点アルゴリズムは，対応付けされた 8 点以上の特徴点から得られる関係式を用いてカメラの相対運動を表す 3×3 の行列を計算する手法である．5 点アルゴリズムは，カメラの内部パラメータが既知である場合に 5 点以上の対応関係からカメラの相対運動を

表す 6 自由度のパラメータを計算する手法である．

最後に，画像全体で再投影誤差を最小化するよう推定パラメータを最適化する．再投影誤差とは，推定した相対運動のパラメータを用いて画像上の特徴点をもう一方の画像に投影した点と，投影された点に対応付けられた特徴点との距離である．再投影誤差を最小化することで，より正確な相対運動のパラメータを得ることができる．Structure from Motion では，入力された全画像で再投影誤差を最小化するバンドル調整 [3] によって高精度な推定結果が得られる．

Visual-SLAM は動画像のような時系列画像を入力として，シーン構造とカメラ運動を推定す（図 3(b)）．Visual-SLAM では逐次的に入力される画像からカメラの相対運動とシーン構造を推定するため，オンラインでの動作が可能である．一方で，全ての画像の位置姿勢などをまとめて最適化することができず，誤差が蓄積するという問題がある．この問題を解消するため，既に撮影したシーンを再度撮影することで，シーン構造の推定結果を補正するループクロージング [4] とよばれる手法が提案されている．

2 枚の画像を対象にした手法：　2 枚の画像を対象にした手法は，カメラキャリブレーションや Visual-SLAM の初期値の設定，ベースライン距離の広い Structure from Motion の処理の一部として使用される．一般に，2 枚の画像でのカメラ運動推定は，画像上の特徴点の検出，検出された特徴点の対応付け，対応関係からのカメラの相対運動の推定という 3 つの処理が必要となる．これらの処理は Structure from Motion や Visual-SLAM と同様の処理によって行われる．

2 枚の画像からカメラの相対運動を推定する際，例えば画像にノイズが含まれる場合などに誤って特徴点を検出してしまい，推定に失敗することがある．こうした問題に対し，誤って検出された特徴点の外れ値の影響を低減させる手法として RANSAC [10] や M-estimator を用いて再投影誤差を最小化する手法が一般に利用される．

RANSAC では対応付けられた特徴点の組の中からランダムに選択した一部の特徴点でカメラの相対運動を推定し，再投影誤差を計算する．これを繰り返し，再投影誤差が最も小さくなる推定結果を採用する手法である．十分な回数ランダムに特徴点の組を選択することによりノイズの影響の少ない特徴点のみで推定す

ることができ，外れ値の影響を低減する．

M-estimator では，再投影誤差を最小化する際に外れ値の影響を小さくするよう重みづけすることで外れ値の影響を低減する．最小二乗法によって再投影誤差を最小化する際，誤差が大きくなる程最小化する関数への影響が大きくなるため，少数の外れ値に強い影響を受け全体の推定が失敗することがある．そこで，M-estimator では一定以上の誤差の影響が小さくなるような評価関数を利用することで，外れ値の影響を低減している．

しかし，同じ形状の窓が並ぶシーンなど，似た構造やテクスチャが繰り返されるシーンでは特徴点の記述子が類似するために誤った特徴点の組が対応付けられることがある．これに対し，Fredriksson らの手法 [11] では，特徴点の対応付けとカメラの相対運動の推定を繰り返し，これらを最適化することで推定に取り組んでいる．通常，2 枚の画像間で特徴点を対応付ける際はカメラ間の相対運動が未知の状態で対応付けをするため，画像上で検出される記述子の類似度のみを使用すると誤った対応付けされる場合がある．そこで，Fredriksson らの手法では検出した特徴点の対応付けをする際，各点が複数の対応関係を持つように対応付けをする．その後各特徴点で，複数の対応関係の中から 1 つの対応を選択し，カメラの相対運動の推定をする．この対応付けの選択とカメラの相対運動の推定を繰り返しながら再投影誤差を最小化することで，より頑健にカメラの相対運動を推定している（図 4）．しかし，これまでに述べたような特徴点ベースの手法では，例えば空や道路のような画像上に特徴の少ないシーンでは特徴点を検出できず推定に失敗することがある．

2.2 ディープニューラルネットワークを用いる既存研究

一方で、映像認識などの問題ではディープニューラルネットワークを用いることによる性能の向上が著しいことから [22] 、様々なタスクへの応用についても研究が進められている。本研究で対象とするカメラの相対運動推定においても，ディープニューラルネットワークによって大量の画像データを用いて学習することで，多様なシーンに合わせたカメラの相対運動推定に必要な画像上の特徴を学習することが期待できる．本研究では，ディープニューラルネットワークの学習

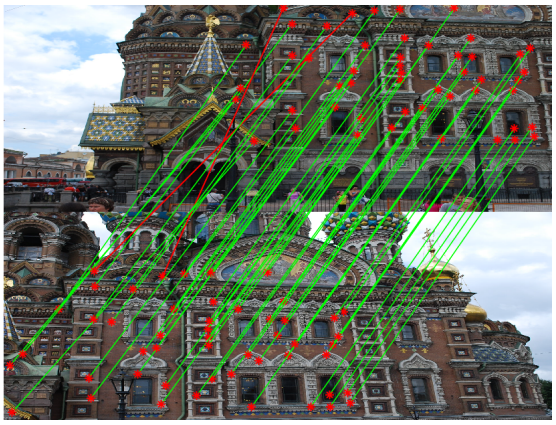


図 4: Fredriksson らの手法 [11]

に，デプス推定のタスクで事前学習したモデルを使用し，またカメラの相対運動推定の精度向上に LSTM を用いたネットワークを使用する．以下では，ディープニューラルネットワークを用いたカメラの相対運動推定の関連手法，デプス推定の関連手法，LSTM を使用した関連手法について記述する．

2.2.1 ディープニューラルネットワークを用いたカメラの相対運動推定の関連手法

Handa ら [12] はディープニューラルネットワークを用いてコンピュータビジョン分野の幾何の諸問題を取り扱うためのライブラリを公開している．ここでは画像の幾何変換や透視投影，オプティカルフローや視差画像を計算する関数に加え，ディープニューラルネットワークの学習に必要な誤差逆伝播に使用される勾配を計算する関数が実装されている．

Agrawal [13] らはディープニューラルネットワークのシーン認識や物体認識のようなタスクの事前学習に，カメラの相対運動推定のタスクが有効であることを示した．従来，シーン認識や物体認識のようなタスクでは訓練データとして人間の手によってラベルづけされた画像データが必要となるが，ディープニューラルネットワークの学習に使用するためのラベルづけデータを作成するには時間とコストがかかる．そのため，訓練データの準備が容易な別のタスクで事前に学習したネットワークを，再学習することで効率的にネットワークを学習することが可

能となる場合がある．Agrawal らはシーン認識や物体に必要な視覚表現とカメラの相対運動推定に必要な視覚表現の一部は共通するというアイデアから，カメラの相対運動の推定がディープニューラルネットワークの事前学習に有効であることを示した．Agrawal らの手法では，カメラの相対運動の推定を表す回転と並進のパラメータを変化量によってそれぞれ 10 のクラスに分け，2 枚の画像からパラメータの変化量にあたるクラスを推定するタスクによってネットワークを事前学習する．その後，事前学習したモデルを使用してシーン認識や物体認識のタスクに合わせ再学習し，事前学習をせずに学習したモデルとの比較し事前学習の有効性を検証している．

2.2.2 デプス推定の関連手法

ディープニューラルネットワークを用いたデプス推定手法に Liu ら [23] の手法がある．Liu らの手法では通常 2 枚以上の画像から推定するデプス画像を，ディープニューラルネットワークと Conditional Random Fields (CRF) を使用することで 1 枚の入力画像から推定する．本手法ではまず入力画像を領域分割し，分割された小領域の重心を中心として作成したパッチ画像をディープニューラルネットワークに入力する．入力したパッチ画像毎に推定したデプス値を CRF に入力することでパッチの周辺のデプス値を考慮したデプス画像を推定する（図 5）．

本研究と類似した入出力を持つ関連研究として，2 枚の入力画像からオプティカルフローを推定する Dosovitskiy [24] らの手法がある．Dosovitskiy らのネットワークは，畳み込み層によって画像の特徴を抽出する抽出パートと，逆畳み込み層による拡張パートによって構成される（図 6）．通常，畳み込みニューラルネットワークの学習では，畳み込みを繰り返すにつれて得られる特徴マップは画像の広範囲から得られる情報になり，画像上の局所的な細かい情報は失われ，畳み込みと逆畳み込みによって直接オプティカルフローを推定すると荒い推定結果になるという問題がある．Dosovitskiy らのネットワークでは逆畳み込みをする際に浅い層の畳み込み層で出力された特徴マップを再利用する（図 7）ことで，画像上の局所的な細かい情報を保存したままオプティカルフローを推定することができる．

