

PointpartNet: 位置合わせのための 3 次元点群の部分的特徴抽出

○ 顔世荀[†], Pathak Sarthak[‡], 梅田 和昇[‡]

○ Shixun YAN[†], Sarthak PATHAK[‡], and Kazunori UMEDA[‡]

[†]: 中央大学大学院 理工学研究科, karasu@sensor.mech.chuo-u.ac.jp

[‡]: 中央大学 理工学部, {pathak, umeda}@mech.chuo-u.ac.jp

＜要約＞ 本論文は、異なる範囲の点群位置合わせのための深層学習モデルを提案する。点群をベースとした深層学習モデルが提案されたが、これらの手法は同一またはほぼ同じ視野の点群のみに適している。実際には、位置合わせをする点群の範囲が大きく異なる可能性がある。本論文では、部分的な特徴抽出に基づいた新しい深層学習モデル“PointpartNet”を提案する。部分点群の特徴を利用し、範囲の異なる点群間のマッチング領域を探索する。これにより、異なる範囲の点群を位置合わせすることが可能となる。また、実際に異なる範囲の点群の位置合わせにおいて、先行研究と比較実験を行い、高いロバスト性と精度を持つことを示した。

＜キーワード＞ 点群, 位置合わせ, 深層学習

1. 序論

複数の距離画像センサを組み合わせることで、より高いロバスト性と汎用性なシステムを構築する研究は多く存在する。近年深層学習をベースにした点群を対象とした位置合わせ手法も提案されている[1-2]。しかし、点群を位置合わせする際に、範囲が異なる点群を対象とすると位置合わせの精度が低下する傾向がある。この問題を解決するために、我々は分割に基づく探索可能深層学習手法を提案した[3]。しかし、この手法ではマッチング領域探索に留まり、位置合わせを行うにはまた別の手法を加える必要がある。そこで、本論文は分割に基づく探索可能な位置合わせ手法を提案する。

算コストが小さい特長がある。しかし、これらの手法の多くは、範囲に大きな差がある点群の位置合わせには適していない。一方、近年点群ベースの深層学習モデルが多く提案されている。PointNet[6]は点群を対象としクラス分類とセマンティックセグメンテーションを目的とした最初の深層学習モデルである。このモデルは簡単なニューラルネットワークを用いて、点群の順序不変性を実現し、高い性能の結果を得ている。また、PointNet[6]を用いた位置合わせ手法として PointNetLK[1]があげられる。しかし、この手法は二つの点群がほとんど同じでなければならないという制約がある。従って、点群が欠損しているあるいは片方がもう片方の一部の場合、位置合わせ精度が大幅に落ちる。この問題を解決するために

2. 関連研究

古くから多くの位置合わせ手法が提案されている。ICP (Iterative Closest Point) [4]は、その簡便性から、最も広く利用されている手法の一つである。しかし、外れ値がある場合に性能が劣ることや点群の初期位置と姿勢に強く依存することが知られている。正規分布変換 (NDT) [5]は、点群空間をボクセルとグリッドで等間隔に分割し、それぞれについて、求めた正規分布を用いて点群を表現し照合する手法で、計

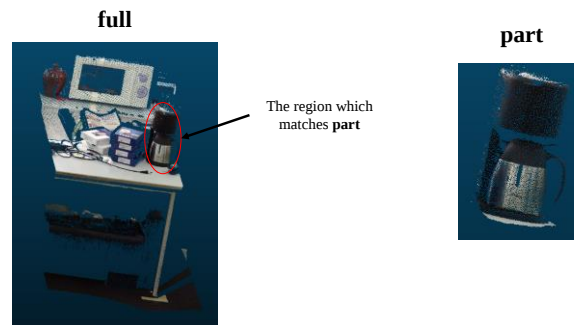


図 1 点群サンプル

MaskNet[2]というモデルが提案された。MaskNet[2]は点群同士のマッチング領域を計算するモデルである。しかし、点群大域的特徴または点群一点ずつの局所特徴しか用いないため、範囲が大きく異なる点群同士或いは姿勢が大きく異なる点群同士に対応できず、マッチングが失敗することが多い。本研究は部分的特徴抽出を用いて、範囲が大きく異なる点群同士でも位置合わせ可能な手法を提案する。

3. PointpartNet

本研究では以下の2種類の点群の位置合わせを想定する：1. LiDARなどで取得した広範囲点群 **full**. 2. RGB-D カメラなどの小範囲スキャナで取得したより詳細な点群 **part**. それぞれ図1に点群のサンプルを示す。PointpartNetの概要は以下の2ステップで構成される。

