

# 固定物のみからなるメタ AR 空間構築のための可動物検出手法の試作 Developing an Unfixed Object Detection Method Constructing a Meta-AR Space Composed of Only Fixed Objects

安江 洸希<sup>†</sup> 菊地 真人<sup>†</sup> 大園 忠親<sup>†</sup>  
Koki Yasue Masato Kikuchi Tadachika Ozono

## 1 はじめに

LiDAR スキャナを搭載した汎用的なモバイルデバイス (例えば, Apple 社の iPad Pro) の性能向上により, 誰でも容易に 3 次元 (3D) の現実空間の詳細な 3 次元形状モデル (3D マップと呼ぶ) を作成することが可能になりつつある. これらの汎用機器の性能向上は, 拡張現実感技術 (AR: Augmented Reality) の性能向上に貢献している. しかし, AR を使用している現地 (AR 空間) のユーザと, その場所にはいないユーザとの間のインタラクションを支援するためのフレームワークの整備が不十分である. 本稿では, そのようなフレームワークの実現に必要な可動物の検出手法について述べる.

## 2 メタ AR 空間のための可動物検出とその課題

本研究では, 現地 (AR 空間) および遠隔地 (仮想空間) との間での効果的なインタラクションを支援するための要素技術として, AR 空間と仮想空間との間でオブジェクトの共有が可能な仮想空間であるメタ AR 空間を開発している [1]. ここでのメタ AR 空間とは, AR 空間をメタに俯瞰しながら操作可能な空間を意味している. ここでは, Apple 社の AR 技術である ARKit により, iPad Pro に搭載された LiDAR スキャナを用いて現実空間を計測し, 現実空間の形状を 3D 点群として取得 (3D スキャン) することでメタ AR 空間を作成する.

メタ AR 空間において, 現実空間内における椅子やドアといった動くことが想定される物 (可動物) に関しては, メタ AR 空間でも移動や消去などの操作を可能にすることが期待される. これにより, 現実空間とメタ AR 空間との間における双方向インタラクションの機能向上を期待できる. しかし, 先行研究のメタ AR 空間において, 3D スキャンして取得した点群, すなわち, 3D スキャンされた現実空間は 1 つのオブジェクトとして構築されており, 可動物を分離するための手作業が必要であり, 非常に手間であった.

可動物の検出には, 可動物のみ位置が異なる複数の 3D 点群を比較することとした. すなわち, 3D スキャンして得られたある状況 A と状況 B とは可動物の位置が異なる状況 B に対して 3D 点群を比較する. ここでは可動物以外を固定物とする. 図 1 は, 可動物検出の流れを

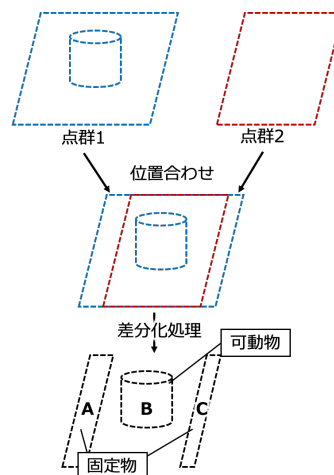


図 1 点群の位置合わせと差分処理

表している. まずは, 可動物が含まれている点群 1 (図 1 上部左) と含まれていない点群 2 (図 1 上部右) の間の差分を取る. この際に点群 1 と点群 2 の対応点 (同じ場所を表す点) を見つける位置合わせが必要である.

点群の差分化や位置合わせといった処理は, Cloud Compare (<https://www.danielgm.net/cc/>) や Point Cloud Library (<https://pointclouds.org/>) 等の既存ソフトウェアで実現可能であるが, 最新のモバイルデバイスでもこれらの処理を短時間で実行することは困難である. これらの, 処理の実行には 100GB 超のメモリを要する場合もあり, 現在のモバイルデバイスが標準的に搭載しているメモリ (1~8GB 程度) では不十分である. 例えば, 点群間の差分の取得に必要な, 点群間の位置合わせを行う手法である Iterative Closest Point (ICP) アルゴリズムとその亜種 [2] を ARKit で取得した約数十万点から成る点群で適用する場合, iPad Pro (第 4 世代) で数十分以上を要することが予備調査により明らかになっている.

ICP アルゴリズムは, 対応点の探索と位置合わせのための変換パラメータ (回転および並進) を推定する手法である. 高速化には点群のダウンサンプリングやグループ化等により, 処理すべき点の個数を減らすことが有効であり, 適切な点を選ぶことが重要である. 位置合わせの精度は点群のオーバーラップ率に大きく依存する [3]. すなわち, オーバーラップしている点群間の部分集合を処理すればよい. 本研究では, 点群と同時に取得した写真 (画像) を利用し, ICP を適用する点群を絞り込む手法を検討した. 画像間の位置合わせを行い, 画像間の対応点を取得する. ここで取得した対応点を 3D 点群に変

<sup>†</sup> 名古屋工業大学大学院工学研究科工学専攻情報工学系プログラム, Computer Science Program, Department of Engineering, Graduate School of Engineering, Nagoya Institute of Technology

換して ICP を適用する。画像間の対応点取得は、3D 点群の対応点取得よりも高速なため、全体的な計算時間の短縮が期待される。例えば、高速な画像間の対応点取得手法として、AKAZE がある。AKAZE は、拡大縮小、回転に不変であり、他の SIFT や SURF のような特徴マッチングよりも処理速度や頑健性の面で優れている [4]。