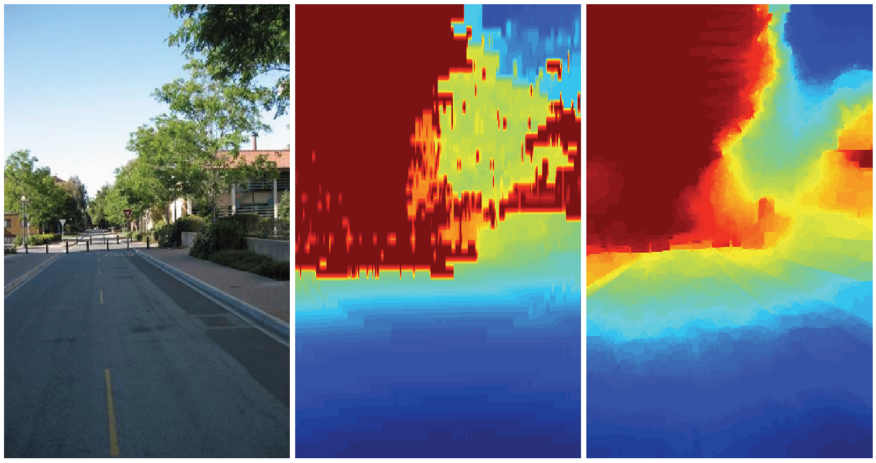


図 5: Liu らの手法 [23]

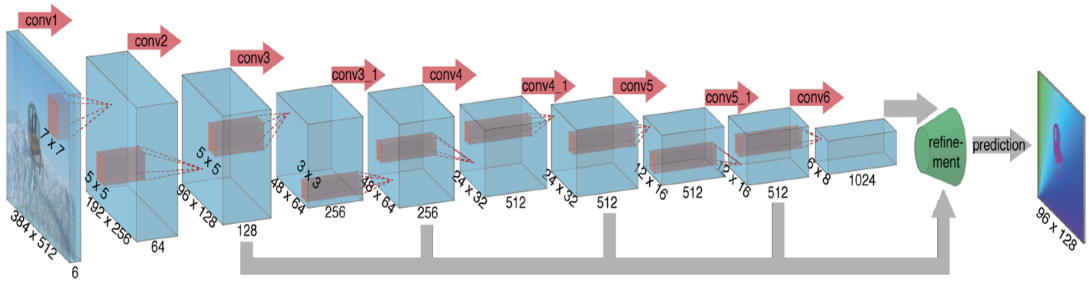


図 6: Dosovitskiy らの手法 [24] で提案されたネットワーク

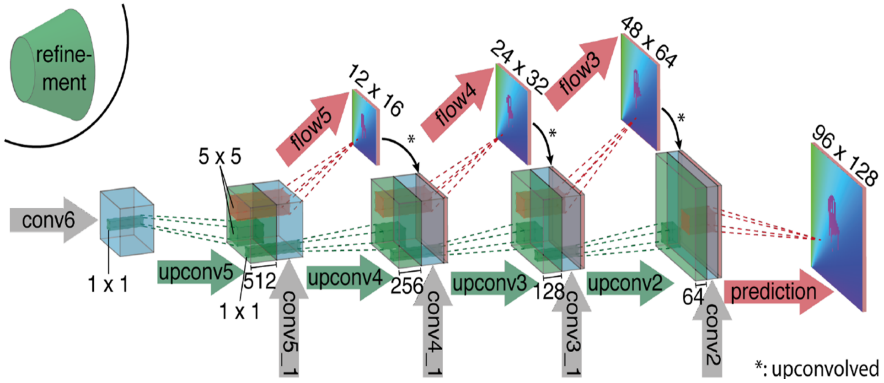


図 7: Dosovitskiy らの手法 [24] のネットワークの拡張パート

2.3 本研究の位置づけと方針

2.1 節，2.2 節で概観したように，カメラの相対運動を推定する手法が数多く提案されている．事前知識を用いた手法ではマーカや CAD モデル，三次元点群といった事前知識を利用することにより，高精度にカメラの相対運動を推定することができる一方で，予めシーン内にマーカを設置する，または事前に CAD モデルや三次元点群を作成する必要があるという問題がある．

事前知識を用いない手法は，3 枚以上の画像を対象にした手法と 2 枚の画像を対象にした手法に大別される．3 枚以上の画像を対象にした手法には Structure from Motion や Visua-SLAM のような手法がある．これらは画像上の特徴点の対応関係からカメラの相対運動とシーン構造を推定し，全ての画像で再投影誤差を最小化するように相対運動のパラメータを調整することで，高精度にカメラの相対運動を推定することができる．

本研究で対象とする 2 枚の画像を対象にした手法は，カメラキャリブレーションや Visual-SLAM の初期値の設定，ベースライン距離の広い Structure from Motion の処理の一部として使用される．2 枚の画像を対象にした手法も 3 枚以上の画像を対象にした手法と同様に画像上の特徴点を検出し，その対応関係からカメラの相対運動を推定する．2 枚の画像を対象にした手法の頑健性を高める手法として，RANSAC や M-estimator のような手法がある．しかし，似た構造が繰り返されるシーンにおいては，特徴点の記述子が類似するために誤った特徴点の組が対応付けられ，推定に失敗することがある．そのような問題に対し，Fredriksson らの手法 [11] では，特徴点の対応付けと推定を反復的に適用しながら最適化するより頑健な推定手法を提案している．しかし特徴点ベースの手法では，空や道路のような画像上に特徴の少ないシーンでは，特徴点を検出できず推定に失敗することがある．

一方で，映像認識などの問題ではディープニューラルネットワークを用いることによる性能の向上が著しいことから，他のタスクへの応用についても研究が進められている．Handa らはディープニューラルネットワークを用いてコンピュータビジョン分野の幾何に関する諸問題を取り扱うためのライブラリを公開している．また，Agrawal らはシーン認識や物体認識などのタスクの事前学習に，カメ

ラの相対運動の推定のタスクが有効であることを示した．しかし現段階で，2 枚の画像からのカメラの相対運動推定にディープニューラルネットワークを用いた手法は存在しない．

また，本研究ではカメラの相対運動を推定するネットワークの事前学習として，2 枚の画像からデプスマップを推定するタスクを使用する．ディープニューラルネットワークを用いたデプス推定の手法では，Liu らの手法が高い性能を示している．しかし，Liu らの手法では 1 枚の画像からデプスマップを推定するため，本研究の事前学習としては利用できない．そこで．類似した入出力を持つ関連研究として，2 枚の画像からオプティカルフローの推定をする Dosovitskiy らの手法がある．

以上を踏まえ，本研究ではディープニューラルネットワークを用いて 2 枚の画像からカメラの相対運動を推定するネットワークを提案する．ネットワークの事前学習には Dosovitsikiy らの手法を参考に，2 枚の画像からデプスマップを推定するタスクによって事前学習をする．これにより，ネットワーク内のパラメータにより良い初期値を与え推定精度を向上させる．また，本研究では LSTM を利用したネットワークによって時系列データの過去の情報を考慮することで，推定精度を向上させる．

3. ディープニューラルネットワークを用いたカメラの相対運動推定

本章では，提案手法について詳述する．まず初めに提案手法の概要について述べ，本研究で扱うカメラの相対運動の定義について詳しく説明する．次に，ネットワークの構成として提案手法で使用するデプス推定のネットワーク，カメラの相対運動推定のネットワーク，LSTM を用いたネットワークの構成について述べる．

3.1 提案手法の概要

本論文ではディープニューラルネットワークを用いて 2 枚の入力画像から 6 自由度のカメラの相対運動を表すパラメータ $p \in \mathbb{R}^6$（ただし、最初 3 つの要素は回転、残り 3 つの要素は並進を表す）を推定する手法を提案する．畳み込みニューラルネットワークを用いて 2 枚の画像のみからパラメータを推定する独立型ネットワーク、及び畳み込みニューラルネットに LSTM を導入することでこれまでの推定結果を考慮したパラメータを推定する時系列型ネットワークの 2 種類のネットワークを提案する。

学習の際は，まず 2 枚の画像からデプスマップを推定するタスクによって事前学習をする．これにより，ネットワーク内のパラメータをランダムに設定した際と比較して，より効果的にネットワークを再学習することができると考えられる．ここで事前学習したモデルは，独立型ネットワーク，時系列型ネットワークの双方のネットワークのパラメータの初期値として利用する．独立型ネットワークでは，事前学習したモデルに，カメラの相対運動を推定する畳み込み層を追加したネットワークを再学習する．同様に独立型ネットワークでは，畳み込み層と LSTM を追加する．LSTM を追加することにより，時系列データの過去の情報を考慮した推定が可能になる．そのため，畳み込み層のみによるネットワークと比較して，精度を向上させられると考えらえる．また，これらのネットワークの学習には，誤差関数として Mean Squared Error を使用した．