1. 部分的特徴抽出とマッチング領域探索[3]
 2. 1で見つけた対応点群同士の位置合わせ
- PointpartNetのイメージ図を図2に示す。

3.1 部分的特徴抽出とマッチング領域探索

位置合わせをする際、点群同士の密度が同じとは限らないため同じ密度になるようダウンサンプリングを行う。その際、範囲が広い点群を $\mathbf{full} = \{\mathbf{f}_1, \dots, \mathbf{f}_g\}$, 範囲が狭い点群を $\mathbf{part} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ とする。点 $\mathbf{p}_i = (x_i, y_i, z_i)$, $\mathbf{f}_j = (x_j, y_j, z_j)$ は3次元点である。また、 g は **full** の点数で、 n は **part** の点数とする。その後、部分的特徴を抽出するために **full** の分割を行う。具体的には、**full** の各点に対し、 n 個の最近傍点を選択し、 g 個サブ点群セット $\mathbf{full_group} = \{\mathbf{full_group}_i | i = 1, \dots, g\}$ を作成する。

続いて、PointNet[6]に **full_group** と **part** を入力し、それぞれの局所特徴 $\mathbf{Feature}_i = \{\mathbf{F}_{fi} | \mathbf{F}_{fi} \in \mathbb{R}^d, i = 1, \dots,$

$g\}$, $\mathbf{Feature}_p = \{\mathbf{F}_p | \mathbf{F}_p \in \mathbb{R}^d\}$ を抽出する。特徴抽出するためのPointNet[6]を図3に示す。位置に依存しない部分的特徴を得るため、PointNet[6]への入力点群の重心が原点に一致するように平行移動させる。さらにスケールに対しロバスト性を高めるために、**full** に応じたユニットボックス[1]へ正規化も行う。

特徴抽出後、**part** は **full** のどの領域が最も類似しているかを探索する。つまり、ここで **global translation** $\mathbf{t}_G$ を推定する。 $\mathbf{Feature}_p$ と $\mathbf{Feature}_f$ を MLP_1 (Multi-Layer Perceptron) に入力し、マッチング尤度 $\text{Score} = \{\text{Score}_i | i = 1, \dots, g\}$ を計算する。 Score_i は、**full_group** と **part** が類似している場合に1、類似していない場合0となる。 MLP_1 のサイズは(256, 128, 128, 1)となる。そして、最も高いスコア Score_H を選択し、それと対応する $\mathbf{full_group}_H$ が **part** とマッチング領域と判断される。また、 $\mathbf{full_group}_H$ と **part** の位置関係を **global translation** $\mathbf{t}_G$ とする。

3.2 位置合わせ

選択されたマッチング領域 $\mathbf{full_group}_H$ と **part** の位置合わせを行う。位置合わせは大域位置合わせと局所位置合わせの2つの部分に分けられる。

ここで、マッチング領域探索の際 $\mathbf{t}_G$ を求めたため大域位置合わせにおいて **global rotation** $\mathbf{R}_G$ のみを求めればよい。大域位置合わせはできるだけ早く真値に近い $\mathbf{R}_G$ を求める手順である。ここでは図2のようにDirectNet[7]を用いて大域位置合わせを行う。 $\mathbf{full_group}_H$ と **part** の大域特徴を MLP_2 の入力として $\mathbf{R}_G$ を計算する[1]。 MLP_2 のサイズは(512, 256, 128, 64, 3)である。

一方、局所位置合わせはさらに高い精度で点群同士の位置姿勢を求めるものである。ここでは、大域位置合わせ後の $\mathbf{full_group}_H$ と **part** の局所特徴空間の最近傍点を見つけて、SVD (Singular Value Decomposition) を用いて剛体変換行列を求める。

3.3 学習と損失関数

学習では、マッチング領域探索と大域位置合わせの2段階の学習を行う。また、学習処理の効率化のため、学習で用いる大域位置合わせは $\mathbf{full_group}_H$ では

