画像修復を活用した点群内のオブジェクト除去手法の提案と評価

日本大学文理学部情報科学科　卒業演習発表会（令和6年2月7日）

谷聖一研究室　東智輝

目次

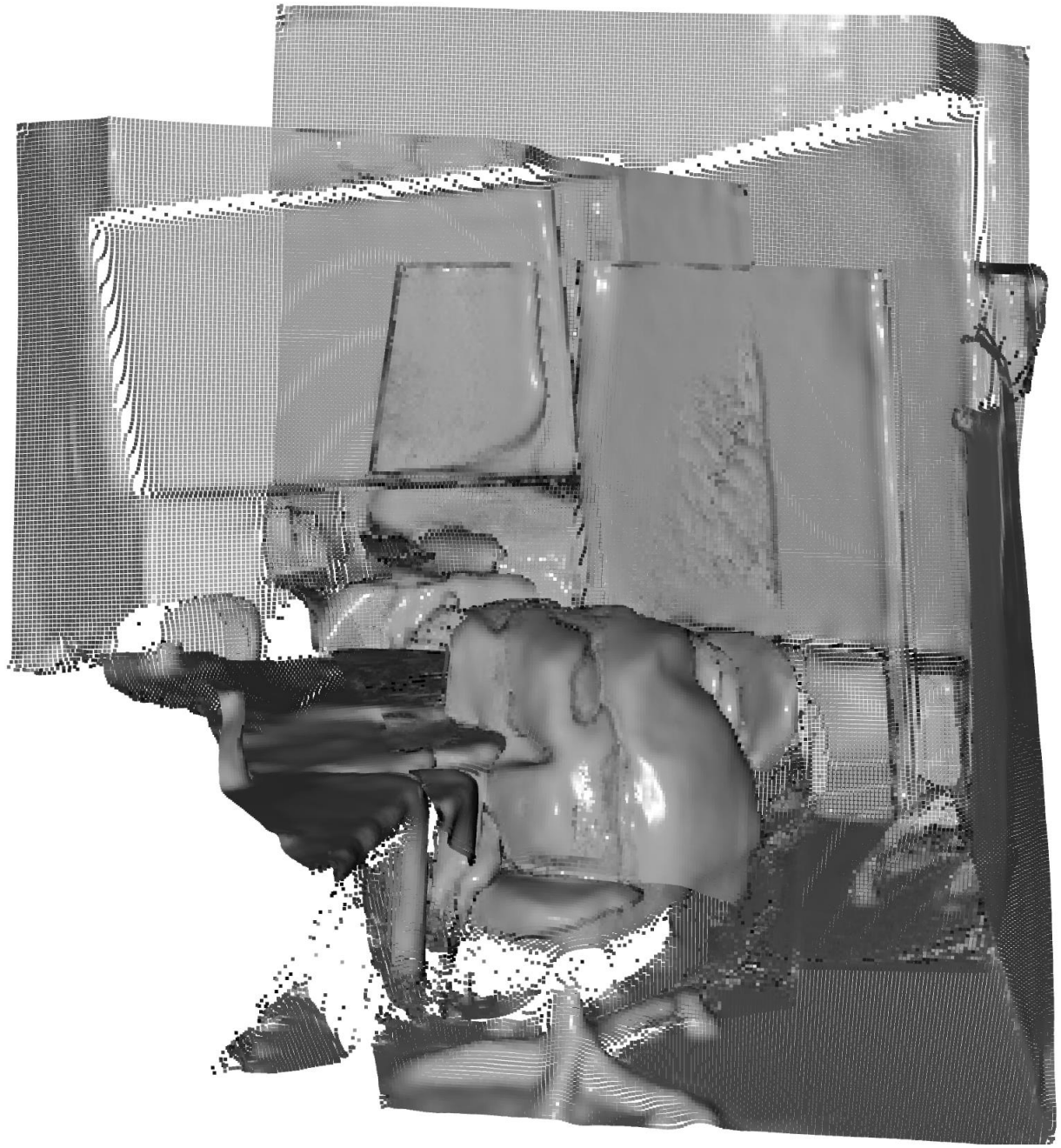
目次

1.1 点群とは

三次元空間内の点の集合

各点の情報

- **三次元座標値(X,Y,Z)**

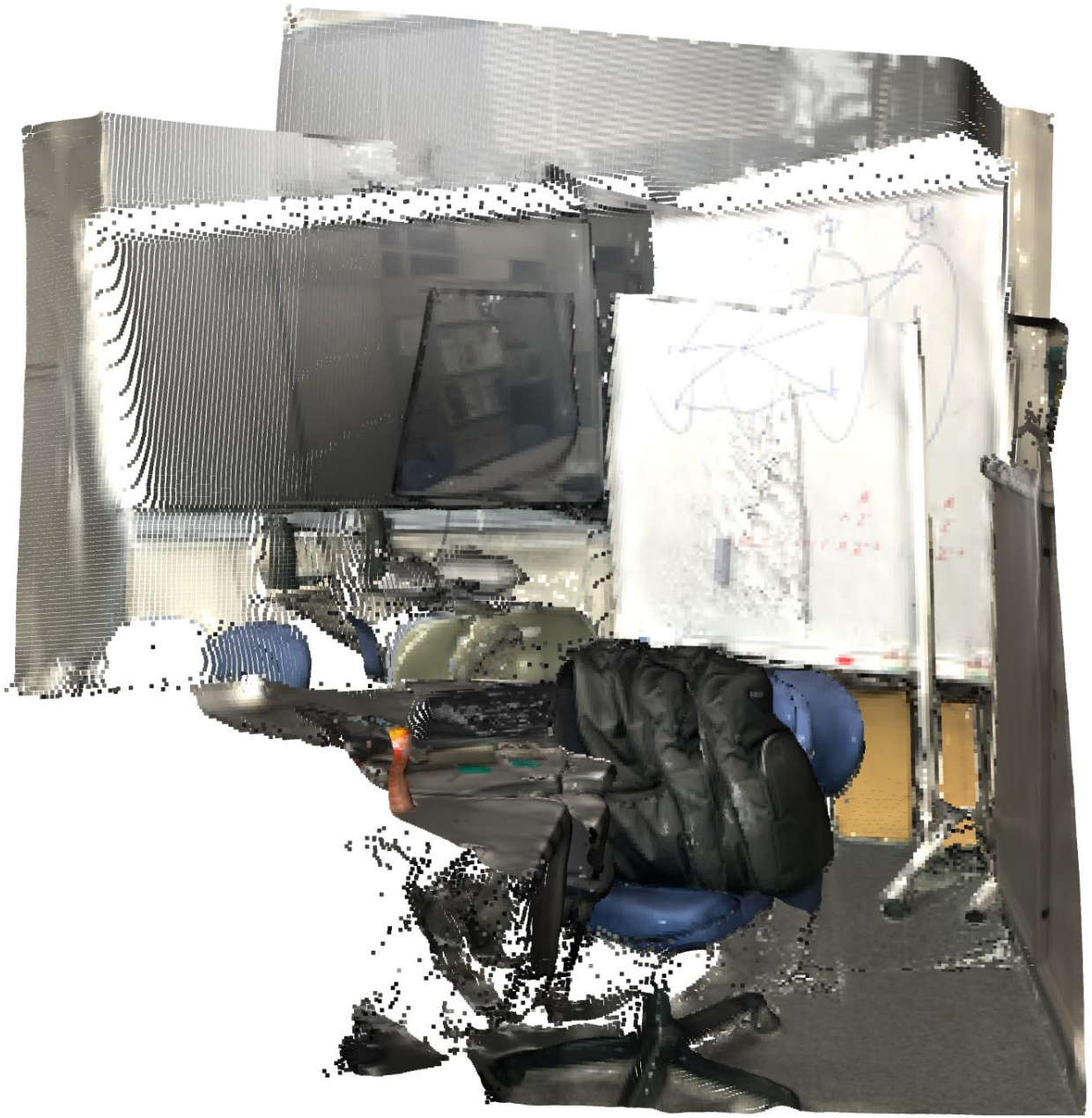


1.1 点群とは

三次元空間内の点の集合

各点の情報

- **三次元座標値(X,Y,Z)**
- **色情報（R,G,B）**



1.2 点群データの活用例

点群データはどのような事に活用されているのか？

- 車の自動運転
- デジタルツイン

1.2 点群データの活用例 - 車の自動運転

LiDARを搭載することで外部の情報を立体的に取得する
セグメンテーション・物体検出・物体認識などのタスクに活用

1.2 点群データの活用例 - デジタルツイン

現実空間と対になる双子（ツイン）をデジタル空間上に構築し、
現実空間のモニタリングやシミュレーションを可能にする仕組み

VIRTUAL SHIZUOKA（バーチャル静岡）
静岡県をまるごと点群データ化するプロジェクト
LiDARなどのセンサを用いて都市を計測し点群化する

画像引用:
https://3dview.tokyo-digitaltwin.metro.tokyo.lg.jp/#share=s-mVduTQw8aGtl08ZJ