3.2 ネットワークの構成

提案手法では事前学習で用いるデプス推定ネットワークに加え，独立型ネットワーク，時系列型ネットワークの，3つのネットワークを学習する．3つのネットワークの概略図を図8示す．デプス推定ネットワークでは，オプティカルフローの推定で高い性能を示している Dosovitskiy らのネットワーク [24] を利用する．ネットワークの中でも，畳み込み層によって画像の特徴を抽出する抽出パートを F_{base}，逆畳み込み層による拡張パートを F_{depth} とする．独立型ネットワークでは，デプス推定ネットワークで事前学習した F_{base} のモデルを使用し，3層の畳み込み層 $F_{independent}$ によって，カメラの相対運動パラメータ p を推定する．時系列型ネットワークでは，独立型ネットワークと同様に事前学習した F_{base} のモデルに，3層の畳み込み層と2層の LSTM からなる F_{lstm} を追加することで，カメラの相対運動パラメータ p を推定する．以下では，デプス推定ネットワーク，独立型ネットワーク，時系列型ネットワークの，3つのネットワークの構成について詳述する．

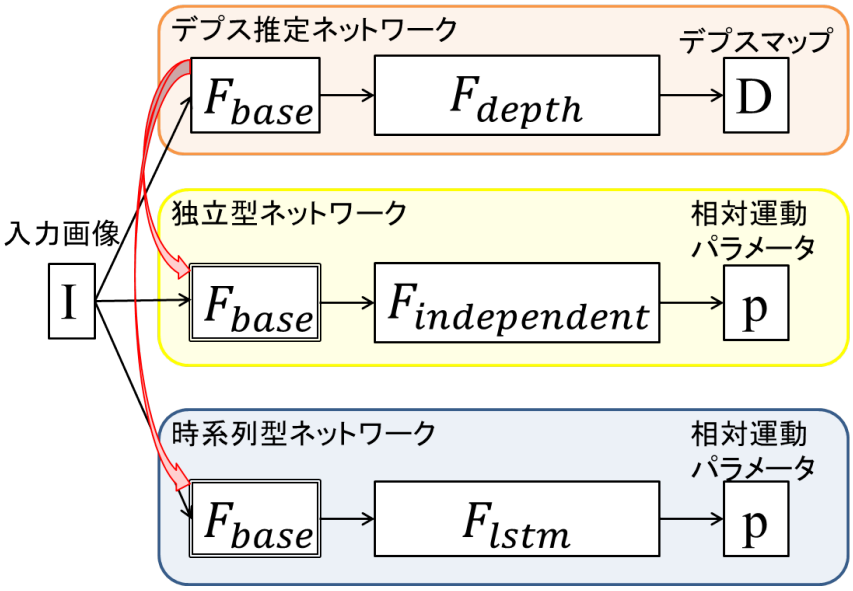


図 8: 3つのネットワークの概略図

3.2.1 デプス推定ネットワーク

ディープニューラルネットワークの学習では，ネットワークの規模が大きくなるにつれてネットワークの表現力が向上する一方，学習の収束に時間がかかる，過学習に陥りやすくなるといった問題がある．別のタスクでネットワークを事前学習し，対象とするタスクに合わせて再学習することでこの問題を解消できる場合がある．そこで，提案手法ではカメラの相対運動推定のための事前学習にデプス推定のタスクでネットワークを学習する．通常，2 枚の画像を用いて特徴点の対応関係からカメラの相対運動を推定する際，カメラの相対運動が推定されると同時に，使用した特徴点のデプス情報を推定することができる．そのため画像のデプス推定に必要な画像上の特徴はカメラの相対運動推定を行う際にも有用な情報であると考え，本研究ではカメラの相対運動推定の事前学習にデプス推定のタスクを使用した．

ディープニューラルネットワークを用いたデプス推定の従来研究に Liu らの手法 [23] などがある．Liu らの手法では高精度にデプスを推定しているが，推定には 1 枚の画像からデプス情報を推定しているため，本研究で対象とする 2 枚の画像からのカメラの相対運動推定の事前学習には適さない．デプス推定に関連する手法として，2 枚の画像からオプティカルフローや視差マップを高精度に推定している Dosovitskiy らの手法 [24] がある．オプティカルフローは 2 枚の画像間で各画素の移動量を 2 次元のベクトルで表したものであり，これにより 2 枚の画像間で画素同士の密な対応関係を得ることができる．通常，密なデプスマップを推定するためにはカメラの相対運動に加え画像間の対応関係が必要となるため，デプスマップの推定にはオプティカルフローの推定と同様の特徴が必要になると考えられる．提案手法では Dosovitskiy らのネットワークを使用し，事前学習としてデプスマップを推定するネットワークを学習する．

Dosovitskiy らのネットワークは，畳み込み層により 2 枚の画像から特徴を抽出する抽出パートと，逆畳み込み層による拡張パートによって構成される．抽出パートでは 9 層の畳み込み層によって畳み込みをしながら，大きさ 2 のストライドによってネットワークの出力のサイズを縮小する．これにより，画像全体の大まかな情報を抽出することができる一方，細かいエッジなど画像の局所的な情報

は失われてしまう．そこで，拡張パートでは抽出パートで計算された特徴マップを再利用することにより，推定するオプティカルフローの局所的な情報を表現する．また，拡張パートの各層で，縮小されたオプティカルフローを推定しながらアップサンプリングする．これにより，拡張パートの各層で真値と比較しながらアップサンプリングをすることができ，推定精度の向上が期待できる．

学習にはネットワークの推定値と真値との誤差を計算し，真値との誤差を小さくするようネットワークのパラメータを調整することで学習する．デプス推定ネットワークでは誤差関数として Mean Squared Error を使用する．ネットワークによるデプスマップの推定値を D_{pred}，真値を D_{gt} とすると，誤差は以下の式で計算できる．

$$Loss_d(D^{gt}, D^{pred}) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(D_{i,j}^{gt} - D_{i,j}^{pred})^2 \tag{1}$$

このとき，n，m はデプスマップのサイズを表す．

3.2.2 独立型ネットワーク

本ネットワークでは，デプス推定のネットワークの抽出パート F_{base} に加え，カメラの相対運動を推定する 3 層の畳み込み層 $F_{independent}$ を接続した構造のネットワークを使用する図 9．通常，畳み込みニューラルネットワークによるパラメータ推定には出力層として全結合層を使用することが多い．しかし，全結合層は畳み込み層に比べ使用するパラメータ数が多くなる．そのため，本ネットワークでは全ての層で畳み込み層を使用する．

また，デプス推定ネットワークと同様に誤差関数として Mean Squared Error を使用する．独立型ネットワークと時系列型ネットワークともに，カメラの相対運動推定での誤差関数の計算は以下のようになる．

$$Loss_p(p^{gt}, p^{pred}) = \frac{1}{6}\sum_{i=1}^{6}(p_i^{gt} - p_i^{pred})^2 \tag{2}$$

このとき，$p^{pred} \in \mathbb{R}^6$ をカメラの相対運動パラメータの推定値，$p^{gt} \in \mathbb{R}^6$ を真値とする．

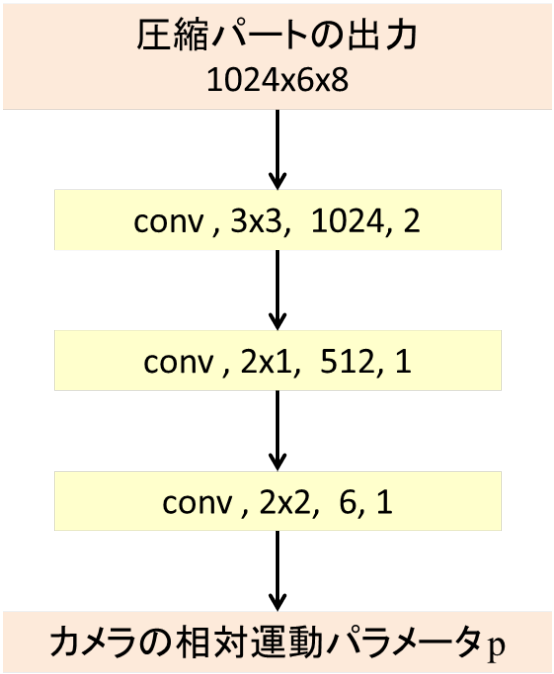


図 9: 独立型ネットワーク

3.2.3 時系列型ネットワーク

本研究では，車載動画や UAV で撮影されたような動画像において操舵の継続性や慣性などのために連続する画像間では相対運動は大きく変化しないというアイデアから LSTM を使用する．LSTM を用いることで動画像における過去の情報を考慮し，高精度にカメラの相対運動の推定が可能になると考えられる．

