

# 風景知識を学習するカメラ-LiDAR-DNNによる自己姿勢推定

尾崎 亮太<sup>\*1</sup>, 黒田 洋司<sup>\*2</sup>

## Camera-LiDAR-DNN-based self-attitude estimation with learning landscape regularities

Ryota OZAKI<sup>\*1</sup> and Yoji KURODA<sup>\*2</sup>

<sup>\*1\*</sup> Graduate School of Science and Technology, Meiji University  
1-1-1, Higashimita, Tama-ku, Kanagawa 214-8571, Japan

This paper presents camera-LiDAR-DNN (deep neural network) -based self-attitude estimation with learning landscape regularities. The proposed DNN infers a gravity direction with a camera and a LiDAR. A color image and a depth image are input to the network. The depth image is obtained with the LiDAR. Convolution layers are separately applied to them, and their outputs are concatenated. The concatenated feature is input to fully connected layers, and the gravity vector is output. The DNN is pre-trained with datasets collected in a simulator. Fine-tuning with datasets collected with real sensors is done after the pre-training. Comparison with conventional methods is carried out for the performance evaluation. Real-time estimation using EKF (extended Kalman filter) is also performed.

**Key Words** : Self-attitude estimation, Mobile robotics, Deep learning, Extended Kalman filter

### 1. 緒 言

移動ロボットにおいて、ロボットの自己姿勢推定は典型的な問題の一つである。関連研究<sup>(1)</sup>では、1枚の画像から直接、重力ベクトルが推定される。この手法は、人間のよう、写真を見るだけでその写真が撮影された方向を推定することができる。これは、重力方向と風景の間に規則性があることを示唆している。この手法はカメラを用いる手法なので、費用対効果は高いが、夜間やテクスチャのない環境では上手く機能しない。上記の問題を解決するために、本論文では、カメラとLiDARの両方を用いるDNNで重力方向を推論する。さらに、これら複数のセンサを用いることで、適用可能な環境を増やすだけでなく、より高精度かつロバストに推定を行うことを目指す。

### 2. 重力方向を推論するカメラ-LiDAR-DNN

提案手法では、ロボット座標系における重力方向を推定するために、DNNに風景知識を学習させる。

**2.1 データセット収集** データセットは、カメラ画像、LiDAR点群、それに対応するロボット座標系における重力ベクトルで構成される。シミュレーションデータセットはAirSim<sup>(2)</sup>で収集される。実データは、

IMU(Xsens MTi-30)とカメラ(RealSense D435)を搭載したセンサースイートを手で持って収集される。データは、センサースイートが静止しているときのみ記録される。静的な状態のIMUデータを学習することで、動的な状態であっても、DNNが静的状態のIMUを再現することができる。つまり、ロボット自身の加速度や振動を含まない加速度ベクトルが推論される。

**2.2 データ前処理** 各入力データとラベルデータには前処理が適用される。カメラ画像には、リサイズ、正規化 ( $mean = (0.5, 0.5, 0.5)$ ,  $std = (0.5, 0.5, 0.5)$ ) が適用される。点群データはデプス画像に変換される。ラベルの重力ベクトルにはL2正規化が適用される。さらに、学習時には毎エポックで、50%の確率で各データを水平反転することで、データ水増しを行う。

**2.3 ネットワーク構造** ネットワーク構造をFig.1に示す。DNNは2つのCNN層とFC層で構成されている。カラー画像とデプス画像に対して、それぞれCNN層が適用される。ただし、カラー画像に適用される層には、ImageNet<sup>(3)</sup>で事前学習されたVGG16<sup>(4)</sup>を採用する。2つのCNN層から出力される特徴量は結合され、FC層に入力される。最終層の出力は3次元の重力ベクトルである。

**2.4 最適化** 損失関数は、ラベルデータと推定値のMSE(平均二乗誤差)である。この関数で計算される損失を最小化するように、Adam (adaptive moment

<sup>\*1</sup> 明治大学理工学研究科機械工学専攻 (〒214-8571 神奈川県川崎市多摩区東三田 1-1-1-D103) ce192021@meiji.ac.jp

<sup>\*2</sup> 明治大学理工学研究科機械工学専攻 ykuroda@meiji.ac.jp

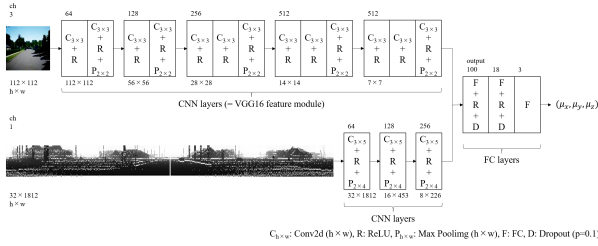


Fig. 1 Proposed network architecture.