1.3 演習概要

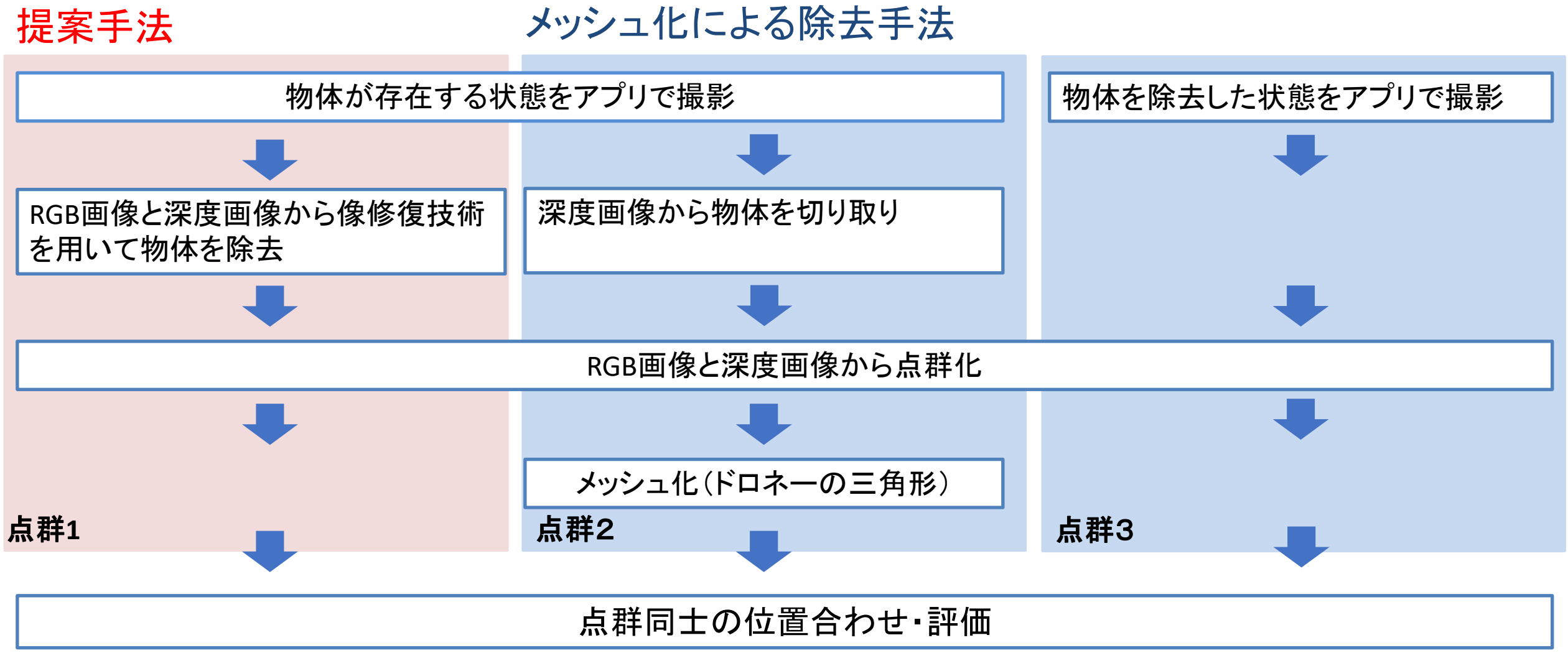
三次元点群内のオブジェクト除去手法を提案

→評価実験を通して提案手法を評価

提案手法

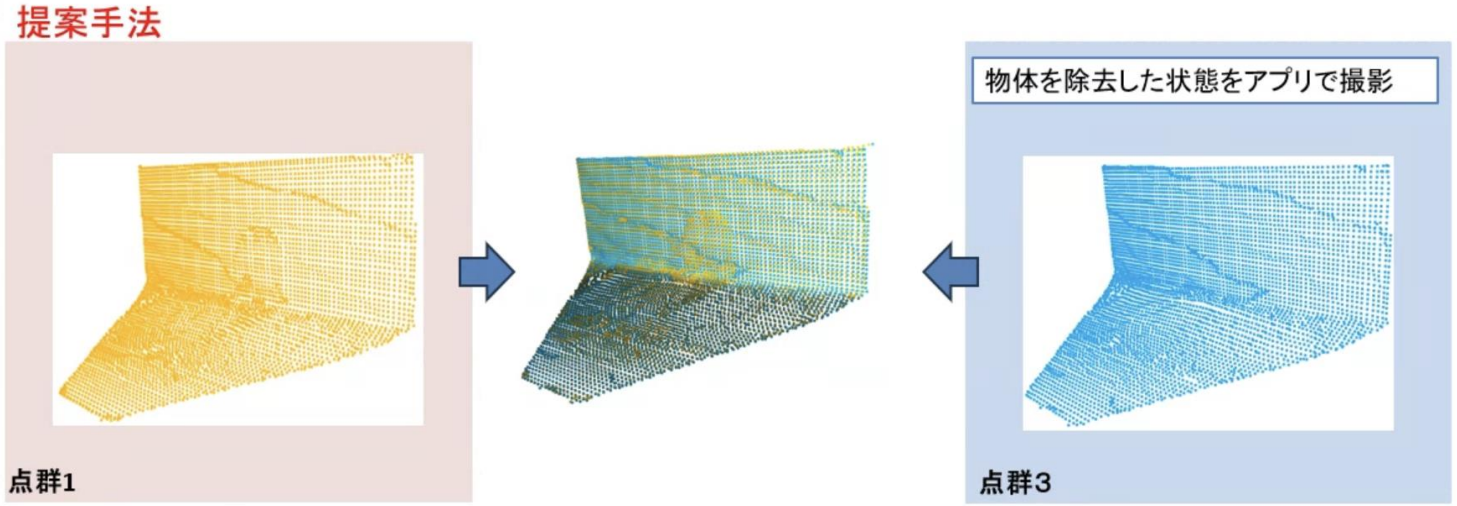
画像修復を深度画像に適用することで点群内のオブジェクトを除去する

1.4 演習概要

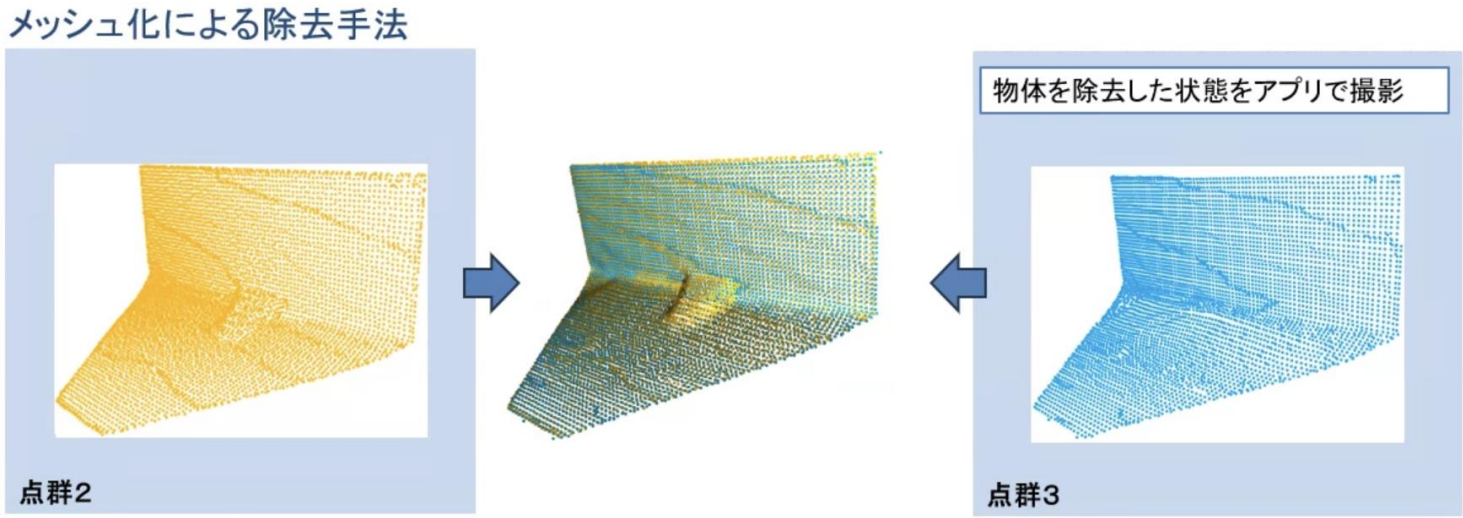

提案手法
メッシュ化による除去手法
物体が存在する状態をアプリで撮影
物体を除去した状態をアプリで撮影
RGB画像と深度画像から像修復技術を用いて物体を除去
深度画像から物体を切り取り
RGB画像と深度画像から点群化
メッシュ化（ドロネーの三角形）
点群1
点群2
点群3
点群同士の位置合わせ・評価

1.4 演習概要

位置合わせ1

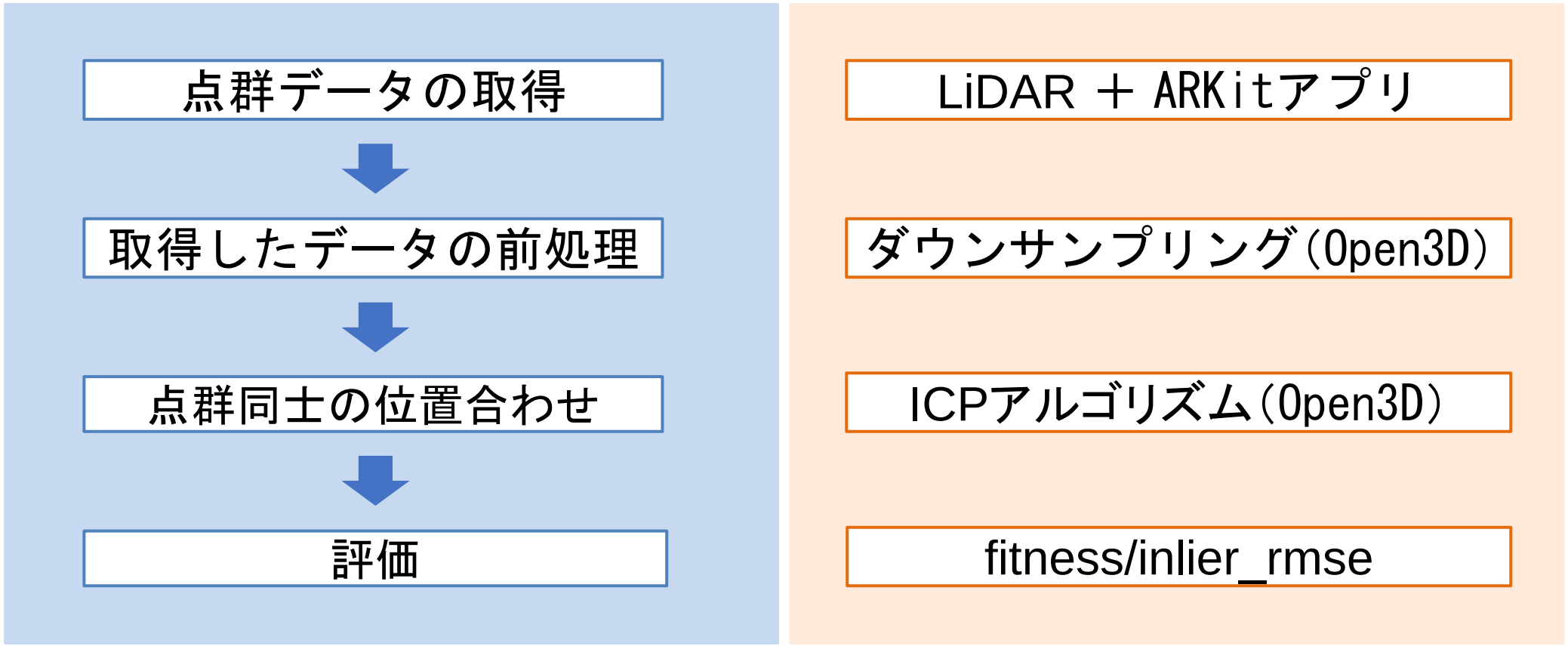


位置合わせ2

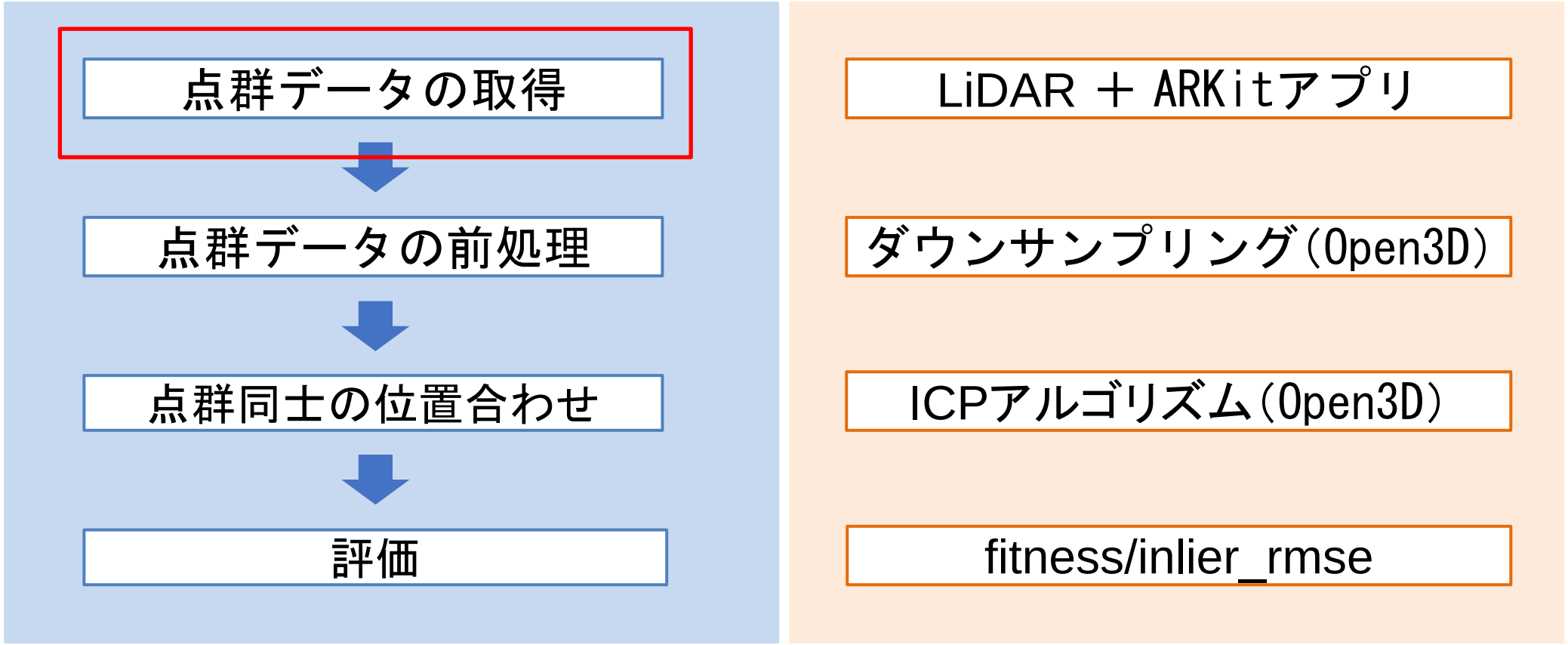


目次

2.1 点群を扱う流れ


点群データの取得
取得したデータの前処理
点群同士の位置合わせ
評価
LiDAR ＋ ARKitアプリ
ダウンサンプリング(Open3D)
ICPアルゴリズム(Open3D)
fitness/inlier_rmse