LSTM (Long Short Term Memory) [25] は RNN (Reccurent Neural Network) の一種で，時系列データの学習に用いられる．通常の RNN で時系列データを学習する際，RNN を時系列で展開することにより通常のニューラルネットワーク同様に誤差逆伝播法によって学習することができる．しかし，入力する時系列データの長さが長くなると，深い層では伝播する誤差が過剰に大きくなる，もしくは小さくなるため，時系列データの長期的な依存関係を学習することは難しいと知られている [26] ．LSTM では，セルと呼ばれる構造によりこの問題を解決している．セルでは時系列データのあるステップ t において，$t-1$ ステップ目で算出さ

れたセルに乗算と加算による線形の演算のみを行いセルの値を更新する．これにより，深い層においても勾配を計算する経路を確保し，長期的な依存関係を学習することができる．

LSTMのユニットの構造を図10に示す．LSTMは3つのゲートからなり，LSTMが現在の入力を使用するか選択する入力ゲート i_t，LSTMが過去のセルの情報を忘れるか選択する忘却ゲート f_t，現在どれだけ情報を出力するか選択する出力ゲート o_t がある．これらはそれぞれ入力 x が $(-\infty, \infty)$ の範囲で $(0, 1)$ の値をとる単調増加関数であるシグモイド関数

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{3}$$

の出力を乗算することで，セルに値を保持するか，廃棄するかを選択することができる．ある時間tにおける，LSTMへの入力を x_t，ネットワークの出力を h_t とすると，それぞれのゲートの出力は以下のように計算される．

$$\begin{aligned}
i_t &= \sigma(W_i[{x_t}^{\mathrm{T}}, h_{t-1}^{\mathrm{T}}]^{\mathrm{T}} + b_i) \\
f_t &= \sigma(W_f[{x_t}^{\mathrm{T}}, h_{t-1}^{\mathrm{T}}]^{\mathrm{T}} + b_f) \\
o_t &= \sigma(W_o[{x_t}^{\mathrm{T}}, h_{t-1}^{\mathrm{T}}]^{\mathrm{T}} + b_o) \\
c_t &= i_t \odot tanh(W_c[{x_t}^{\mathrm{T}}, h_{t-1}^{\mathrm{T}}]^{\mathrm{T}} + b_c) + f_t \odot c_{t-1} \\
h_t &= o_i \odot tanh(c_t)
\end{aligned} \tag{4}$$

このとき，Wとbはニューラルネットワークのパラメータであり，角括弧 $[\cdot, \cdot]$ はベクトルの連結，$\odot$ はベクトルの要素ごとの積を表す．

時系列型ネットワークでは，デプス推定のネットワークの抽出パート F_{base} に，畳み込み層とLSTMによるカメラの相対運動を推定する層 F_{lstm} を追加する．F_{lstm} の詳細を図11に示す．F_{lstm} は3層の畳み込み層と2層のLSTMによって構成される．ネットワークに入力する画像は，動画像のある時間 $t-1$ の画像 I_{t-1} と，時間 t から一定時間経過した時間 t の画像 I_t のペアを入力画像として与える．次に，画像 I_t と画像の I_{t+1}，画像 I_{t+1} と画像 I_{t+2} というように時系列順に画像を入力していく．これにより動画像の時間変化によるカメラの相対運動の変化を考慮して，カメラの相対運動を推定するようネットワークを学習する．ネットワークを学習

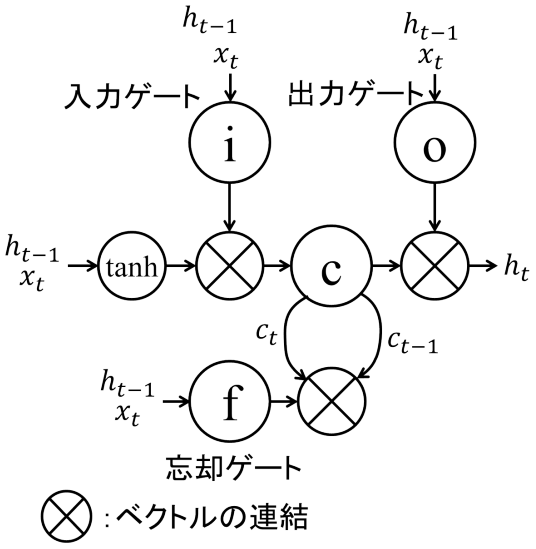


図 10: LSTM のユニットの構造

する際，効率的にネットワークを学習するテクニックとして Curriculum Learning [27] という手法を利用する．Curriculum Learning は，ネットワークを学習する際に，初めは目的のタスクと比較して簡単な問題設定によって学習し，後に難易度を上げて目的のタスクで学習することで学習の収束が早まり，より効果的に学習することができる．提案手法では 2 層目の LSTM に入力されるカメラの相対運動パラメータを，初めは時間 $t-1$ における真値を入力に用いて学習する．その後は，2 層目の LSTM の入力を 1 ステップ前の推定値に変更し，再度学習をする．

また，入力する動画像の系列が長くなると，深い層での誤差が過剰に大きく，もしくは小さくなるという問題がある．そこで，時系列データの誤差伝播を一定のステップで止める Truncated Backpropagation [28] という手法を利用し学習する．

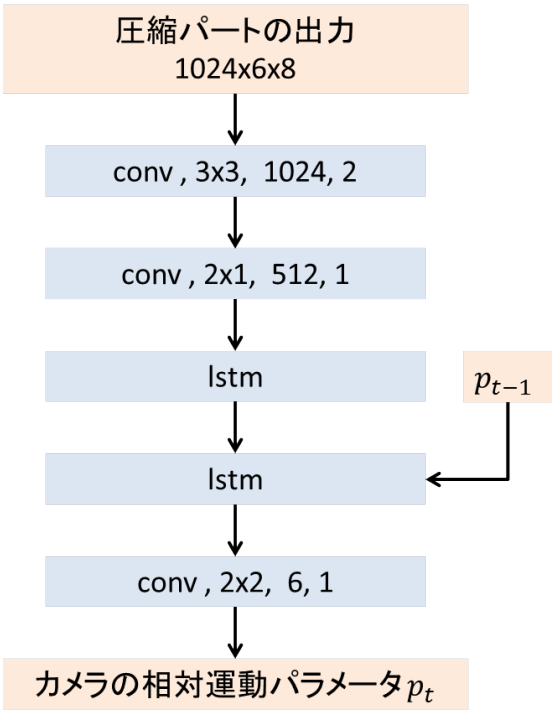


図 11: 時系列型ネットワーク

4. 実験と考察

4.1 実験概要

提案手法の有効性を検証するため，CG 画像によるシミュレーションデータを用いてネットワークを学習する実験を行った．ネットワークの学習には約 25 万ペアの画像を作成し，デプス推定のネットワーク，独立型ネットワーク，時系列型ネットワークをそれぞれ学習した．また，ネットワークの出力の精度を確認し，LSTM を追加することで，カメラの相対運動推定の精度が向上することを確認した．本章ではまず実験の概要を述べ，データセットの作成方法，評価方法について説明し，実験結果を示す．

4.2 使用するデータセット

本研究で学習に使用するデータとして，同じシーンを撮影した RGB 画像のペア，デプスマップ，画像間のカメラの相対運動の情報を含むデータセットが必要となる．これらの情報を含むデータセットとして KITTI データセット [29] などがある．KITTI データセットは車で移動しながら撮影されたデータセットで，車に取り付けられたカメラやレーザーレンジファインダ，GPS センサなどによって取得された，RGB 画像やデプスマップ，カメラの位置姿勢情報などを利用することができる．KITTI データセットでは，11 種類のシーケンス動画から，RGB 画像，デプスマップ，カメラの相対運動の情報を持つ 23,201 セットのデータが使用できる．しかし，ディープニューラルネットワークの学習には通常，多数の訓練データが必要となる．また，KITTI データセットでは車に取り付けられたカメラの位置姿勢を計測しているため，移動方向が水平方向に限られる．

ディープニューラルネットワークの学習のために Dosovitskiy ら [24] は，画像上に CG モデルを重畳表示した単純な合成画像によりオプティカルフローの学習に成功している．また，Mayer ら [30] は CG により作成したデータセットによって視差画像の学習に成功している．このことから，デプス推定やカメラの相対運動推定のようなタスクにおいても，シミュレーションにより幾何学的な整合性が

考慮されたCG画像を作成することでディープニューラルネットワークを学習することができると考えられる．

本研究では，Unreal Engine 4 (UE4) というゲームエンジンを使用し，学習用のデータセットを作成する．UE4はオープンソースのゲームエンジンで，本研究で使用するRGB画像，デプスマップ，カメラの相対運動の学習データを取得することができる．UE4によって得られる学習データの例を図12に示す．