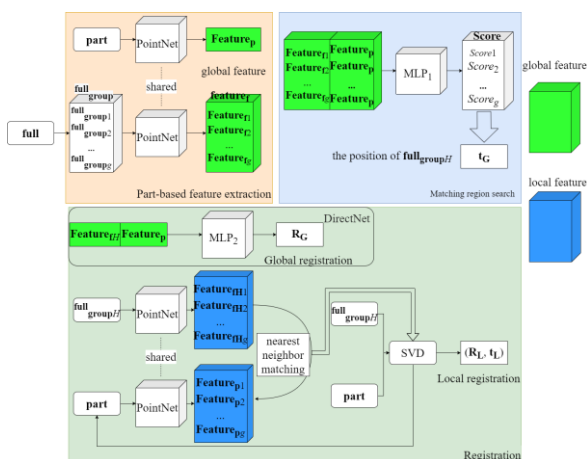


図2 PointpartNetのイメージ図

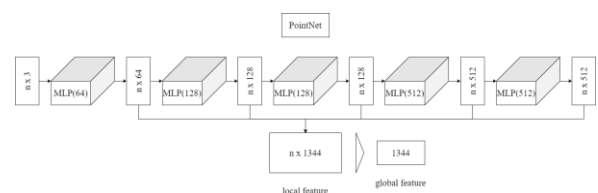


図3 PointNet[6]のイメージ図

OS2-2

なく、マッチング領域探索の真値に相当する $\mathbf{full}_{group_{gt}}$ を入力としている。

提案モデルでは、最も高いスコアを選択するマッチング領域探索と、大域位置合わせの2つのネットワークを学習させるため、損失関数はマッチング損失 $loss_m$ と位置合わせ損失 $loss_r$ の2つに分かれる。 $loss_m$ は分類問題でよく使われる負の対数尤度損失である。

$$loss = loss_m + loss_r \quad (1)$$

$$loss_m = -\log \left(\frac{\exp(\text{Score}_{label})}{\sum_{i=1}^g \exp(\text{Score}_i)} \right) \quad (2)$$

ここで、 $label$ は中心点 $\mathbf{f}_{gt}$ のインデックスで、真値である。 $loss_r$ は位置合わせ $\mathbf{G}_G = \{\{\mathbf{R}_G, \mathbf{t}_G\}\}$ と真値の位置姿勢を表す $\mathbf{G}_{gt}$ との差を最小化する。そこで MSE (Mean Square Error) を用いて以下のように示す。

$$loss_r = \|(\mathbf{G}_G)^{-1} \cdot \mathbf{G}_{gt} - \mathbf{I}_4\|_F \quad (3)$$

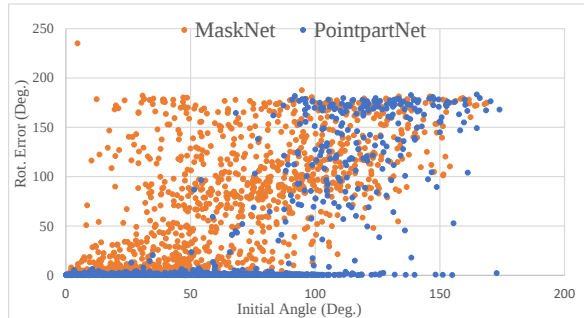
4. 実験

PointpartNet の有用性を検証するために、従来手法である MaskNet[2] との比較実験を行った。実験ではデータセットは 3D モデル ModelNet40[8] を用いる。ModelNet40[8] は 40 種類を持つ 3D CAD モデリングである。20 種類のモデリングは学習時で使用してお

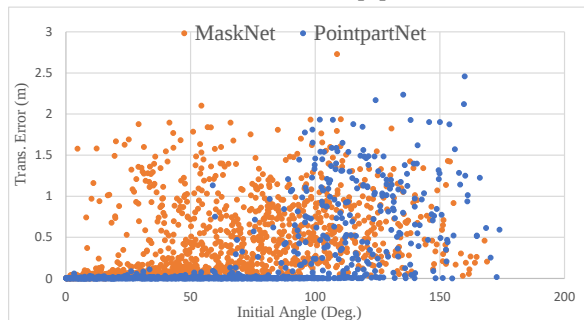
り、残り 20 種類のモデリングはテスト時に使用している。 $\mathbf{full}$ は ModelNet40[8] からリサンプリングした点群であり、 $\mathbf{part}$ は最近傍点で $\mathbf{full}$ から選択した点群である。また、 $\mathbf{part}$ には正規分布に従う平均 0 m、分散 0.05 m^2 のノイズを加えた。学習中で用いる姿勢の真値は $[0, 90]^\circ$ からランダムで決定したもので、位置の真値は $[0, 1.57] \text{ m}$ からランダムで決定したものである。テスト中未学習の位置姿勢のロバスト性を検証するために姿勢と位置の真値はそれぞれ $[0, 180]^\circ$ と $[0, 3.14] \text{ m}$ から決定した。そして、姿勢と位置の推定誤差がそれぞれ 10° と 0.1 m 以下の場合、推定成功と見なす。以上の条件で提案モデルと MaskNet[2] の比較実験を行った。実験条件を以下に示す。