2.1 点群を扱う流れ


点群データの取得
点群データの前処理
点群同士の位置合わせ
評価
LiDAR ＋ ARKitアプリ
ダウンサンプリング(Open3D)
ICPアルゴリズム(Open3D)
fitness/inlier_rmse

2.2 点群データの取得

本演習では現実世界のオブジェクトを点群化する

三次元計測装置を用いて測量
測量したデータをもとに各点のx,y,z座標を求める

三次元計測装置
装置から対象物までの距離[m]を測定できる
本演習では光学式の装置を使用

2.2 点群データの取得

光学式の三次元計測装置には主に2種類の計測方式がある

- **ステレオ方式**
  2つ以上のカメラを搭載し，視差を用いて奥行きを計算する
- **LiDAR**
  光の速さを利用した計測方式

  ToF(Time of Flight)
  主に不可視光線を照射し，反射光が返ってくるまでの時間を計測
  往復にかかった時間をもとに奥行きを計算

2.2 点群データの取得

光学式三次元計測装置

川口演習で使用

RealSense D435(Intel)

ステレオ方式

画像引用:https://www.intelrealsense.com/depth-camera-d435/

本演習で使用

iPad Pro(Apple)

LiDAR（ToF）

画像引用:https://www.apple.com/jp/ipad-pro/

2.2 点群データの取得

本演習ではARKitを用いて開発したアプリを使用

→シャッターボタン押下時に同一画角のRGB画像と深度画像を保存

2.2 点群データの取得

ARKit

iPhone/iPadにてAR体験を容易に構築できるフレームワーク
ARKitを用いることでLiDARセンサから深度情報を取得できる

画像引用: https://developer.apple.com/jp/augmented-reality/arkit/

2.2 点群データの取得

本演習ではARKitを用いて開発したアプリを使用

→シャッターボタン押下時に同一画角のRGB画像と深度画像を保存

2.2 点群データの取得

深度画像

グレースケールで対象物との距離を表現した画像

本演習では8bit（0～255）でグレースケールを表現する

2.2 点群データの取得

深度画像から距離の復元

G : ピクセル画素値
距離[m] = G × depth_scale

本演習の設定 → depth_scale=0.01

（例）
ピクセルの画素値 : 255
255*0.01 = 2.55[m]