UE4で出力されるデプスマップは$[0, 255]$の範囲で正規化されたデプス画像として出力される．また，UE4内のワールド座標における絶対位置姿勢を取得することができるため，そこから相対運動を計算することができる．また，Muellerら [31] によってUE4を用いたUAVのシミュレータが提供されており，本研究ではMuellerらのシミュレータを用いてデータセットを作成する．

本研究では，UE4を用いてランダムに位置姿勢を決定するランダムデータセットと，人間の手による操作でUAVを移動させながら連続で撮影したシーケンスデータセットの2種類のデータセットを作成した．ランダムデータセットではランダムに設定した位置姿勢で，学習用のデータを取得する．あるシーン内でランダムに設定された撮影位置からRGB・デプス画像を取得した後，カメラの位置姿勢をわずかに変化させデータを再取得する．この操作を繰り返すことで，RGB・デプス画像のペアとそのカメラの位置姿勢の移動量を収集する．本研究では298,888組の画像ペアとその相対運動を作成した．その内248,888ペアを訓練用，50,000ペアをテスト用として使用した．作成したデータセットの一部を図12に示す．

シーケンスデータセットでは，人間の手による操作でUAVを移動させながら，学習データを連続で取得する．これにより，時系列型ネットワークで学習する時系列データを作成する．学習には18のシーケンスを作成し，合計30609フレームの画像を作成した．その内16のシーケンスを訓練用に，2つのシーケンスをテスト用に使用した．画像のベースライン距離が短い場合，画像間の見えの変化が小さくなりカメラの相対運動が推定できない場合がある．そのため，提案手法では作成したシーケンス動画から20フレーム毎に切り出した画像を使用する．作成したデータセットの一部を図13に示す．

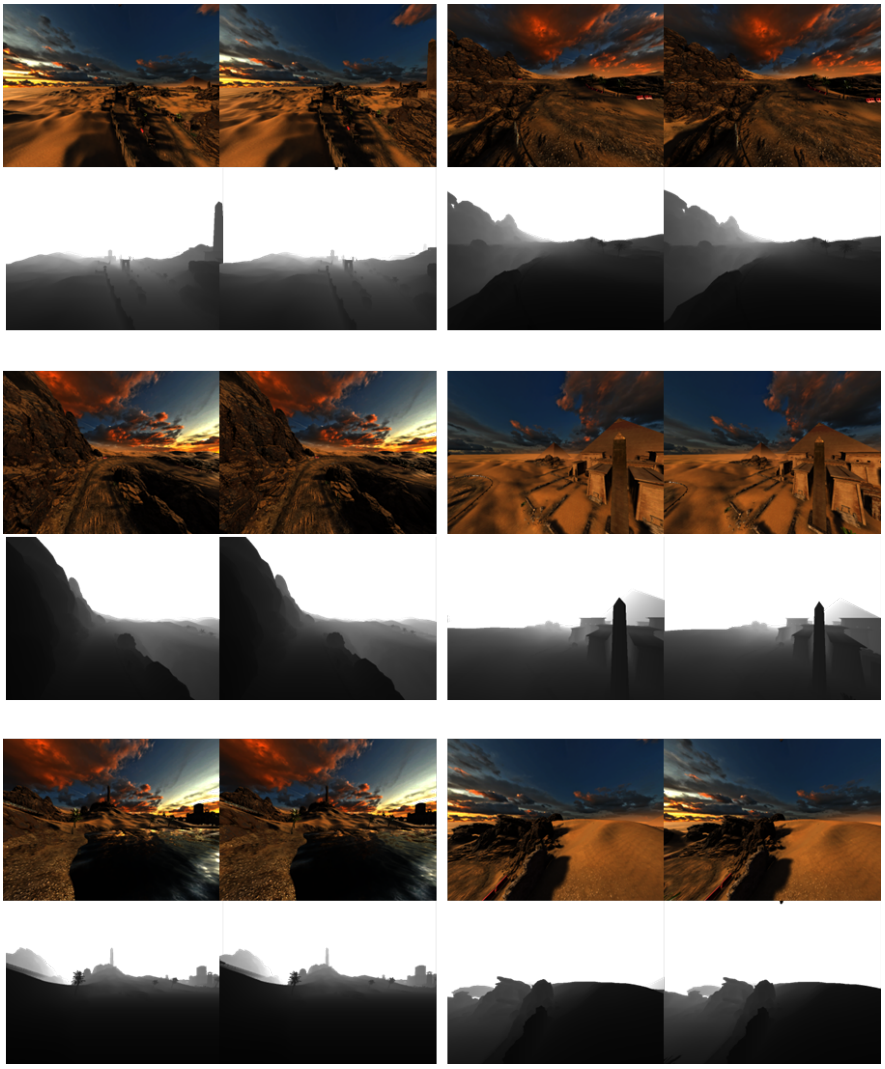


図 12: ランダムデータセットの例

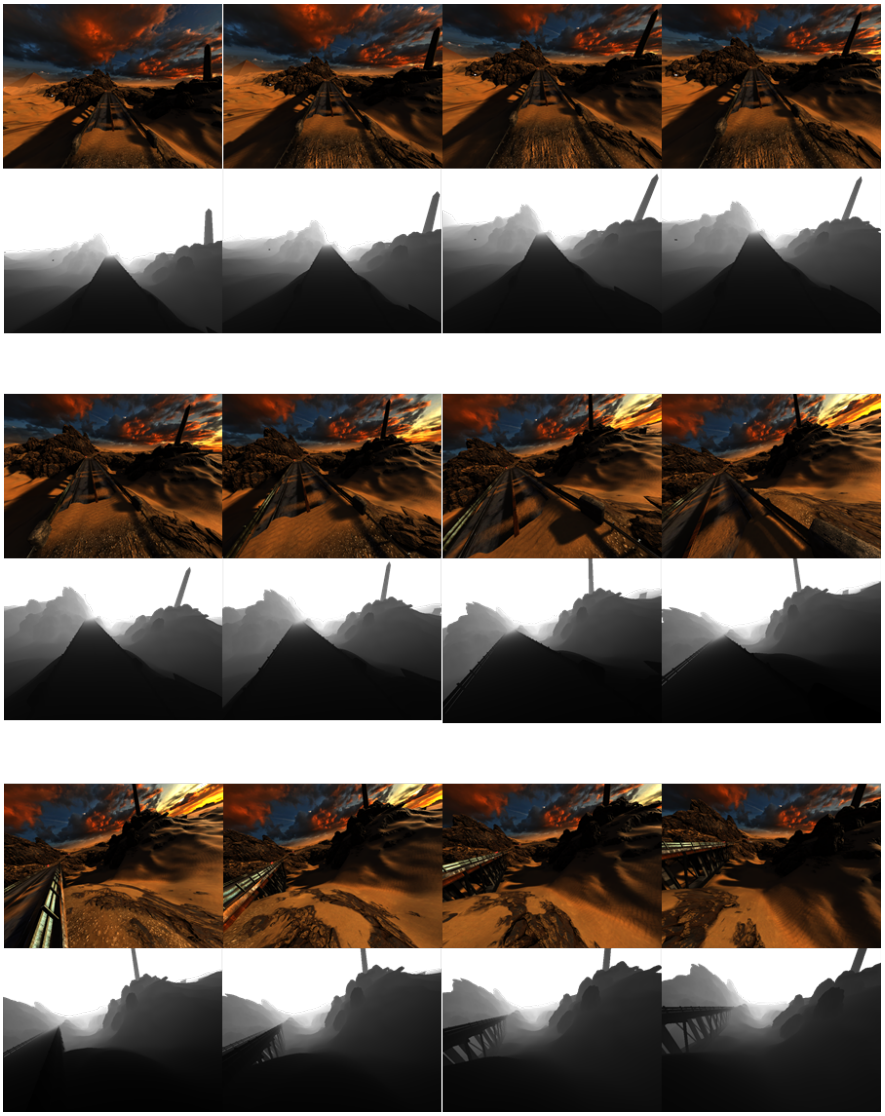


図 13: シーケンスデータセットの例

4.3 実験条件

作成したデータセットを用いて，デプス推定ネットワークで事前学習を行い，独立型ネットワーク，時系列型ネットワークを学習する実験を行った．このとき，デプス推定ネットワークと独立型ネットワークの学習にはランダムデータセットを，時系列型ネットワークの学習には，シーケンスデータセットを使用した．