### 3 提案手法

本研究における可動物検出のアルゴリズムを Algorithm 1 に示す。入力は、可動物が含まれている点群  $P_1$ 、可動物が含まれていない点群  $P_2$ 、 $P_1$  取得時の画像  $img_1$ 、 $P_2$  取得時の画像  $img_2$  の 4 つである。出力として可動物の点群  $P_{unfixed}$  を返す。以降、図 2 の Algorithm 1 について説明する。

#### 3.1 画像を用いた位置合わせ

初めに画像間で特徴点マッチングを行って特徴点を抽出し、この特徴点をもとに対応点の探索を行う。まずは、AKAZE により特徴点の集合  $Q$  を取得する (3 行目)。次に  $Search$  関数で対応点を探索する (4 行目)。ここでの探索では、クロスチェック法を用いて対応点集合  $Q_{2D}$  を得る。すなわち、 $Q$  中の全ての特徴点間の類似度を求め、最も高いものを対応付ける総当たりマッチングを  $img_1$  から  $img_2$  に対して行う。その後、この処理を  $img_2$  から  $img_1$  でも同様に行って両方で一致したものを対応点とした。ここで扱う特徴点は高々数十個程度を想定しており、総当たり法で十分である。ここでは、対応点を 20 点集める (5~6 行目)。

対応点が求まったら  $Transform$  関数で 2D 座標を 3D 座標  $Q_{3D}$  に変換する (8 行目)。 $Q_{3D}$  を ICP アルゴリズムにより位置合わせを行い、変換パラメータである回転  $rot$  および並進  $trans$  を得る (9 行目)。ここでは、確信度  $confidence$  が 0.01 以下になるまで処理を続ける (10 行目)。 $confidence$  は、対応点間の差の 2 乗和に基づいている。

#### 3.2 3D 点群の位置合わせと差分処理

$Align$  関数は、 $rot$  と  $trans$  により、 $P_1$  と位置合わせされた  $P_2$  として、 $P'_2$  を返す (11 行目)。 $Diff$  関数は、点群の各点ごとに一定の範囲の点を削除して差分部分の点群  $P_{diff}$  の抽出を行う (12 行目)。

$P_{diff}$  は、3D スキャン時の誤差や撮影範囲の違いなど、可動物以外の差分を含むため、 $Judge$  関数はこれに対応する処理を行う (13 行目)。不要部分の判定方法としては、差分として出てきた点群を構成している点群の要素に着目する。本手法で扱う点群は、可動物が  $P_1$  および  $P'_2$  のいずれかの点群にしか含まれていないため、 $P_{diff}$  に含まれる可動物に該当する点群は、 $P_1$  および  $P'_2$  の内のいずれかの点群しか含まない。例として図 1 の下部の点群 A, B, C は、可動物として判定される可能性がある。点群 1 と点群 2 の両方から構成されているた

#### Algorithm 1 画像を利用した可動物検出

```

1: function DETECTION( $img_1, img_2, P_1, P_2$ )
2:   while True do
3:      $Q = Akaze(img_1, img_2)$ 
4:      $Q_{2D} = Search(Q)$ 
5:     if  $|Q_{2D}| < 20$  then
6:       continue
7:     end if
8:      $Q_{3D} = Transform(Q_{2D})$ 
9:     ( $confidence, rot, trans$ ) = ICP( $Q_{3D}$ )
10:    if  $confidence < 0.01$  then
11:       $P'_2 = Align(P_2, rot, trans)$ 
12:       $P_{diff} = Diff(P_1, P'_2)$ 
13:       $P_{unfixed} = Judge(P_{diff})$ 
14:      break
15:    end if
16:  end while
17:  return  $P_{unfixed}$ 
18: end function

```

図 2 可動物検出アルゴリズム

め可動物に該当しない (すなわち固定物) を表す差分である。点群 B に関しては点群 1 のみから構成されているため可動物であると判定を行う。説明を省略するが、 $Judge$  関数は、この処理のために他の点群を利用する。

### 4 おわりに

画像間の特徴マッチングを用いて 2D の対応点を求め、この対応点で位置合わせした際の変換パラメータを用いて、3D 点群の位置合わせを行う手法について検討した。また、差分処理した後の点群の要素に着目して可動物の判定を行なった。今後は、得られた点群の差分から複数のオブジェクトを抽出処理する必要がある。また、抽出されたオブジェクトの隠れた面の補完も必要である。

#### 謝辞

本研究の一部は JSPS 科研費 19K12266 の助成を受けたものです。

#### 参考文献

- [1] 安江 洗希, 大園 忠親, 新谷 虎松, “拡張現実感技術によるナビゲーション支援のための AR オブジェクト共有機構の試作”, 人工知能と知識処理研究会 (AI), pp.30-35, (2020).
- [2] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm”, Proc. Int’l Conf. 3D Digital Imaging and Modeling, pp.145-152, (2001).
- [3] P. Li, et al, “Evaluation of the ICP Algorithm in 3D Point Cloud Registration”, IEEE Access, pp.68030 - 68048, (2020).
- [4] Kouta Yamada, et al, “A performance evaluation of keypoints detection methods SIFT and AKAZE for 3D reconstruction”, IWAIT 2018, pp.1-4, (2018).