2.2 点群データの取得

RGB画像と深度画像を用いた点群作成方法

- 同一画角のRGB画像と深度画像を用意
- カメラの主点と焦点を取得
- 主点と焦点から各ピクセルの3次元座標を計算

2.2 点群データの取得

RGB画像と深度画像を用いた点群作成方法

- 同一画角のRGB画像と深度画像を用意　アプリで取得
- カメラの主点と焦点を取得
- 主点と焦点から各ピクセルの3次元座標を計算

2.2 点群データの取得

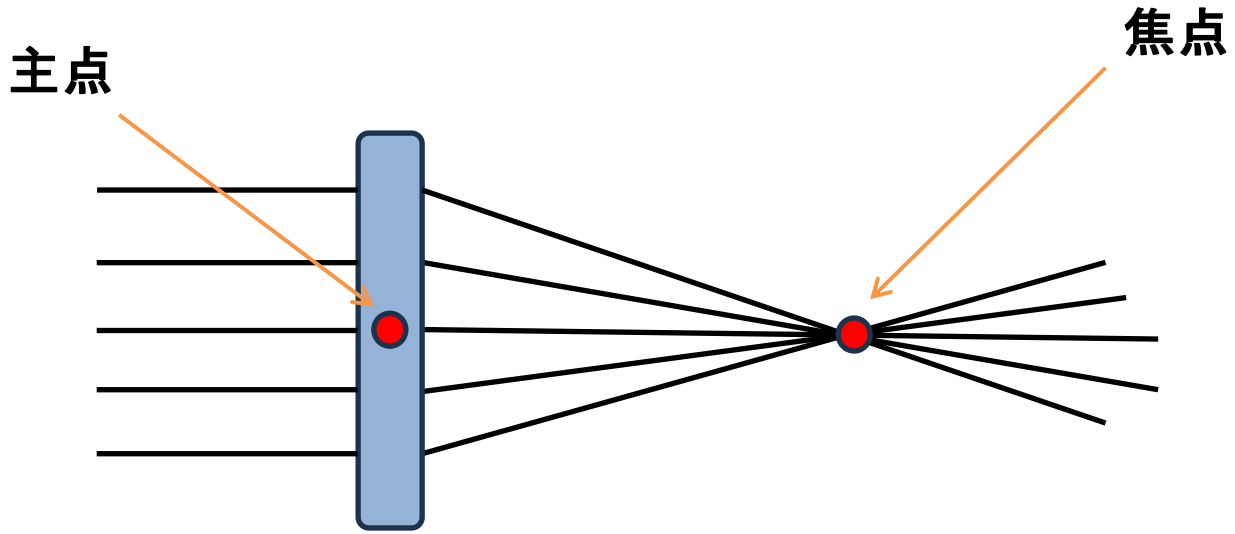
RGB画像と深度画像を用いた点群作成方法

- 同一画角のRGB画像と深度画像を用意
- カメラの主点と焦点を取得
- 主点と焦点から各ピクセルの3次元座標を計算

2.2 点群データの取得

カメラの主点と焦点を取得

主点	焦点
レンズの中心点 主点から焦点までの距離を焦点距離と呼ぶ	レンズに入った光が集まる点



2.2 点群データの取得

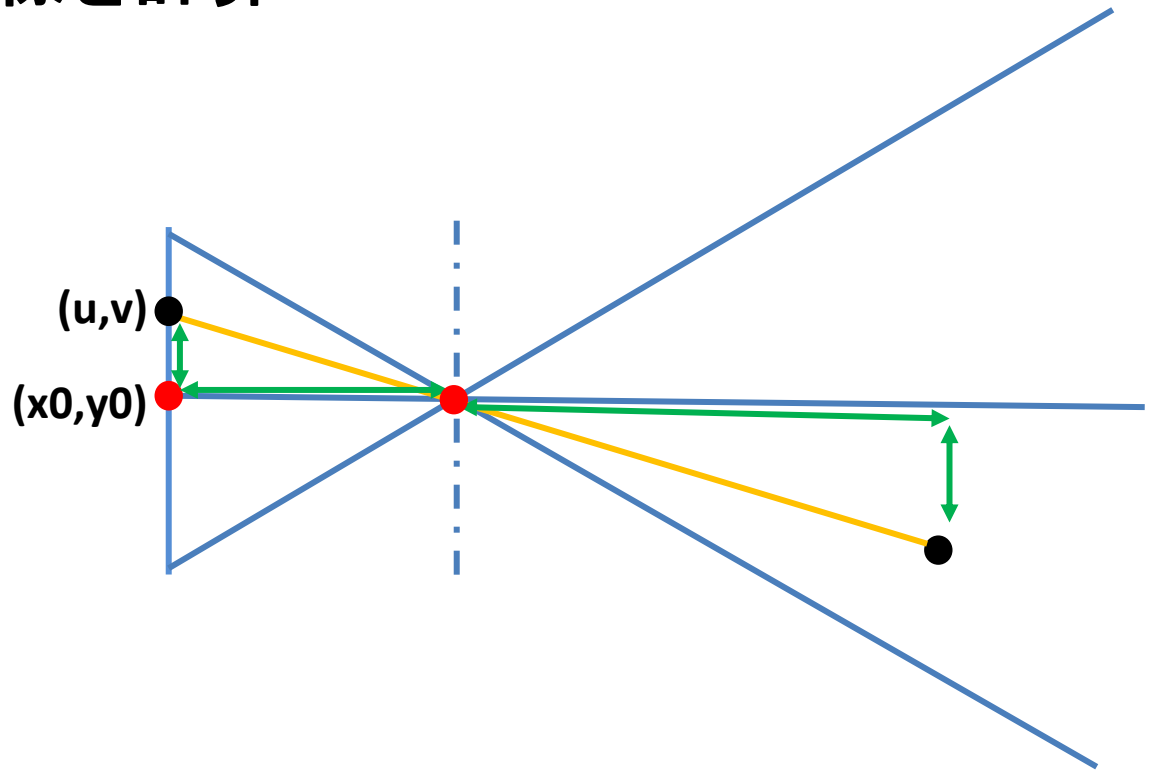
RGB画像と深度画像を用いた点群作成方法

- 同一画角のRGB画像と深度画像を用意
- カメラの主点と焦点を取得
- 主点と焦点から各ピクセルの3次元座標を計算

2.2 点群データの取得

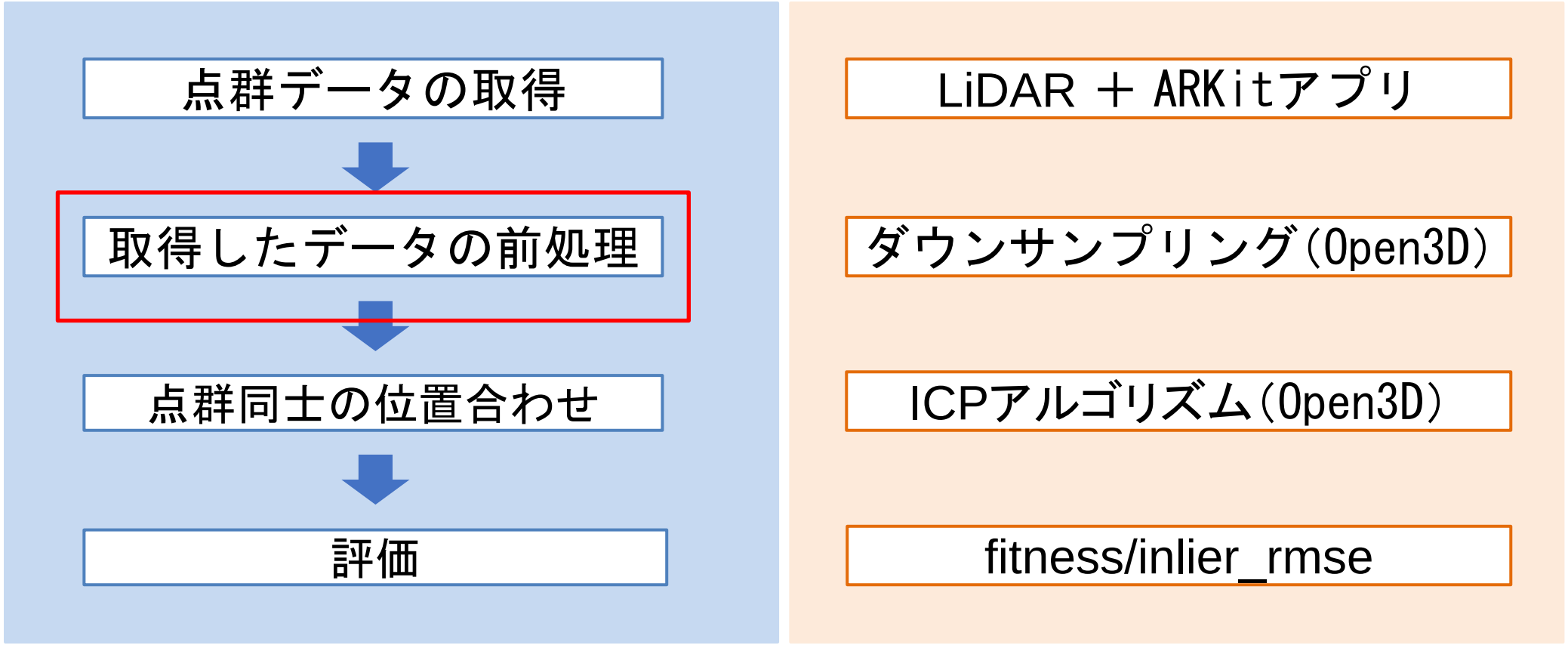
主点と焦点から各ピクセルの3次元座標を計算

fx,fy = 焦点距離
x0,y0 = 主点
x = (u-x0)*z/fx　　y=(v-y0)*z/fy



各ピクセルのx,y,z座標が求まる
→点群データの完成

2.1 点群を扱う流れ


点群データの取得
取得したデータの前処理
点群同士の位置合わせ
評価
LiDAR ＋ ARKitアプリ
ダウンサンプリング(Open3D)
ICPアルゴリズム(Open3D)
fitness/inlier_rmse

2.3 点群データの前処理

Open3D

3Dデータを扱うソフトウェアの開発をサポートするためのオープンソースライブラリ

C++ と Python で利用可能

本演習では点群データの前処理や点群の位置合わせに利用

画像引用:https://www.open3d.org/

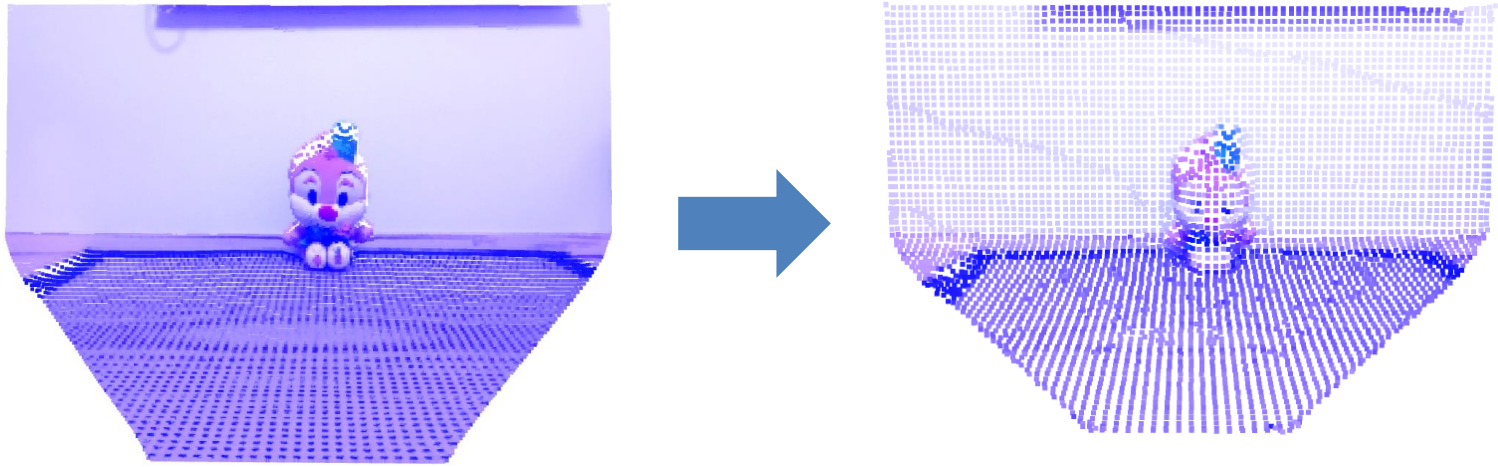
2.3 点群データの前処理

ダウンサンプリング

点群内の点数を減らす処理

- 点密度の均一化
- ICPアルゴリズムの処理時間減少

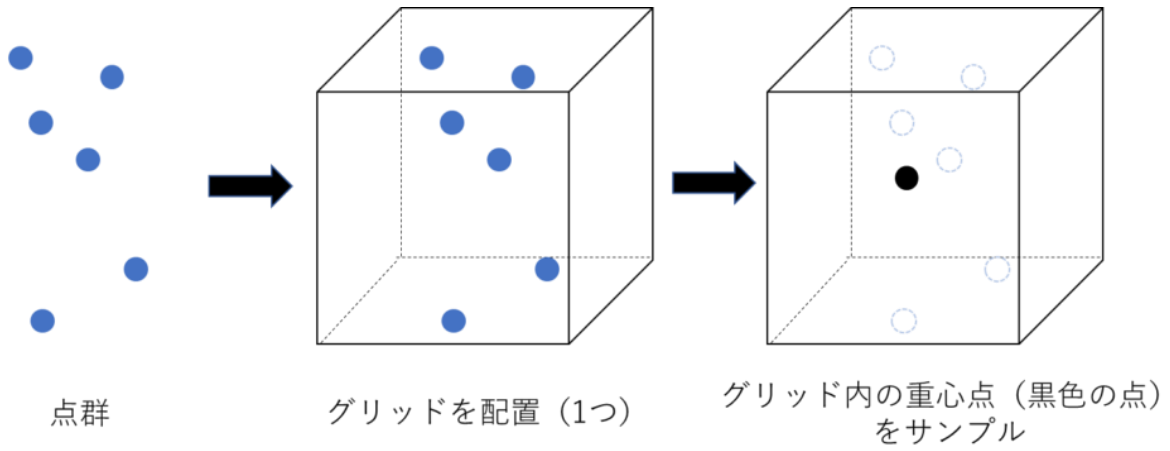
本演習では前処理としてボクセルダウンサンプリングを行う



2.3 点群データの前処理

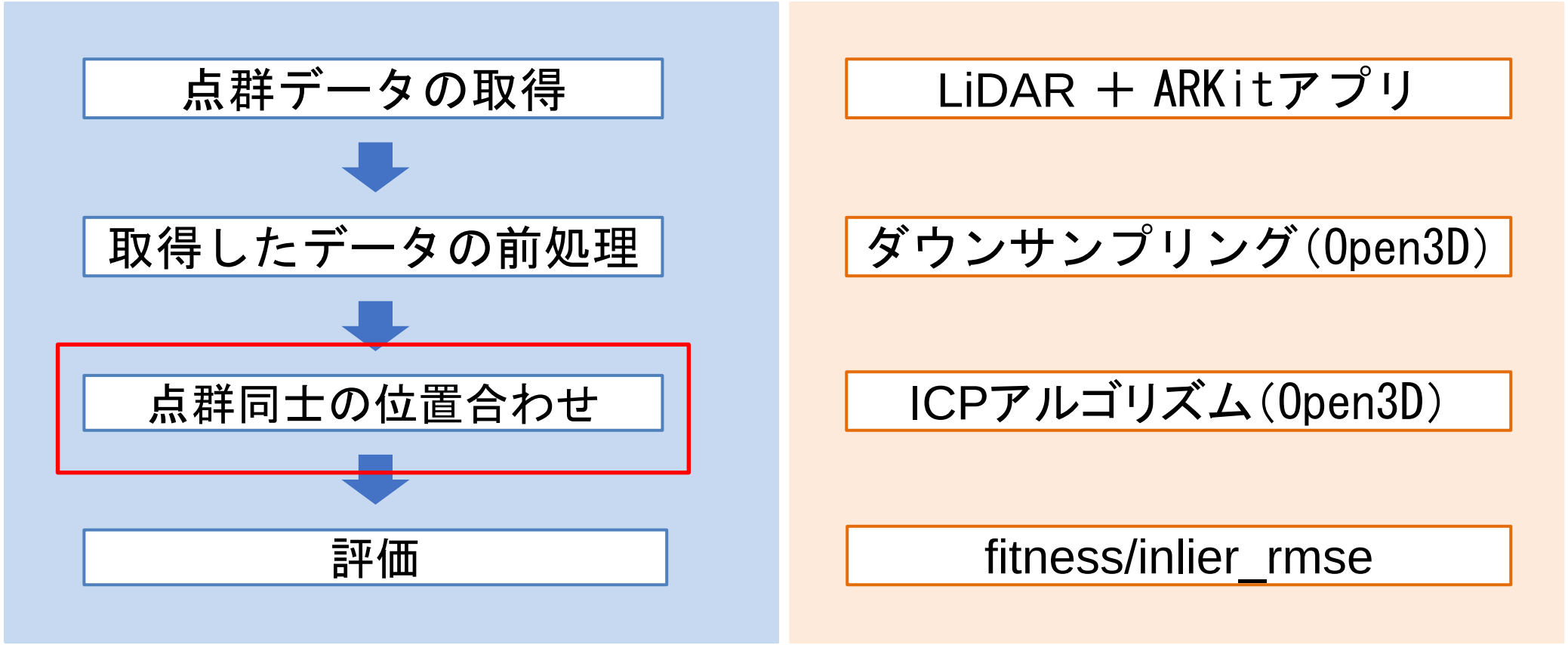
ボクセルダウンサンプリング

1. 全ての点群データを包含する直方体を生成
2. パラメータsを一辺の長さとするボクセルで直方体を分割
3. 各ボクセル内の点の座標の平均値を求め，0 or 1個の点を生成



画像引用：https://tech-deliberate-jiro.com/downsampling-grid/

2.1 点群を扱う流れ


点群データの取得
取得したデータの前処理
点群同士の位置合わせ
評価
LiDAR ＋ ARKitアプリ
ダウンサンプリング(Open3D)
ICPアルゴリズム(Open3D)
fitness/inlier_rmse