1. 使用するライブラリ：python 3.6.10 と pytorch 1.7.0
2. 学習時のバッチサイズ：16
3. 学習時のエポック：200
4. 学習時点群セット数：5114
5. テスト時点群セット数：1266
6. 学習時 $\mathbf{full}$ と $\mathbf{part}$ の点数：256 と 64
7. PC のスペック：3.60GHz Intel i7-9700K と GeForce RTX 2070 SUPER

実験結果を図 4-6 と表 1 に示す。回転推定成功率は MaskNet[2] の 29.38% から 80.25% に向上した。回転平均誤差は MaskNet[2] の 68.95° から 26.40° に減少

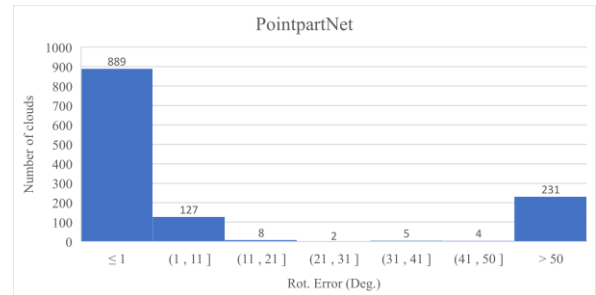


(a) Mean Rotation error of PointpartNet and MaskNet[2]

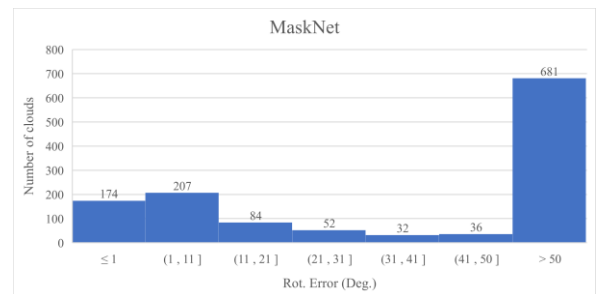


(b) Mean Translation error of PointpartNet and MaskNet[2]

図 4 PointpartNet と MaskNet[2] の実験結果

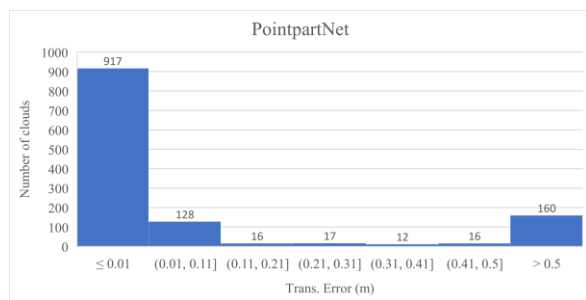


(a) Histogram of Mean error for PointpartNet

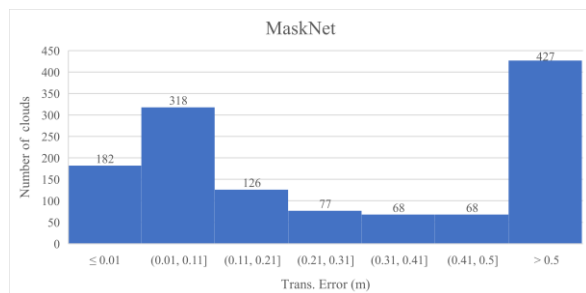


(b) Histogram of mean error for MaskNet[2]

図 5 PointpartNet と MaskNet[2] の誤差ヒストグラム比較 (姿勢)



(a) Histogram of Mean error for PointpartNet



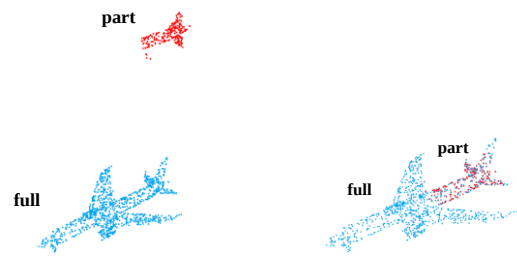
(b) Histogram of mean error for MaskNet[2]

