

三次元デジタルデータの物体検出に関する検証

中 川 高 晃^{*1}前 田 一 輝^{*2}

【要 旨】

データを点の集合体として扱う点群データの活用が広がる中、取得したデータを独自に処理するにはハードルがある。今回、3D スキャナを用いて複数のモデルが混在した点群データを取得し Open3D を用いた特定モデルとの照合・識別を行い、有用性や負荷について検証を行った。

1 はじめに

物体や空間をスキャンして得られる三次元デジタルデータは、工業製品の計測や測量、自動運転、文化財のデジタルアーカイブ等、多様な分野で活用されている。近年の計測技術の進歩による測定精度の向上と、持ち運びが便利なハンディタイプの測定機やドローンへの搭載といった機器の普及もあり、三次元デジタルデータは比較的容易に扱えるようになってきている。

3D スキャナなどを利用して取得した三次元デジタルデータは、その中身が点の集合体であり点の1つ1つが座標を有している。通常、データの張り合わせや編集といった解析には専用のソフトウェアなどで行われることが多い。これらはユーザーインターフェイスに優れ、用途に応じてフィルタリング等の解析編集といった便利な機能が利用できる。その一方で、導入コストの問題や点群処理の内容を把握したうえで個々の目的に応じたカスタマイズな解析を行いたいといったケースがある。今回、オープンソースライブラリを使った点群解析の有用性や実装負荷の検証として、当センターにある装置を用いて点群データの収集を行い点群の中から指定した物体検出を行った。

2 検証方法

2.1 構成

表1に検証環境を示す。点群処理ソフトウェアには Open3D¹⁾ を利用した。Open3D はオープンソースライブラリで点群データのサンプリングや法線推定、位置合わせなど様々な処理関数が提供されている。プログラミング言語 Python にも対応しており、少ないコーディングで他の Python ライブラリとも連携がとりやすい。点群データの取得には、中丹技術支援室にある VIVID9i を利用した。スキャナにより取得したデータに対するフィルタリングや穴埋め等の補正は測定機上ではせず点群処理は全て Open3D により行った。

表1 検証環境

点群処理	Open3D(0.19.0)
言語	Python(3.12.10)
3Dスキャナ	VIVID9i

Open3D は、コーディング操作により多様な点群処理をカスタマイズしながら実装できる一方で、扱える情報量も非常に多いものとなっている。公式サイトはチュートリアルが実装されており、英語であるが関数の説明や例が詳しく示されているほか、書籍であれば日本語で解説されている資料もあり²⁾、これらを参考に検証を行った。

*1 中丹技術支援室 副主査

*2 中丹技術支援室 室長

2. 2 スキャンシーンと対象モデル

図1、2に 3D スキャナを使った対象物の撮像イメージとスキャンシーンを示す。形状の異なる3つの物体をテーブル上に並べて、一方向からスキャンを行っている。スキャナの受光レンズから物体までの距離は約 1.1m となっている。図3は取得したスキャンシーンの点群データである。図からわかるとおり、スキャンされない側面や原理的にスキャン困難な箇所は点群データが欠落している。

検出対象とした物体の個々の点群モデルを図4に示す。実環境では製品 CAD データを予め保有している場合もあるが、今回は CAD モデルを作成して STL で出力後に Open3D で点群データに変換を行った。スキャナで取得した点群データと対象モデルの点群データとを比較照合し、位置合わせをすることで特定の物体検出を行った。



図1 撮像イメージ

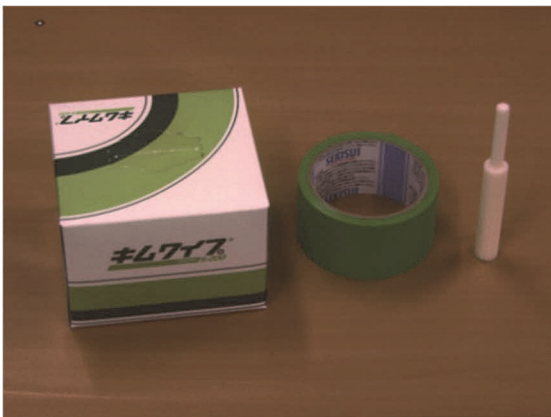


図2 スキャンシーン

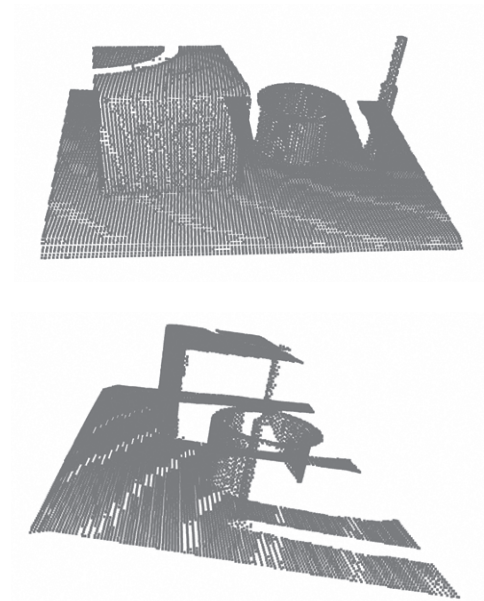


図3 取得点群データ

(上図：撮影方向、下図：側面方向)