2.4 点群同士の位置合わせ

複数の点群データを統合する処理

取得した点群データ（異なる位置から撮影した3次元点群同士）
の位置関係を推定し点群を統合

本演習では位置合わせにICPアルゴリズムを使用

2.4 点群同士の位置合わせ

ICP（Iterative Closest Points）

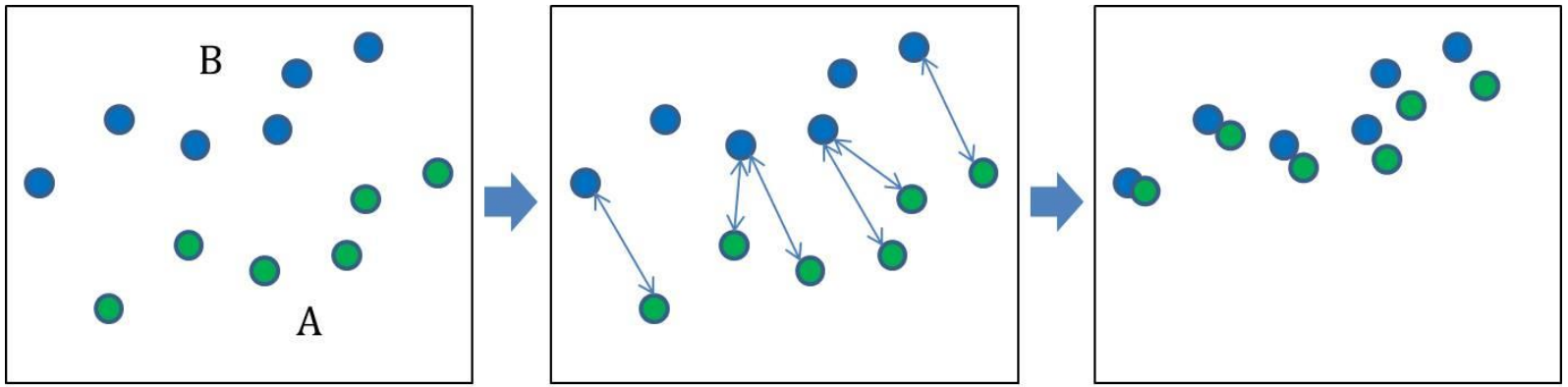
ICPアルゴリズムは点群同士の位置合わせ処理に広く用いられている

入力：ソース点群とターゲット点群
出力：ソース点群をターゲット点群に位置合わせするために必要な回転と推進の推定値

2.4 点群同士の位置合わせ

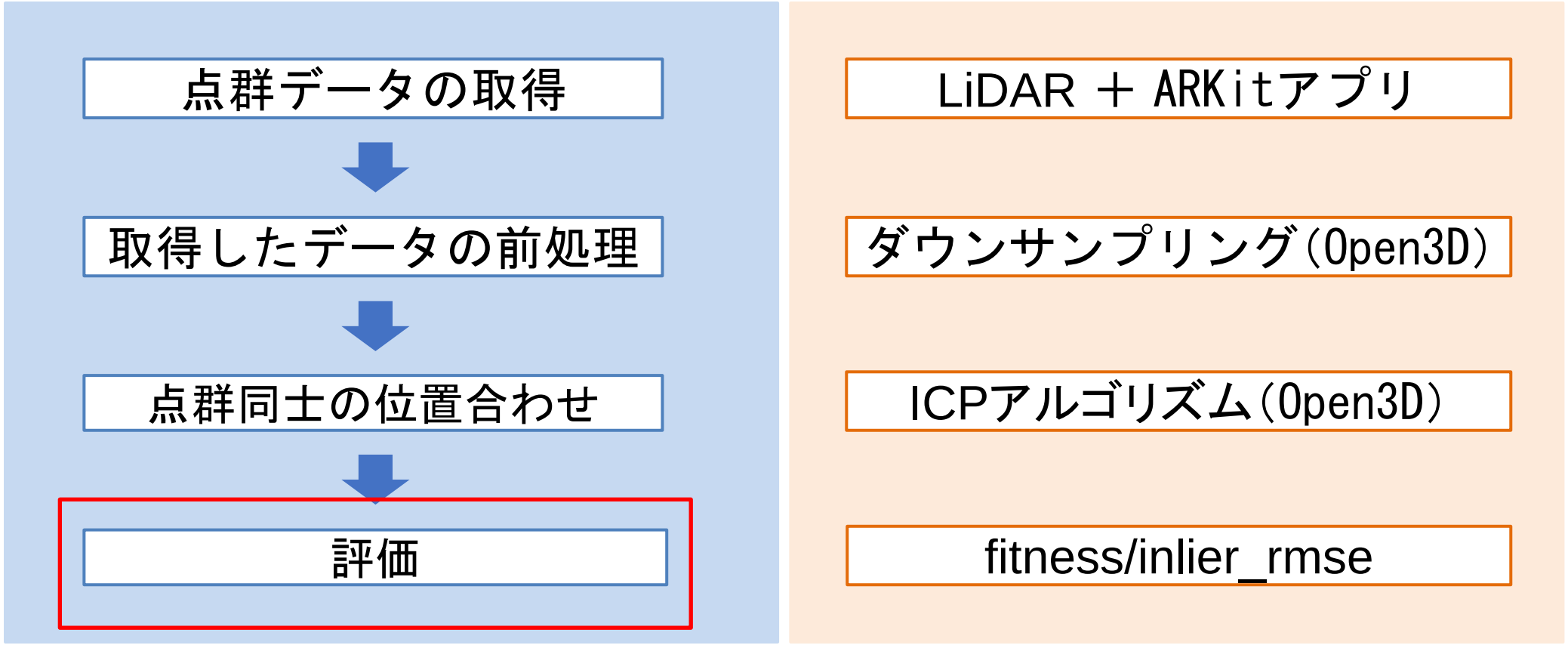
アルゴリズムの手順

1. ソース点群の各点からターゲット点群内の最近傍の点を対応付け
2. 対応付けた点同士の距離が最小になるようにターゲット点群を回転＆推進
3. 対応付けた点同士の二乗距離誤差を計算
4. ひとつ前の誤差と今回の誤差の差が閾値以上であれば手順1に戻る



画像引用：http://www.sanko-shoko.net/note.php?id=3c5m

2.1 点群を扱う流れ


点群データの取得
取得したデータの前処理
点群同士の位置合わせ
評価
LiDAR ＋ ARKitアプリ
ダウンサンプリング(Open3D)
ICPアルゴリズム(Open3D)
fitness/inlier_rmse

2.5 評価

本演習では2つの評価指標を用いて位置合わせを評価

- **fitness**
  対応付けされた点数をターゲット点群内の点数で割ったもの
  値が大きいほど点群同士の重複領域が大きいことを示す
  →本演習では補完の精度を意味する

- **inlier_rmse**
  対応付けされた点同士のユークリッド距離の二乗平均平方根誤差
  →本演習ではマッチングの精度を意味する

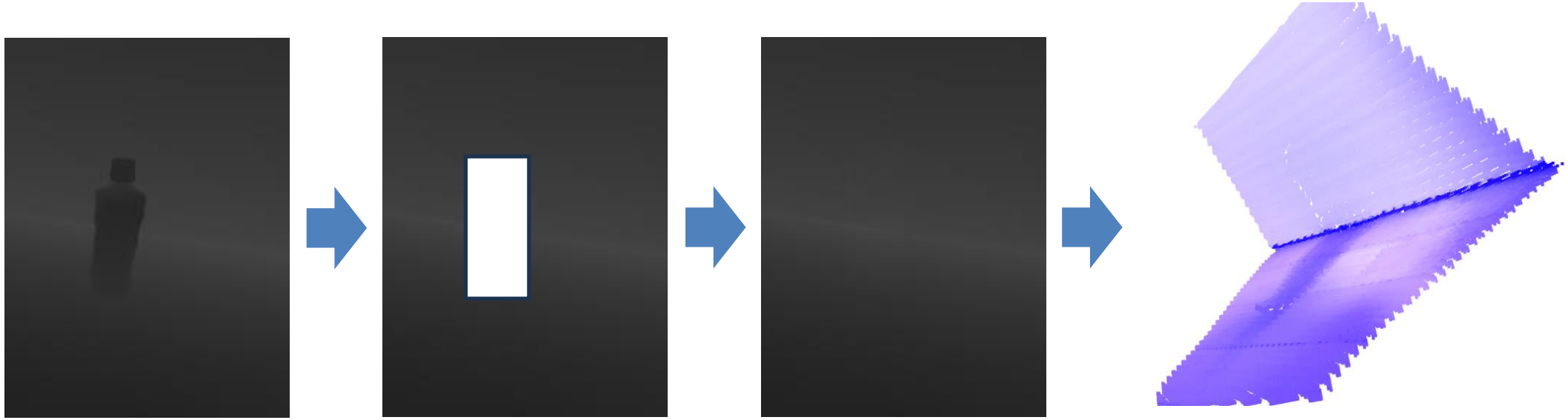
目次

3.1 提案手法

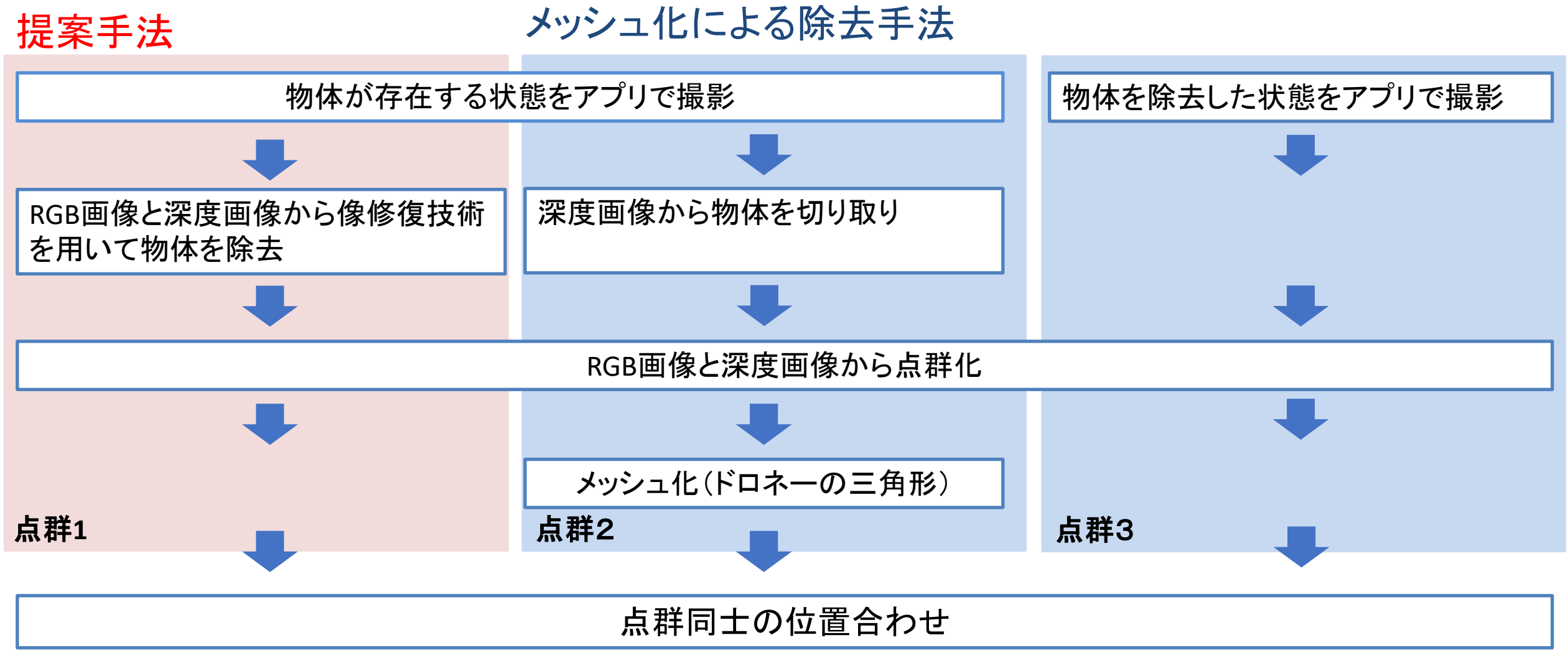
画像修復を活用した点群内の物体除去手法
→本演習では深度画像に画像修復を適用

手順

1. 深度画像内のオブジェクトを消去
2. 画像修復を用いて消去範囲を修復
3. 深度画像をもとに点群を作成



3.2 演習方法


提案手法
メッシュ化による除去手法
物体が存在する状態をアプリで撮影
物体を除去した状態をアプリで撮影
RGB画像と深度画像から像修復技術を用いて物体を除去
深度画像から物体を切り取り
RGB画像と深度画像から点群化
メッシュ化（ドロネーの三角形）
点群1
点群2
点群3
点群同士の位置合わせ