図 6 PointpartNet と MaskNet[2]の誤差ヒストグラム比較 (位置)

した. また, 平行移動推定成功率は MaskNet[2]の 38.47%から 82.31%に向上した. 平行移動平均誤差は MaskNet[2]の 0.417m から 0.160m に減少した. MaskNet[2]では 20°を超えたとき回転と平行移動誤差が急激に増大する傾向がある. 一方, PointpartNet では MaskNet[2]と比べより安定な結果が得られた. 位置合わせのサンプルを図 7 に示す. また, PointpartNet の各手順の処理時間を表 2 に示す. 各処理の中, 部分的特徴抽出の処理時間がほかの処理時間と比べて大幅に増えた. それは **full** の全点を中心として **full_{group}** を切り分け処理及び特徴抽出がかなりの時間が必要があるためである.

5. 結論

本論文では, 異なるサイズの点群に対して, 部分的な特徴抽出を行う深層学習を用いた位置合わせを提案した. PointpartNet は点群を分割し部分点群の特徴抽出することでマッチング領域を探索する. この手法では, 従来の手法では適用できなかった異なる範囲の点群の位置合わせに成功した. また, 従来手法との比較実験により, PointpartNet がよりロバストかつ高精度であることが示された. しかし, 提案モデルの処理が重い場合, GPU メモリが足りない可能性がある. 入力点群はダウンサンプリングを行わないと学習できないという問題点がある. また, 提案



(a) before registration (b) after registration

図 7 位置合わせサンプル. 青: **full**, 赤: **part**.

表 1 各手法の実験結果

姿勢	成功率	平均誤差(°)
MaskNet[2]	29.38%	68.95
PointpartNet	80.25%	26.40
位置	成功率	平均誤差(m)
MaskNet[2]	38.47%	0.417
PointpartNet	82.31%	0.160

表 2 PointpartNet 各手順の処理時間

部分的特徴抽出	マッチング領域探索	大域位置合わせ	局所位置合わせ
0.306 s	0.003 s	0.005 s	0.119 s

モデルは **part** の **full** の前提で処理しているため, **part** が **full** 中に存在するかどうかの判断はしていない. 従って今後は, **part** が **full** 中にあるかどうかの検知機能も追加する予定である. また, PointpartNet は現状カメラの視野など考えず, 最近傍点で点群を切り分けしてマッチング領域探索しているため, **part** が **full** の最近傍点で切り出された点群でない場合, 直接使えない問題がある. 従って今後は, カメラの視野, 位置姿勢などを考慮した手法に改善することも目標とする.

謝辞

本研究は JSPS 科研費 19H04191 の助成を受けたものである.

参考文献

- [1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun, Srivatsan and Simon Lucey: “PointNetLK: Robust & Efficient Pointcloud Registration Using PointNet” Proc. Computer Vision and Pattern Recognition (CVPR), pp. 7156-7165 (2019)
- [2] Vinit Sarode, Animesh Dhagat, Rangaprasad Arun, Srivatsan, Nicolas Zevallous, Simon Lucey and Howie Choset: “MaskNet: A Fully-Convolutional

OS2-2

- Network to Estimate Inlier Points” 2020 International Conference on 3D Vision (3DV), pp. 1029-1038 (2020)
- [3] 顔 世荀, Sarthak Pathak, 梅田 和昇: “3次元点群データの位置合わせのための深層学習を用いた部分的特徴抽出”, 2021年度精密工学会秋季大会学術講演会講演論文集, G98 (2021)
- [4] Paul J. Besl and Neil D. McKay: “A method for registration of 3-D shapes” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256 (1992)
- [5] Peter Biber and Wolfgang Strasser: “The normal distributions transform: a new approach to laser scan matching” Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), vol.3, pp. 2743-2748 (2003)
- [6] Charles R. Qi, Hao Su, Kaichun Mo and Leonidas J. Guibas: “Pointnet: Deep learning on point sets for 3d classification and segmentation” Proc. Computer Vision and Pattern Recognition (CVPR), pp. 77-85 (2017)
- [7] Akiyoshi Kurobe, Yusuke Sekikawa, Kohta Ishikawa and Hideo Saito: “CorsNet: 3D Pointcloud Registration by Deep Neural Network” in IEEE Robotics and Automation Letters, vol. 5, no. 3, pp. 3960-3966 (2020)
- [8] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao: 3d shapenets: “A deep representation for volumetric shapes” In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1912–1920 (2015)