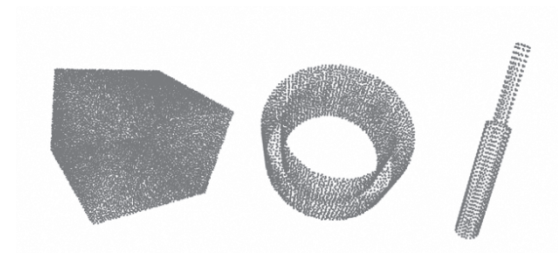


図4 検出対象モデルの点群データ

2. 3 点群処理

点群データから対象を指定した物体検出を行う場合、取得データとモデルデータの点を全て使って比較照合するには計算量が膨大となる。そこで物体を構成している全ての点群からキーとなる点をいくつか選択し、その点が持つユニークな情報となる特徴量を求める。取得データとモデルデータでお互いのキーポイントを比較照合し特徴量の近い点同士を対応付けする。特徴量の決め方には、キーポイント周辺の法線ベクトルの分布状況を利用したり、複数のキーポイント間の位置関係を利用するなど様々な方法が検討されているが、ここ

では Open3D のチュートリアルにもある FPFH (Fast Point Feature Histograms) を用いた。また、点の対応付けにおいて、全てを誤りなく対応させることは難しく外れ値の影響をおさえる必要がある。Open3D では RANSAC (RANdom SAmple Consensus) とよばれる推定法を実装している。これは、点群処理以外にも使われる手法で、ランダムにサンプリングした値から推定モデルを作り、それをサンプリングした元のデータ全体に適用した時の精度の高さを検証するという処理を繰り返す。これらの処理によって取得データとモデルデータの大きな位置合わせ後に ICP (Iterative Closest Point) アルゴリズムにより精密な位置合わせを行っている。予め対象同士をある程度近づけた状態にする必要があるが、対応する点同士の距離を最小化するような姿勢変換を行う。大まかな処理の流れは以上のとおりであり、このほかスキャナで取得した多量の点データをサンプリングしたりといった点群処理を Open3D で行っている。

3 結果

各モデルの検出結果について図5に示す。赤色がモデルデータでありスキャナによる取得データと比較検出して位置合わせを行っている。一致具合が見やすいように、取得データの点群数は間引かずに表示してある。円筒形状の2種類は比較的精度良く検出することができたが、直方体の物体が正しく検出されていない。これは図3を見てわかるとおり、スキャナにより取得した点群データの時点から、直方体を構成する面全体に対して2面しかスキャンできておらず、また上面で一部データの欠落がある。これらの悪条件と合わせて物体をのせているテーブルの平面要素に引きずられた結果と思われる。そこで、図6に示すように、予め検出誤りの原因となりうるテーブルの平面データの除去を行ったうえで再試行した結果、検

出精度が向上した。ただし、この場合でも上手く面同士の検出がされていないことがわかる。モデルデータが理想的な周囲の点群データを使って特徴量を計算しているのに対して、スキャナによる取得データでは欠損がある状態で周囲の点群データから特徴量を計算している。この違いから特徴量が一致せずに点の対応誤りが発生していると考えられ、キーポイントの選択方法や特徴量の計算に用いる周囲の点群の個数・距離といったパラメータの調整、特徴量の決め方といった手法を検討することにより改善されると思われる。

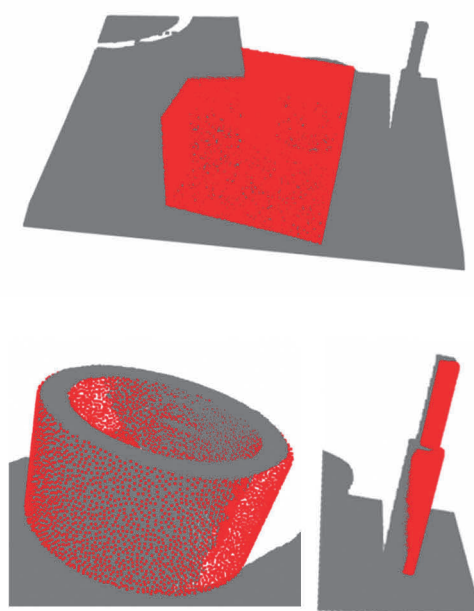


図5 検出結果

(上図：直方体、下2図：円筒形状2種)



図6 平面除去を行っての再試行 (直方体)

4 まとめ

今回、当センターに設置されているスキャナを用いて取得した点群データをもとに、オープンソースライブラリを使った点群解析の有用性や実装負荷について検証した。点群のサンプリングやフィルタリング、位置合わせなど様々な機能が実装されており、目的に応じて柔軟な点群処理が可能であることが確認できた。その一方で、操作の自由度が高い反面、どのような処理が適切なのか、

状況に応じた適切な点群処理の内容についてよく検討することが重要であることも確認できた。

(参考文献)

- 1) <https://www.open3d.org/>
- 2) 金崎朝子，秋月秀一，千葉直也：詳解3次元点群処理 Python による基礎アルゴリズムの実装，講談社（2022）