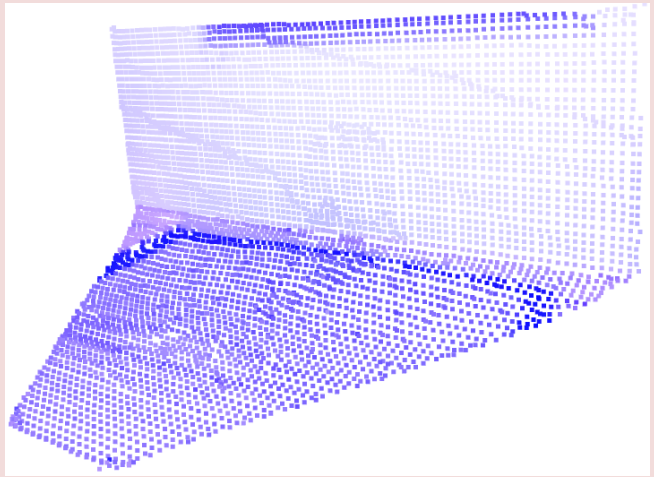
3.2 演習方法

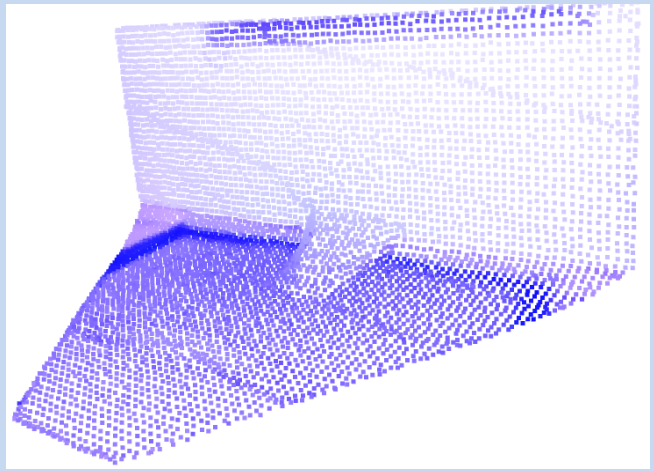
提案手法

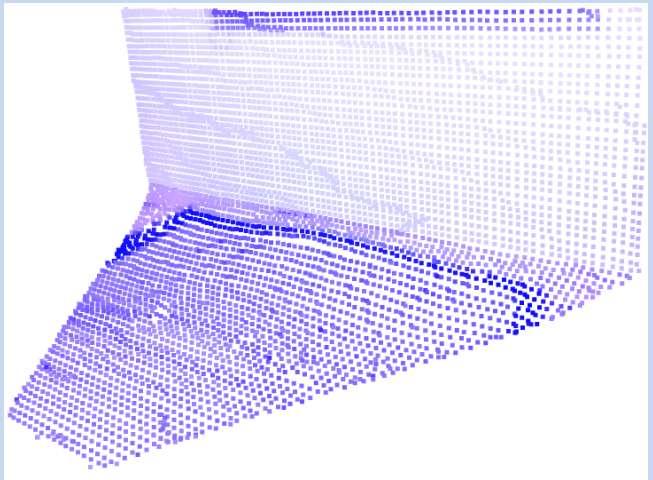
メッシュ化による除去手法

物体が存在する状態をアプリで撮影

物体を除去した状態をアプリで撮影







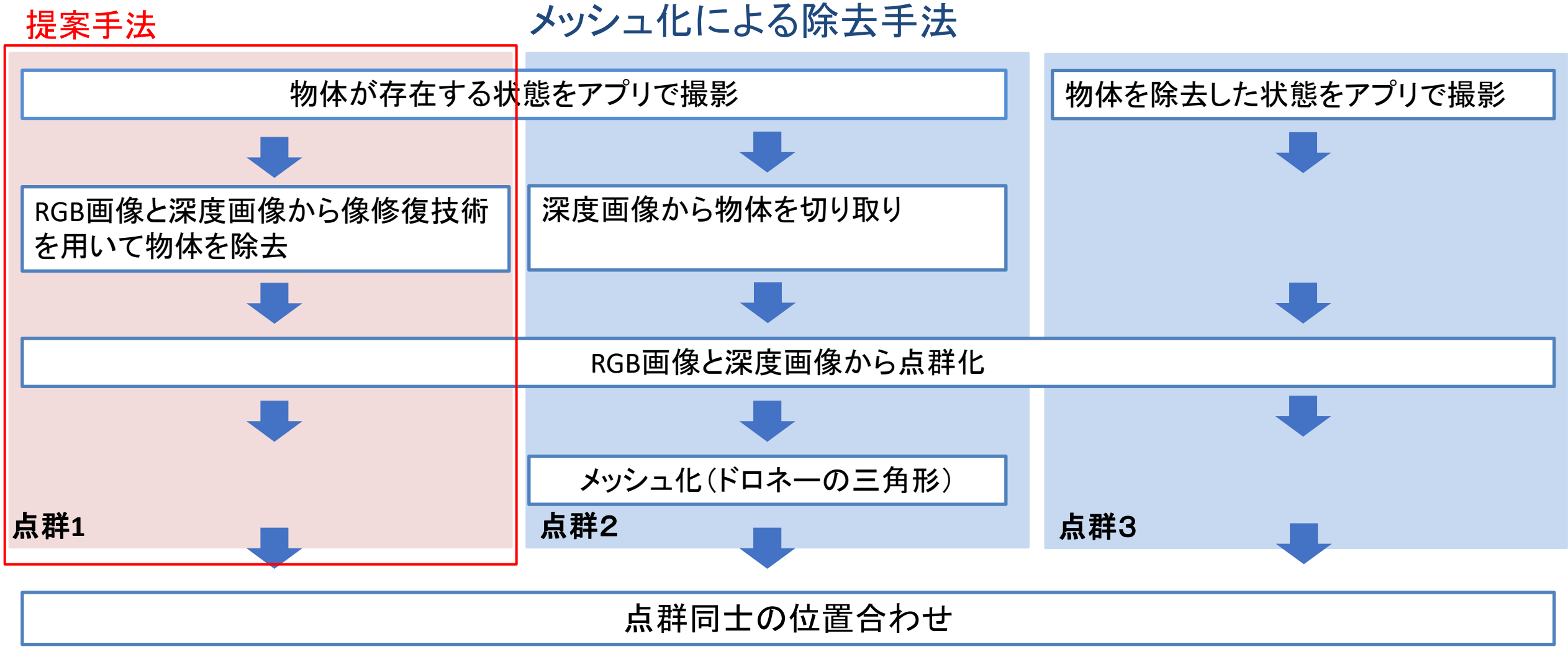
点群1

点群2

点群3

点群同士の位置合わせ

3.2 演習方法


提案手法
メッシュ化による除去手法
物体が存在する状態をアプリで撮影
物体を除去した状態をアプリで撮影
RGB画像と深度画像から像修復技術を用いて物体を除去
深度画像から物体を切り取り
RGB画像と深度画像から点群化
メッシュ化（ドロネーの三角形）
点群1
点群2
点群3
点群同士の位置合わせ