学習の際，入力画像および真値として与えるデプスマップは $[0, 1]$ の範囲をとるように正規化した．また，UE4 によって出力される位置姿勢情報は，UE4 のワールド座標系における絶対位置姿勢となる．そのため，カメラの相対運動推定の真値として与える相対運動のパラメータは，入力画像間の絶対位置姿勢から計算した物を使用した．このとき，ジンバルロックなどの問題を避けるため，相対運動の回転を表す 3 自由度のパラメータは，axis-angle 表現を使用する．このとき，相対運動の基準となるカメラの座標系は右手座標系でカメラの水平方向に x 軸，垂直方向に y 軸，光軸方向に z 軸を取る．カメラの座標系の概略図を図 14 に示す．

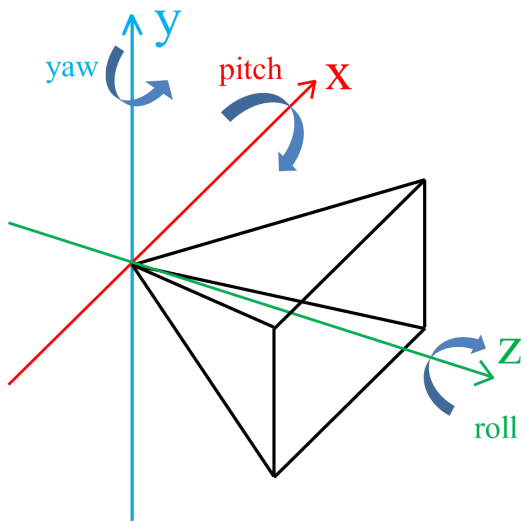


図 14: カメラの座標系の概略図

相対運動パラメータは取り得る値のスケールが大きく異なるため，相対運動パラメータをそのまま真値として与えると，取り得る値のスケール大きいパラメー

タが，モデルのパラメータ更新与える影響が他のパラメータに比べて大きくなることが考えられる．そこで，真値として与える相対運動パラメータを正規化した．回転と並進を表すベクトルをそれぞれ $r_i \in \mathbb{R}^3$，$t_i \in \mathbb{R}^3$ とし，学習に使用されるデータの数を N とし，以下のように相対運動パラメータを正規化する．

$$
\begin{aligned}
r_i{}' &= \frac{r_i}{\phi} \\
t_i{}' &= \frac{t_i}{\psi} \\
\phi &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|r_i\|^2} \\
\psi &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|t_i\|^2}
\end{aligned}
\tag{5}
$$

学習のパラメータとして，Learning Rate は 0.000001 に設定し，10 epoch 分ネットワークを学習した．このとき，1 epoch 毎にテスト用のデータから確認用のデータ 5000 ペアを使用して，真値とネットワークの推定値の誤差を出力しネットワークの汎化性能を確認し，確認用データの誤差が最も少ないモデルを採用した．また，実装にはディープラーニングのフレームワークである Chainer[32] を使用し，学習に使用する最適化手法として Adam [33] を利用した．

4.4 実験結果

4.4.1 デプス推定ネットワーク

ネットワークによるデプスマップの推定結果を図 15 に示す．図 15 では，左側が推定値，右側が真値となる．全テストデータでの，画素毎の平均誤差は 0.4521 となった．また，図 15 を見ると，真値に類似したデプスマップが推定されていることが確認できる．テストデータで最も誤差が大きいものを図 16 に示す．このとき，画素毎の平均誤差は 0.6814 となった．

図 15: デプス推定ネットワークの出力結果

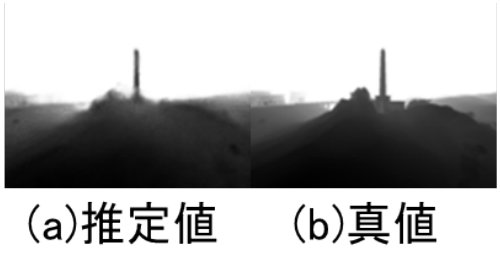


図 16: デプス推定ネットワークの出力結果（誤差最大）

表 1: ランダムデータセットにおける推定値と真値の Mean Squared Error

	pitch	yaw	roll	trans_x	trans_y	trans_z
独立型ネットワーク	0.04147	0.03722	0.04205	203.5	154.7	218.4

表 2: ランダムデータセットの真値データの統計量

	pitch	yaw	roll	trans_x	trans_y	trans_z
最大値	0.1054	0.1812	0.1174	143.3	117.2	141.9
最小値	-0.1043	-0.1804	-0.1179	-0.1425	-118.7	-146.9
平均値	0.00002.870	-0.001493	-0.00003388	-0.2771	0.2650	-0.2227
分散	0.002548	0.01005	0.002626	3325	3339	3317
標準偏差	0.05047	0.1003	0.05125	57.66	57.78	57.59

4.4.2 カメラの相対運動を推定するネットワーク

学習したネットワークでカメラの相対運動を推定する．このとき，カメラ座標系の x，y，z 軸に対応する回転パラメータは $pitch, yaw, roll$，並進パラメータは $trans_x, trans_y, trans_z$ と表す．各パラメータ毎の真値と推定値との Mean Squared Error を計算し，表 1 に示す．また，テストに使用したランダムデータセットの真値データの統計量を表 2 に示す．

表 1 を確認すると，回転パラメータでは yaw が，並進パラメータでは $trans_y$ の誤差が最も小さくなることが分かる．また，ネットワークが正しくカメラの相対運動を推定していることを確認するために，ネットワークの推定結果と真値の散布図を作成した．作成した散布図を図 17 に示す．この散布図はテスト用データの真値とその推定値をパラメータ毎に表示したものであり，横軸がネットワークの推定値，縦軸が対応する真値に対応する．図 17 から，yaw のパラメータでは，真値に近い推定値が得られていることが確認できる．一方，他のパラメータでは推定値と真値が大きく異なるデータが多いことが確認できる．

次に，時系列型ネットワークと比較するため，シーケンスデータセットを入力した際の，カメラの相対運動の推定結果を確認した．カメラの相対運動の各パラ

表 3: シーケンスデータセットにおける推定値と真値の Mean Squared Error

	pitch	yaw	roll	$trans_x$	$trans_y$	$trans_z$
独立型ネットワーク	0.1086	0.1850	0.1690	1180	987.6	1756
時系列型ネットワーク	0.06590	0.1039	0.06239	607.4	520.6	361.5

表 4: シーケンスデータセットの真値データの統計量

	pitch	yaw	roll	$trans_x$	$trans_y$	$trans_z$
最大値	0.3792	0.2861	0.3728	1943	1994	2285
最小値	-0.3853	-0.2470	-0.4169	-1396	-412.0	122.2
平均値	-0.01525	0.1031	-0.04001	345.5	719.6	1655
分散	0.01061	0.009860	0.02611	603940	285400	83060
標準偏差	0.1030	0.09930	0.1616	777.1	534.2	288.2

メータ毎の真値と推定値の Mean Squared Error を計算した結果を表 3 に示す．また，テストに使用したシーケンスデータセットの真値の統計量を表 4 に示す．

独立型ネットワークと時系列型ネットワークの結果を比較すると，全てのパラメータで時系列型ネットワークの推定結果の誤差が小さいことが分かる．また，各パラメータ毎に作成した散布図を図 18，19 に示す．2 つの推定結果の散布図を確認すると，時系列型ネットワークの各パラメータで多くのデータが真値に近い値が推定されていることが確認できる一方，独立型ネットワークでは全てのパラメータで多くのデータが真値と大きく外れた推定結果を出力していることが分かる．