Table 1 MAE of estimation on real data.

Method	Angle [deg]	Dataset#		
		1	2	3
Camera-LiDAR	Roll	2.07	<b>2.57</b>	<b>3.49</b>
DNN (ours)	Pitch	1.75	<b>2.86</b>	<b>2.88</b>
Camera DNN	Roll	<b>2.01</b>	3.53	5.67
	Pitch	<b>1.69</b>	3.46	4.84
LiDAR DNN	Roll	2.41	4.36	4.85
	Pitch	1.98	3.48	3.53
Statistics	Roll	15.87	15.29	19.95
	Pitch	14.81	13.42	15.91

estimation) <sup>(5)</sup>を用いてネットワークパラメータを最適化する。

### 3. 評価実験

提案するDNNを評価するため、静的検証と動的検証を行い、他手法と比較する。静的検証では、訓練データでDNNを訓練し、テストデータで評価する。動的検証では、角速度センサとDNNをEKFで統合し、リアルタイム推定を評価する。

**3.1 静的検証** 提案DNNは、10000個のシミュレーションデータサンプルを用いて事前学習し、1941個の実データサンプル (Dataset#1) を用いてファインチューニングされた。バッチサイズを100、エポック数を100とした。さらに、別のエリアで、日中に収集した443個の実データサンプル (Dataset#2) と、夜間に収集した447個の実データサンプル (Dataset#3) をテストデータとして用意した。

各データセットに対する推定誤差をTable 1に示す。ここで、‘Camera DNN’はカメラのみを用いたDNN<sup>(1)</sup>、‘LiDAR DNN’はLiDARのみを用いたDNN、‘Statistics’はラベルデータの標準偏差を表す。テストデータ (Dataset#2, #3) において、提案DNNによる推定誤差が最も小さくなった。夜間のデータ (Dataset#3) に着目すると、日中のデータ (Dataset#2) に比べて、‘Camera DNN’による推定誤差は大きくなった。

Table 2 Error of estimated attitude at last pose.

	Roll [deg]	Pitch [deg]
Gyro+DNN (ours)	<b>-2.15</b>	<b>0.46</b>
Gyro	5.70	-5.26

**3.2 動的検証** 屋外環境で約5分間、センサースイートを手で持って移動した。ただし、DNNはこのエリアのデータで訓練されていない。移動中は真値が得られないため、移動終了時の推定姿勢を評価し、誤差の蓄積を確認した。実験の開始時と終了時に、平らな床面にセンサを置き、そのときの真値を $\phi_{gt} = 0 \text{ deg}$ ,  $\theta_{gt} = 0 \text{ deg}$ と仮定した。i7-6700 CPU, GTX1080 GPU, 16 GB RAMを搭載したコンピュータで、DNNの推論計算時間は0.01-0.02秒だった。

表2に、最終姿勢での推定の誤差を示す。提案手法は、屋外走行中に、誤差の蓄積を抑制することができた。

## 4. 結 言

重力方向と風景や形状の規則性を学習するDNNを用いた自己姿勢推定法を提案した。提案したDNNは、カメラ画像とLiDARデータを入力として、そのときの重力ベクトルを推論する。静的姿勢推定の実験では、提案DNNは、カメラとLiDARの両方を用いることで、昼夜問わず、より高精度に姿勢を推定できることが示された。動的姿勢推定の実験では、提案DNNがリアルタイム推定にも適用可能であることが示された。

## 参 考 文 献

- (1) G. Ellingson, D. Wingate and T. McLain, “Deep visual gravity vector detection for unmanned aircraft attitude estimation”, *Proceedings of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (2017), pp.5557–5563.
- (2) S. Shah, D. Dey, C. Lovett and A. Kapoor, “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles”, *Field and Service Robotics*, (2018), pp.621–635.
- (3) J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database”, *Proceedings of 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2009), pp. 248–255.
- (4) K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, *arXiv preprint arXiv:1409.1556*, (2014).
- (5) D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, (2015).