撮影

↓

物体除去

↓

点群化

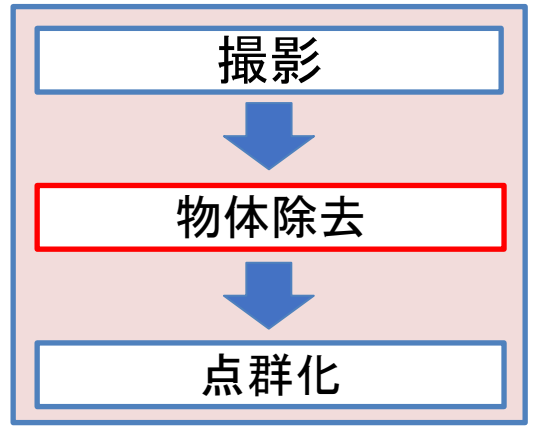
3.2 演習方法

物体が存在する状態をアプリで撮影

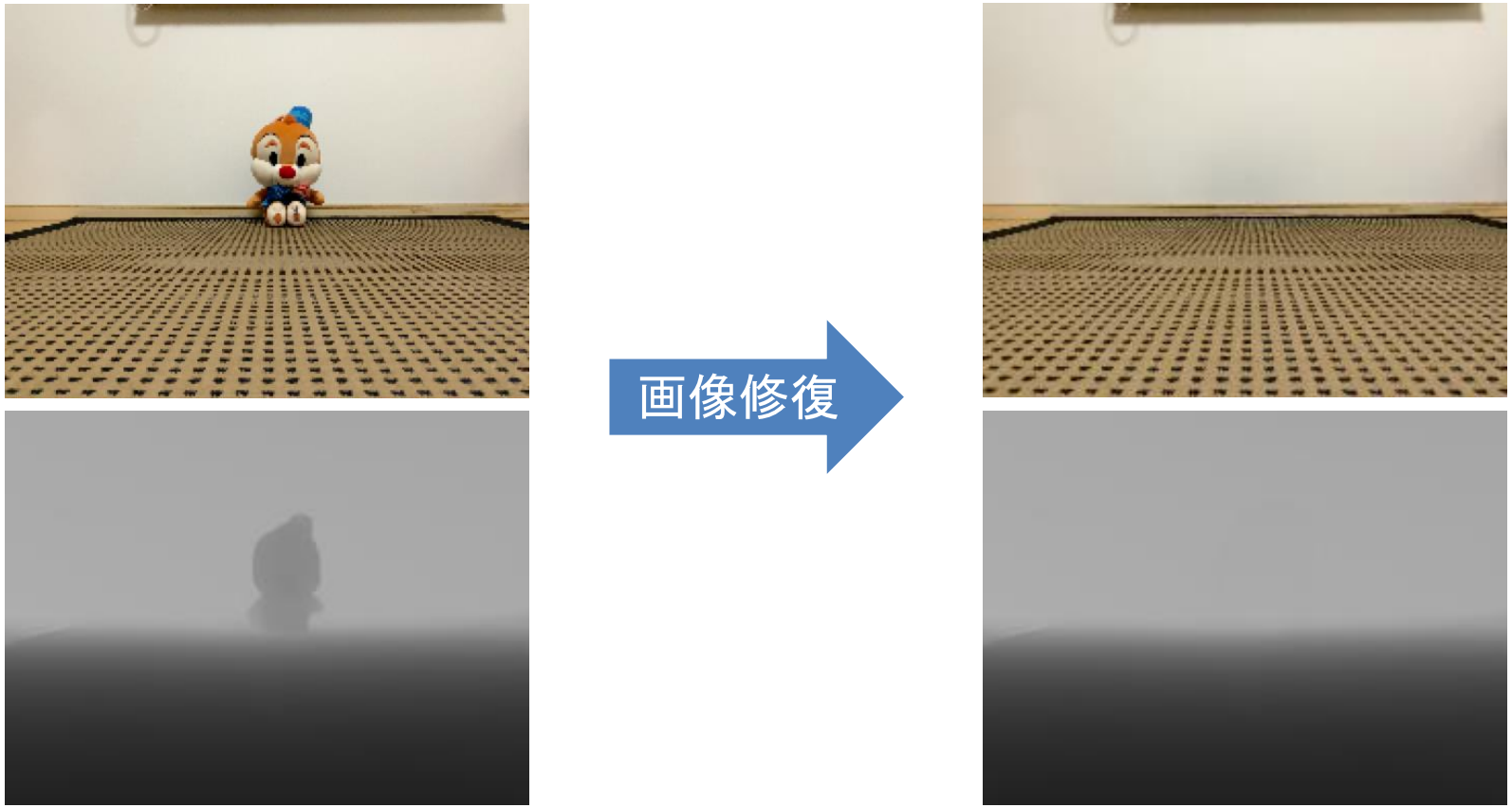
RGB画像

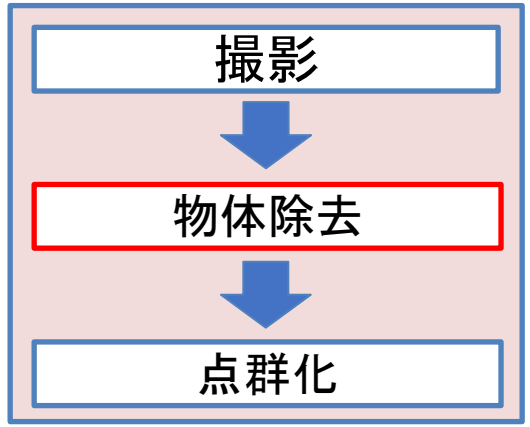
深度画像

3.2 演習方法



RGB画像と深度画像から画像修復を用いて物体を除去



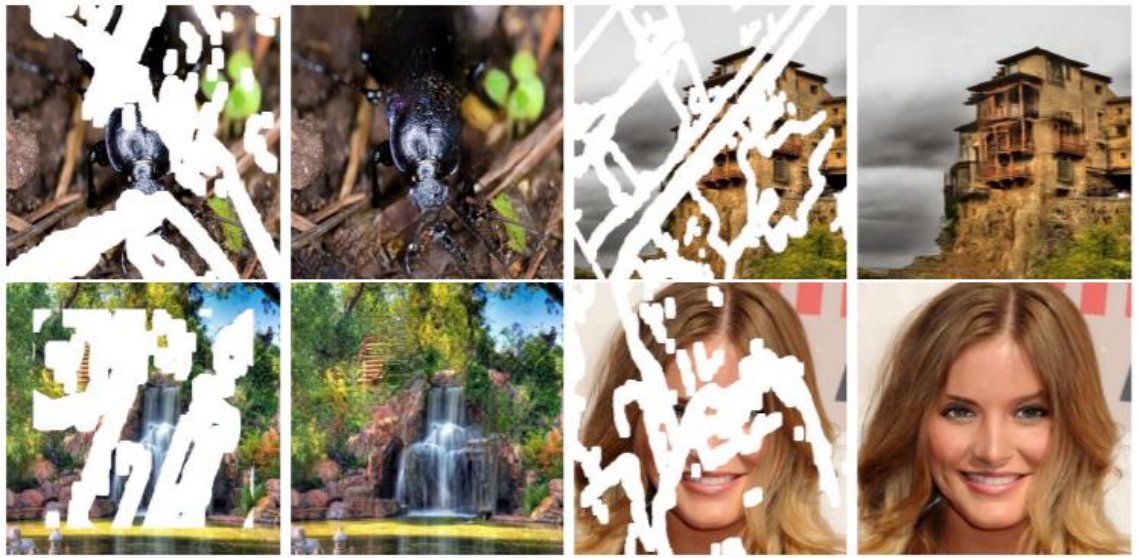


3.2 演習方法

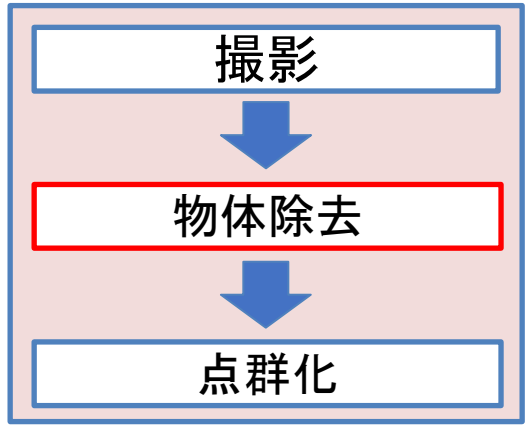
RGB画像と深度画像から像修復技術を用いて物体を除去

画像修復

画像修復（Image Inpainting）とは画像中の欠損を違和感なく直す技術
画像上のマスクされた領域を再構成



画像引用:https://qiita.com/jw-automation/items/0131b24a0b5a4e6fd0a6



3.2 演習方法

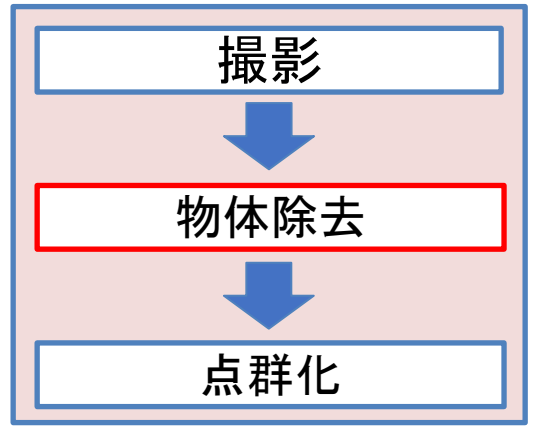
LaMa

機械学習を用いた画像修復手法
モスクワとサムスンAIチームが2021年9月に発表した

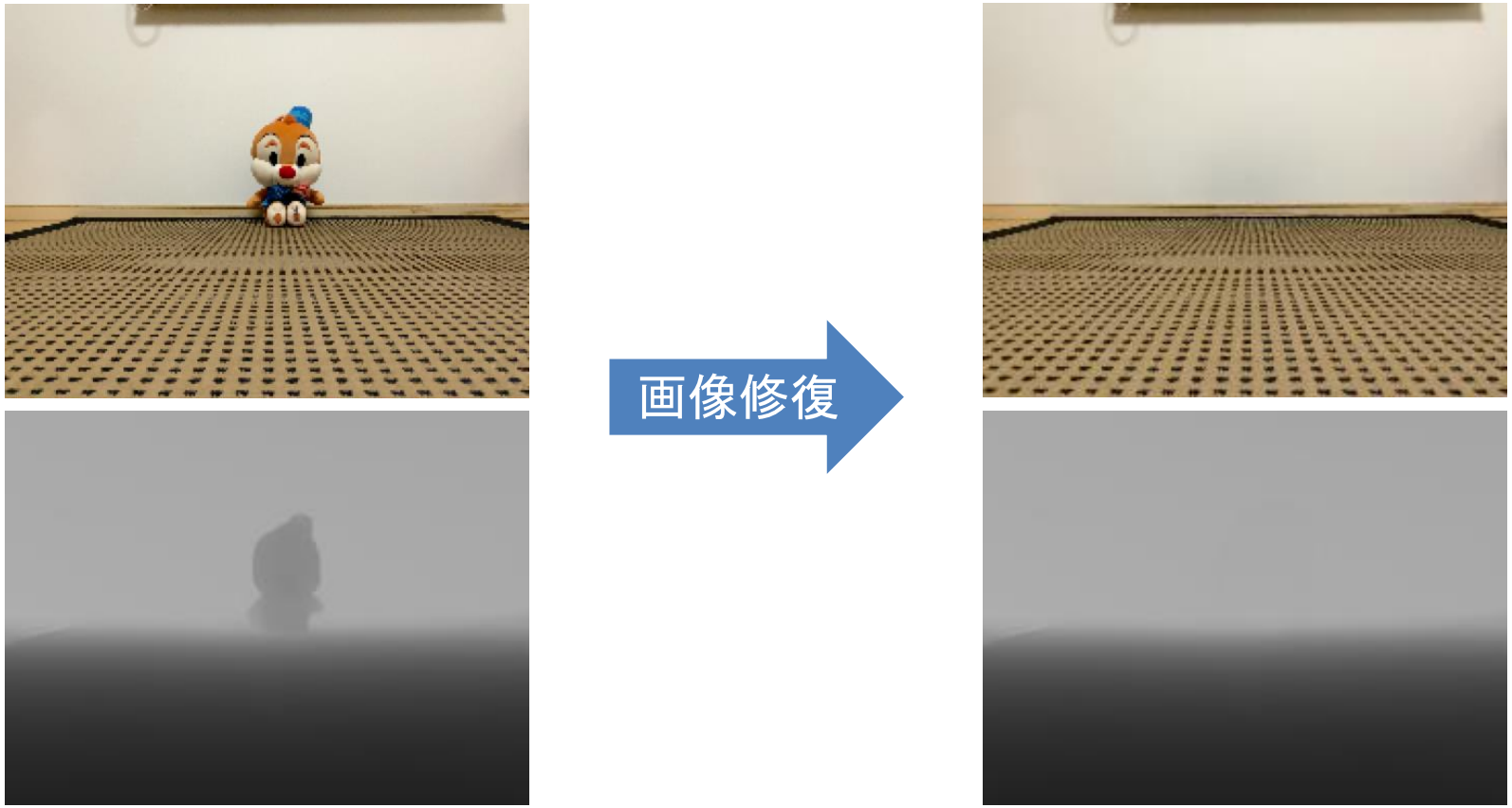
本演習ではLaMaを搭載した画像修復ツール**LaMaCleaner**を用いる