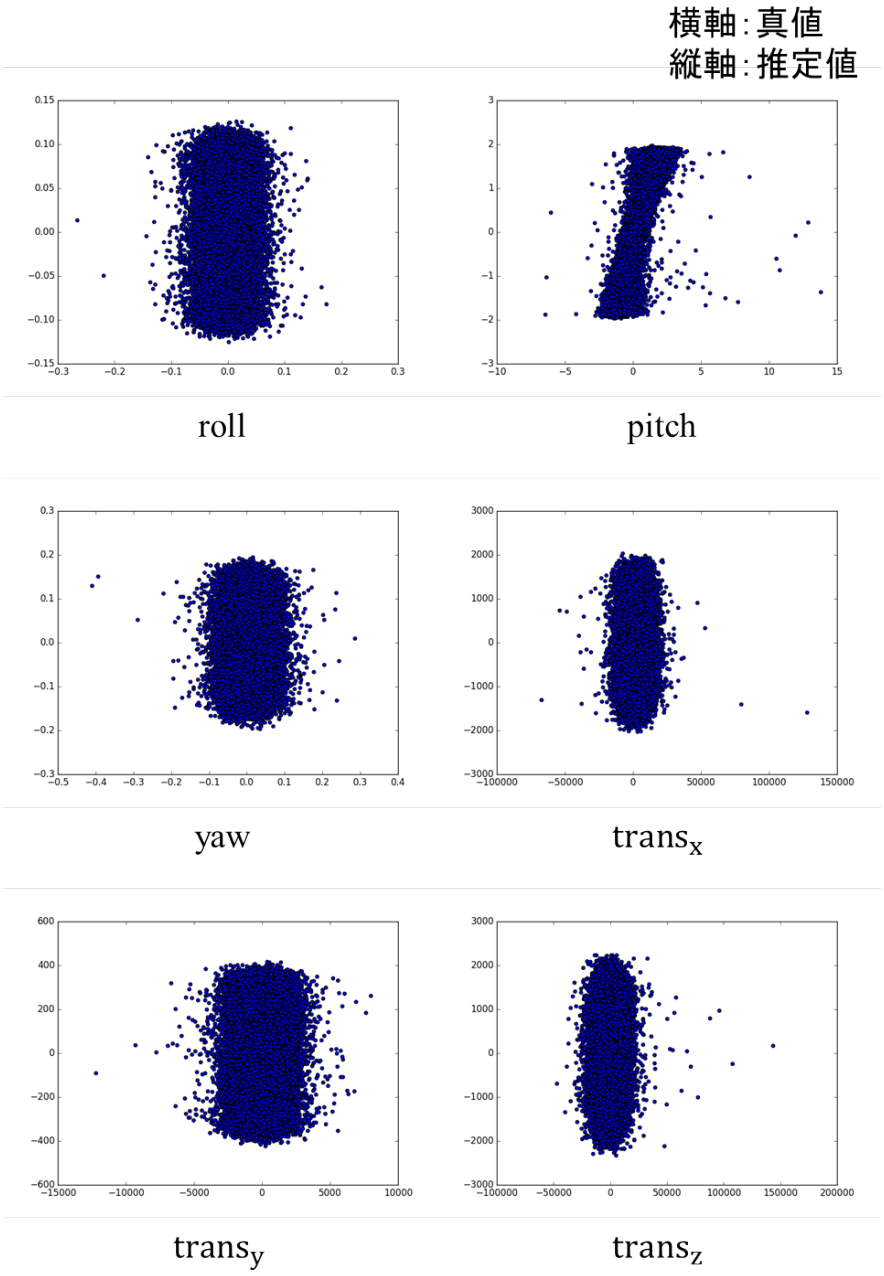


図 17: 独立型ネットワークの真値と推定値の散布図（ランダムデータセット）

4.5 考察

デプス推定のネットワークでは，推定したデプスマップが真値と類似することを確認し，また画像 1 枚での各画素の平均誤差が最大となる推定結果を示した．

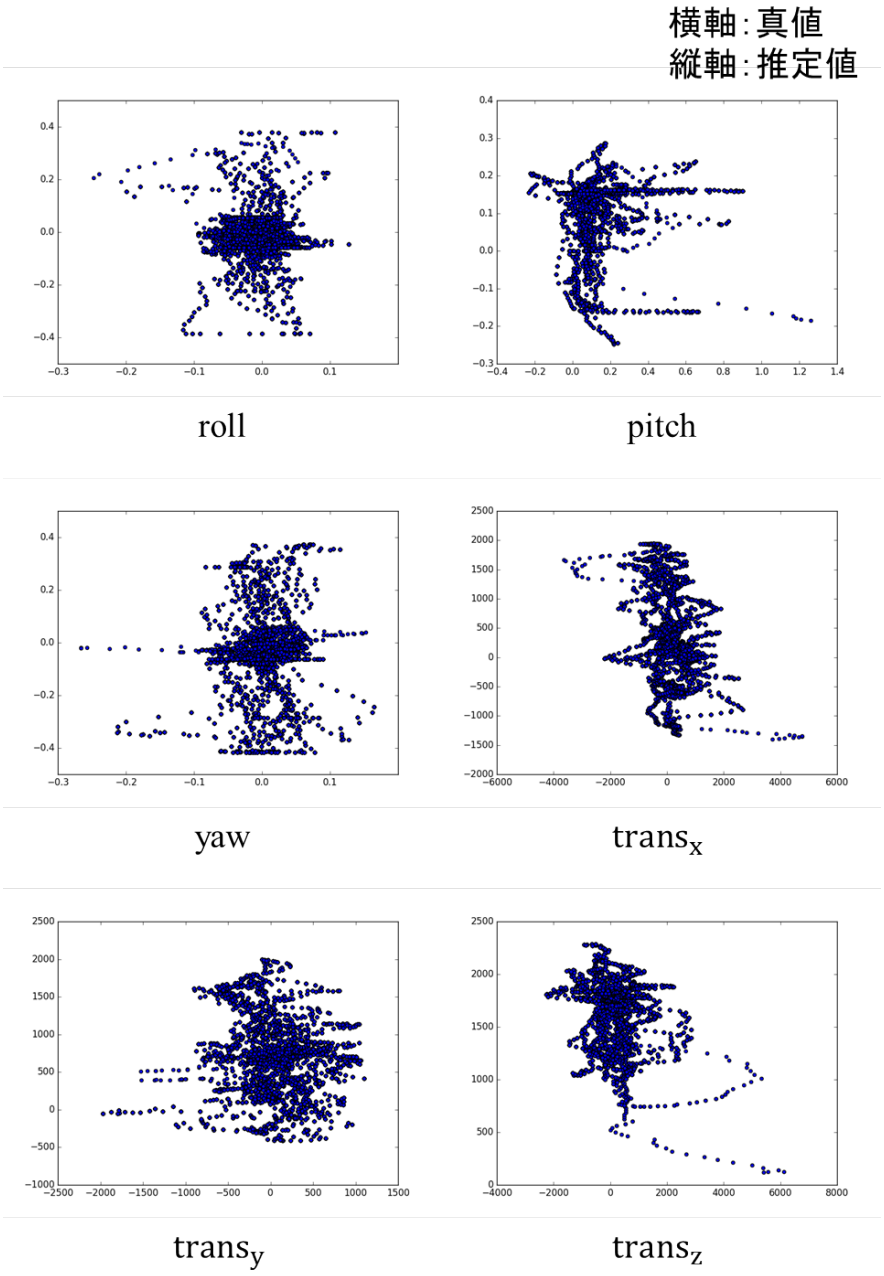


図 18: 独立型ネットワークの真値と推定値の散布図（シーケンスデータセット）

この誤差が最大のデータで画素毎の平均誤差は 0.6814 となる．推定されたデプスマップの各画素は $[0, 255]$ の範囲で値を取るため，1 画素辺りの誤差はかなり小さ

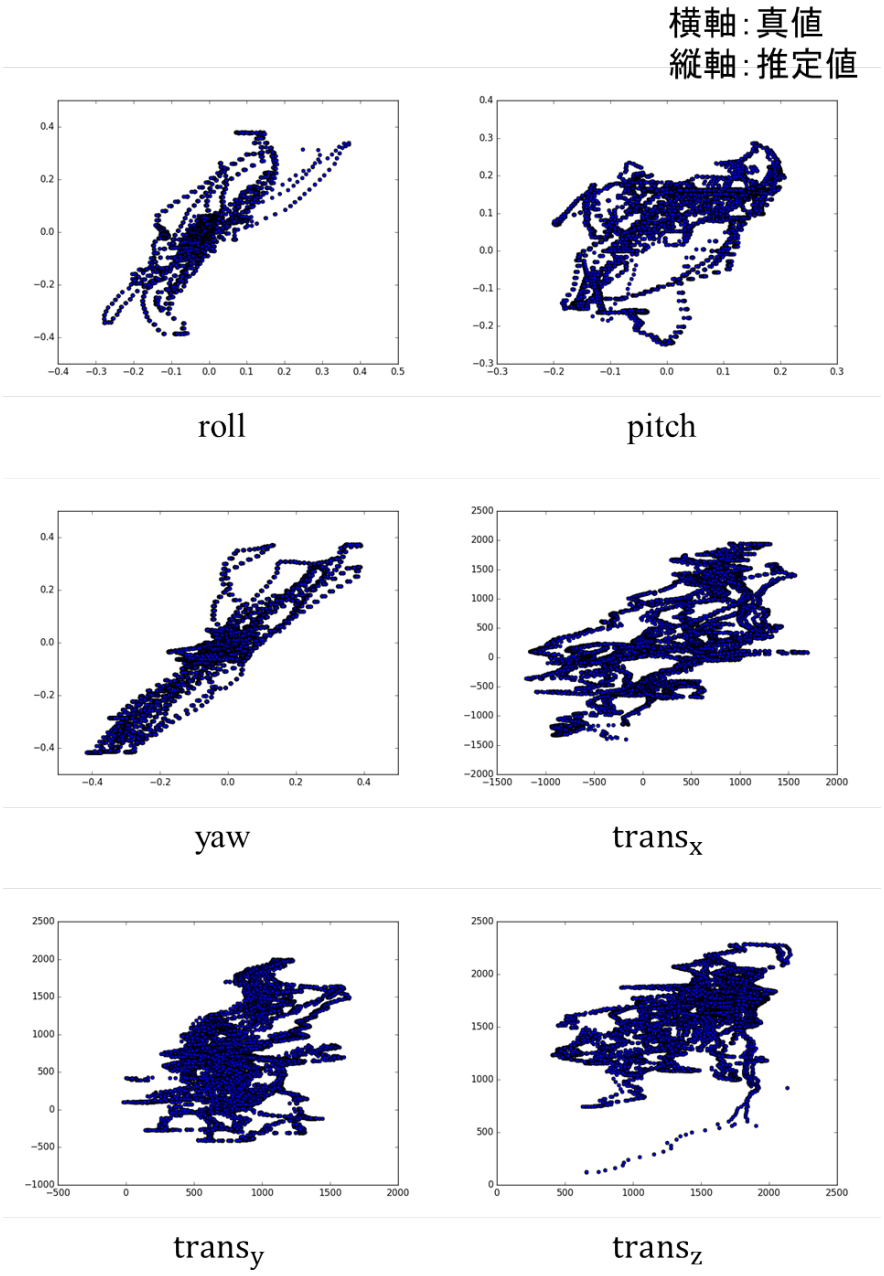


図 19: 時系列型ネットワークの真値と推定値の散布図（シーケンスデータセット）

い値となっていることが分かる．その理由として，今回作成したデータセットでは，空のシーンが多く含まれることが 1 つの要因として挙げられる．UE4 によっ

て出力されるデプス画像は $[0, 255]$ の値で出力される．シミュレーション環境内で計測されたデプス値を $[0, 255]$ の範囲に正規化する際，一定以上のデプス値は全て 255 の値として出力される．そのため，図 16 のように，空のような遠景のシーンが画像中に多く含まれる場合，画像中のデプス値の多くが 255 の値を持つこととなり，デプス画像推定が容易となると考えられる．

カメラの相対運動を推定するネットワークの学習では，前節で示された結果から，独立型ネットワークと比較して，LSTM を取り入れた時系列型ネットワークが，より高精度かつ頑健にカメラの相対運動を推定していることが分かる．

前節の散布図から，時系列型ネットワークでは真値とネットワークの推定値の間に一定の相関が見られるのに対し，独立型ネットワークのほとんどのパラメータで真値とネットワークの推定値の間に相関関係は見られなかった．その原因としては，相対運動パラメータの正規化方法が挙げられる．式 5 での正規化は，6 自由度の相対運動パラメータの内回転と並進のパラメータのスケールを揃える．これにより，学習の際に回転と並進のパラメータが取り得る値の範囲の違いのために，モデルの学習の際に回転と並進のパラメータの間で偏りが出ることを避ける．しかし，データセットの真値データが，例えば同じ回転のパラメータ間で相対運動の変化量に偏りが起きた場合，値の取り得る範囲が大きいパラメータが，モデルのパラメータ更新与える影響が他のパラメータに比べて大きくなることが考えられる．

5. まとめ

本論文では，ディープニューラルネットワークを用いて 2 枚の画像からカメラの相対運動を推定する手法を提案した．提案手法では，まず初めに 2 枚の画像からデプスマップを推定するタスクによってネットワークを事前学習する．次に，事前学習したモデルを用いて，畳み込み層のみからカメラの相対運動を推定する独立型ネットワーク，LSTM を用いて動画像の過去の情報を考慮した相対運動の推定をする時系列型ネットワークの 2 つのネットワークを学習した．

本論文では，ネットワークの学習のためにシミュレーションデータによる実験を行った．実験ではまず，ニューラルネットワークの学習に使用する大量の訓練データを用意するため，シミュレーションデータによって RGB・デプス画像，カメラの位置姿勢データを持つデータセットの作成に取り組んだ．データセットは自動で大量のデータを用意するランダムデータセットと，時系列型のネットワークの学習を目的とした，シーケンスデータセットの 2 種類を作成した．

作成したデータセットを用いて，デプス推定ネットワーク，独立型ネットワーク，時系列型ネットワークの 3 つのネットワークを学習した．実験の結果，デプス推定ネットワークでは高い精度でデプスマップを推定することを確認した．独立型，時系列型のネットワークでは，LSTM を追加することにより推定精度，頑健性が向上することを確認した．

謝辞

本研究を進めるにあたり，細やかな御指導，御鞭撻をいただいた視覚情報メディア研究室 横矢 直和 教授に心より感謝いたします．また，本研究の遂行にあたり，有益なご助言，御鞭撻をいただいたインタラクティブメディア設計学研究室 加藤 博一 教授に厚く御礼申し上げます．そして，本研究を進めるにあたり，温かいご指導をしていただいた視覚情報メディア研究室 佐藤 智和 准教授に深く感謝いたします．また，研究に関して的確な御指導，御鞭撻をいただいた視覚情報メディア研究室 中島 悠太 客員准教授（現 大阪大学 データビリティフロンティア機構 准教授）に厚く御礼申し上げます．また，本研究を遂行するにあたり，実装等多大なご助言やご指摘をいただきました視覚情報メディア研究室 大谷 まゆ氏に心より感謝いたします．また，研究室での生活を支えていただいた視覚情報メディア研究室秘書 石谷 由美氏に感謝申し上げます．また，研究室生活において様々な支援をいただいた視覚情報メディア研究室秘書 南 あずさ氏に感謝申し上げます．また，研究室生活の支援をしていただいた視覚情報メディア研究室秘書 中村 美奈氏に感謝申し上げます．最後に，研究活動だけでなく日々の生活においても大変お世話になった視覚情報メディア研究室の諸氏に心より感謝いたします．

参考文献

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," *Proc. IEEE and ACM Int. Symposium on Mixed and Augmented Reality*, pp. 1–10, 2007.

[2] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," *Proc. IEEE Int. Symposium on Mixed and Augmented Reality*, pp. 83–86, 2009.

[3] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," *Int. Workshop on Vision Algorithms*, pp. 298–372, 1999.

[4] A. Angeli, S. Doncieux, J. Meyer, and D. Filliat, "Visual topological SLAM and global localization," *IEEE Int. Conf. on Robotics and Automation*, pp. 4300–4305, 2009.

[5] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[6] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *Proc. Conf. SIGGRAPH*, pp. 835–846, 2006.

[7] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," *Int. Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[8] N. Snavely, S. M. Seitz, and R. Szeliski, "Skeletal graphs for efficient structure from motion.," *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, p. 2, 2008.

[9] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building rome in a day," *Communications of the ACM*, vol. 54, no. 10, pp. 105–112, 2011.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[11] J. Fredriksson, V. Larsson, C. Olsson, and F. Kahl, "Optimal relative pose with unknown correspondences," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1728–1736, 2016.

[12] A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac, and A. Davison, "gvnn: Neural network library for geometric computer vision," *Europian Conf. Computer Vision Workshops*, pp. 67–82, 2016.

[13] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," *Proc. Int. Conf. Computer Vision*, pp. 37–45, 2015.

[14] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," *Proc. Int. Workshop on Augmented Reality*, pp. 85–94, 1999.

[15] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana, "Virtual object manipulation on a table-top ar environment," *Proc. Int. Symposium on Augmented Reality*, pp. 111–119, 2000.

[16] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 932–946, 2002.

[17] T. Taketomi, T. Sato, and 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.

[18] 黒川陽平, 中島悠太, 佐藤智和, and 横矢直和, “特徴点の明示的な対応付けを伴わないカメラ位置姿勢推定,” 研究報告コンピュータビジョンとイメージメディア *(CVIM)*, vol. 2015, no. 60, pp. 1–4, 2015.

[19] R. I. Hartley, “In defense of the eight-point algorithm,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

[20] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[21] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Proc. Advances in Neural Information Systems*, pp. 1097–1105, 2012.

[23] F. Liu, C. Shen, G. Lin, and I. Reid, “Learning depth from single monocular images using deep convolutional neural fields,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[24] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” *Proc. Int. Conf. Computer Vision*, pp. 2758–2766, 2015.

[25] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," *Proc. Int. Conf. Machine Learning*, pp. 41–48, 2009.

[28] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for online training of recurrent network trajectories," *Neural Computation*, vol. 2, no. 4, pp. 490–501, 1990.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. Journal of Robotics Research*, pp. 1231–1237, 2013.

[30] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," *Proc. Conf. Computer Vision and Pattern Recognition*, pp. 4040–4048, 2016.

[31] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for uav tracking," *Proc. European Conf. Computer Vision*, pp. 445–461, 2016.

[32] s. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: A next-generation open source framework for deep learning," *Proc. Neural Information Processing System*, 6 pages, 2015.

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Int. Conf. Learning Representations*, 13 pages, 2014.