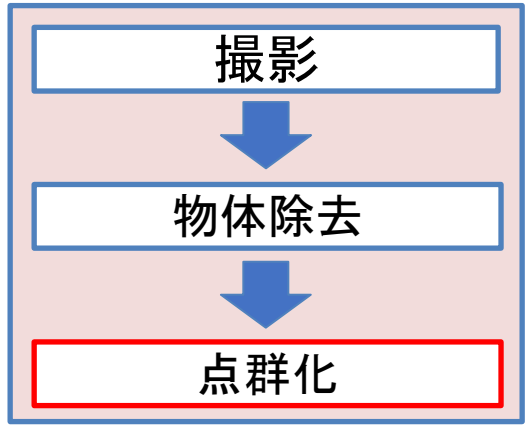
画像引用:https://github.com/advimman/lama

3.2 演習方法



RGB画像と深度画像から画像修復を用いて物体を除去



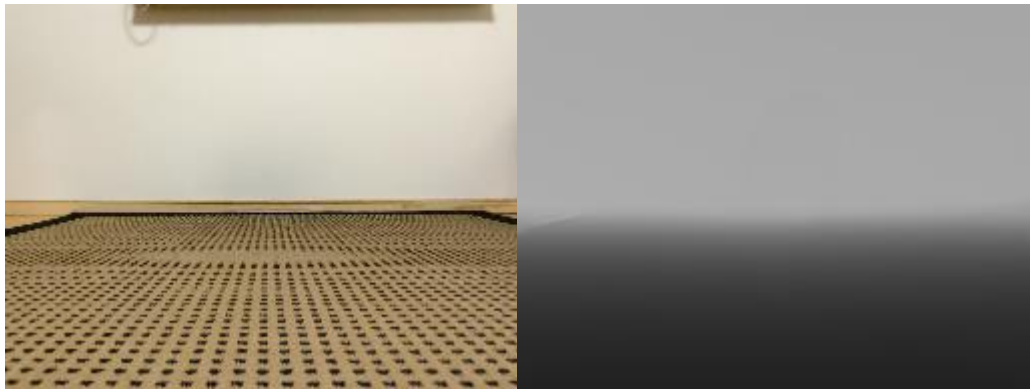


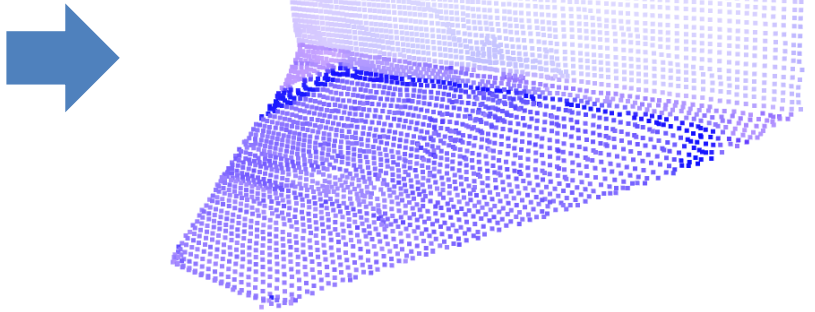
3.2 演習方法

RGB画像と深度画像から点群化

点群化にはOpen3D内のライブラリを用いる

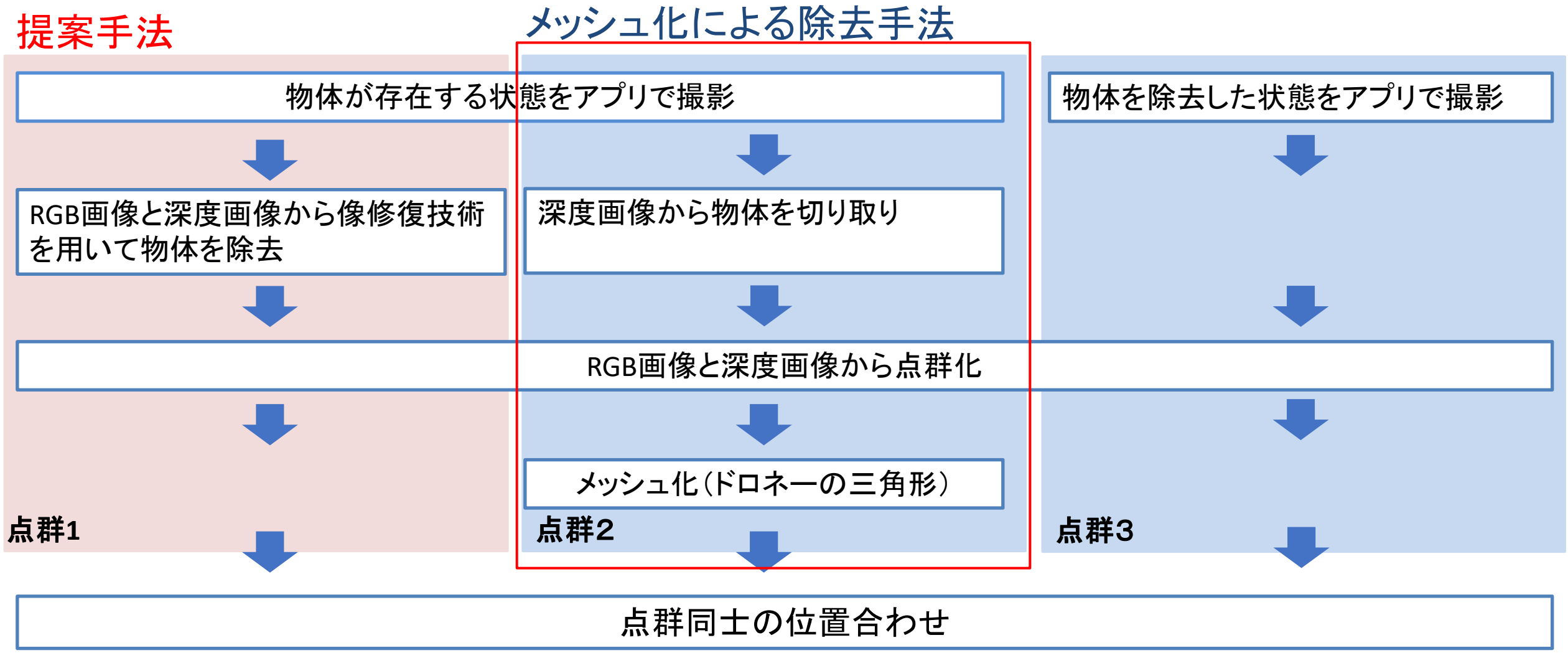
得られた点群を点群1とする





点群1

3.2 演習方法


提案手法
メッシュ化による除去手法
物体が存在する状態をアプリで撮影
物体を除去した状態をアプリで撮影
RGB画像と深度画像から像修復技術を用いて物体を除去
深度画像から物体を切り取り
RGB画像と深度画像から点群化
メッシュ化（ドロネーの三角形）
点群1
点群2
点群3
点群同士の位置合わせ

3.2 演習方法

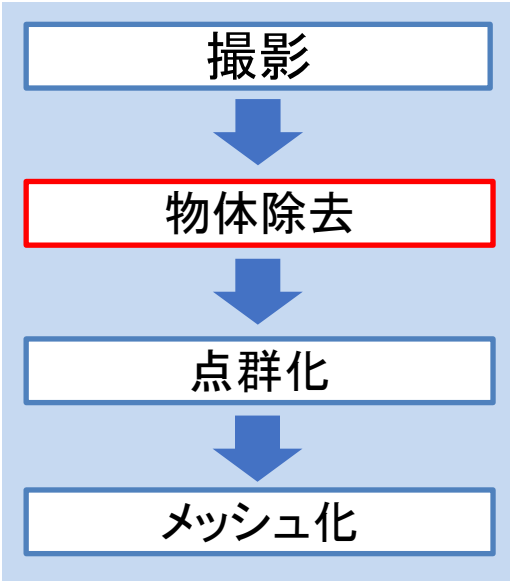
撮影
↓
物体除去
↓
点群化
↓
メッシュ化

物体が存在する状態をアプリで撮影

RGB画像

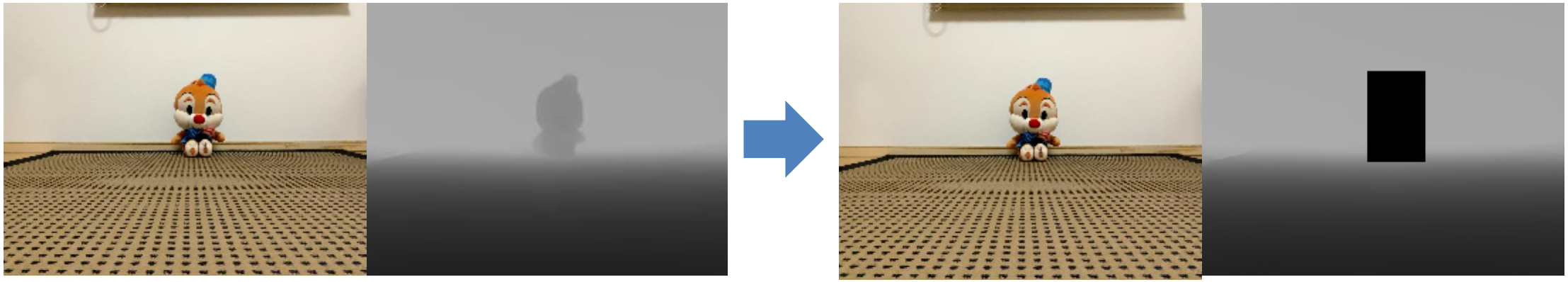
深度画像

3.2 演習方法

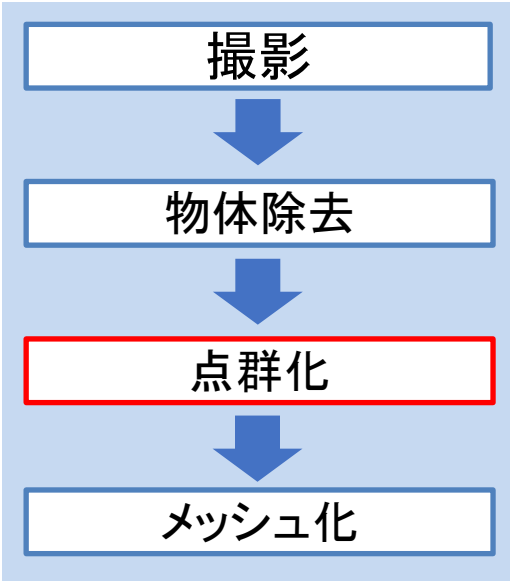


深度画像から物体を切り取り

深度画像内の物体領域を黒の直方体で隠す
→点群化した際に物体部分が除去される

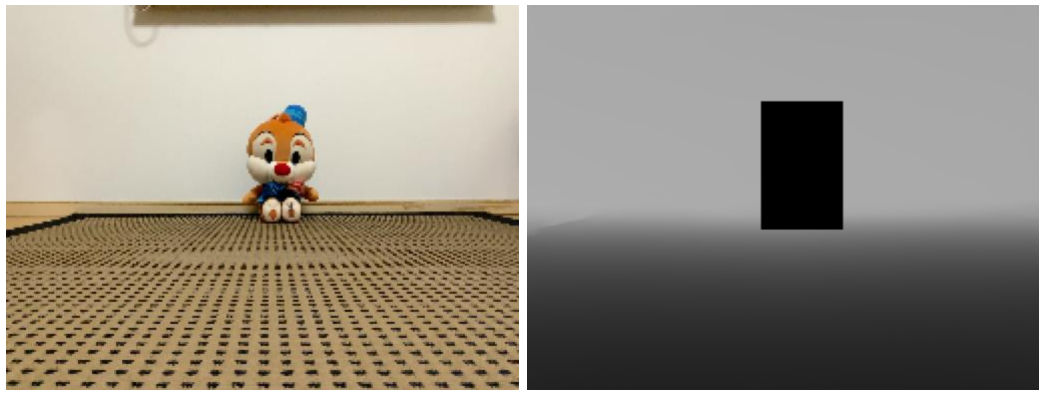


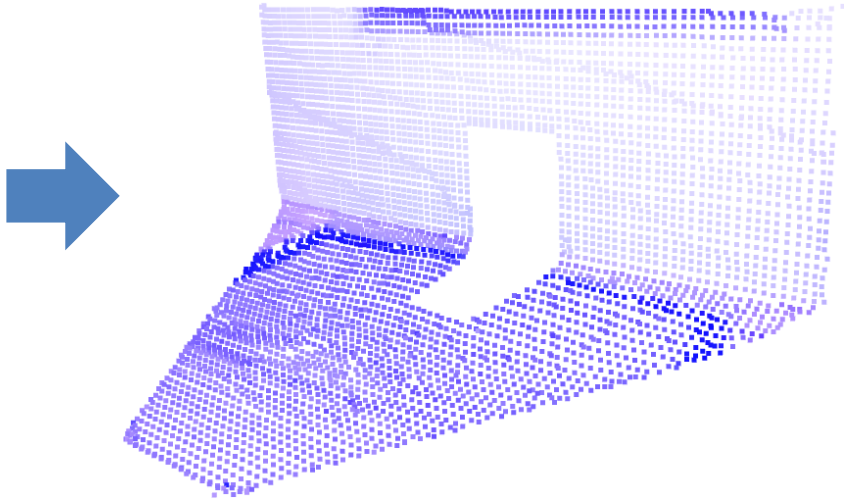
3.2 演習方法



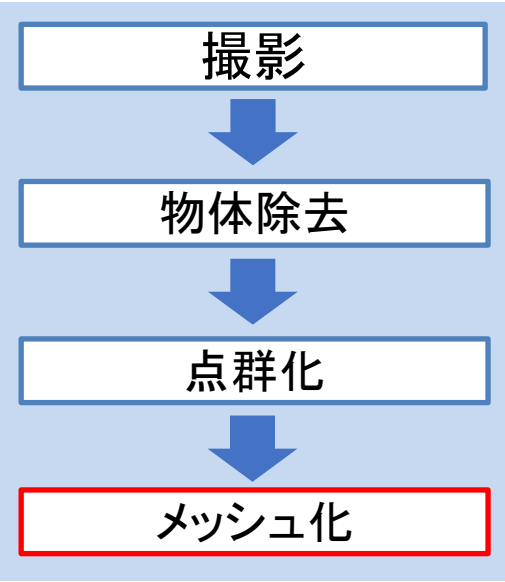
RGB画像と深度画像から点群化

点群化にはOpen3Dのライブラリを用いる





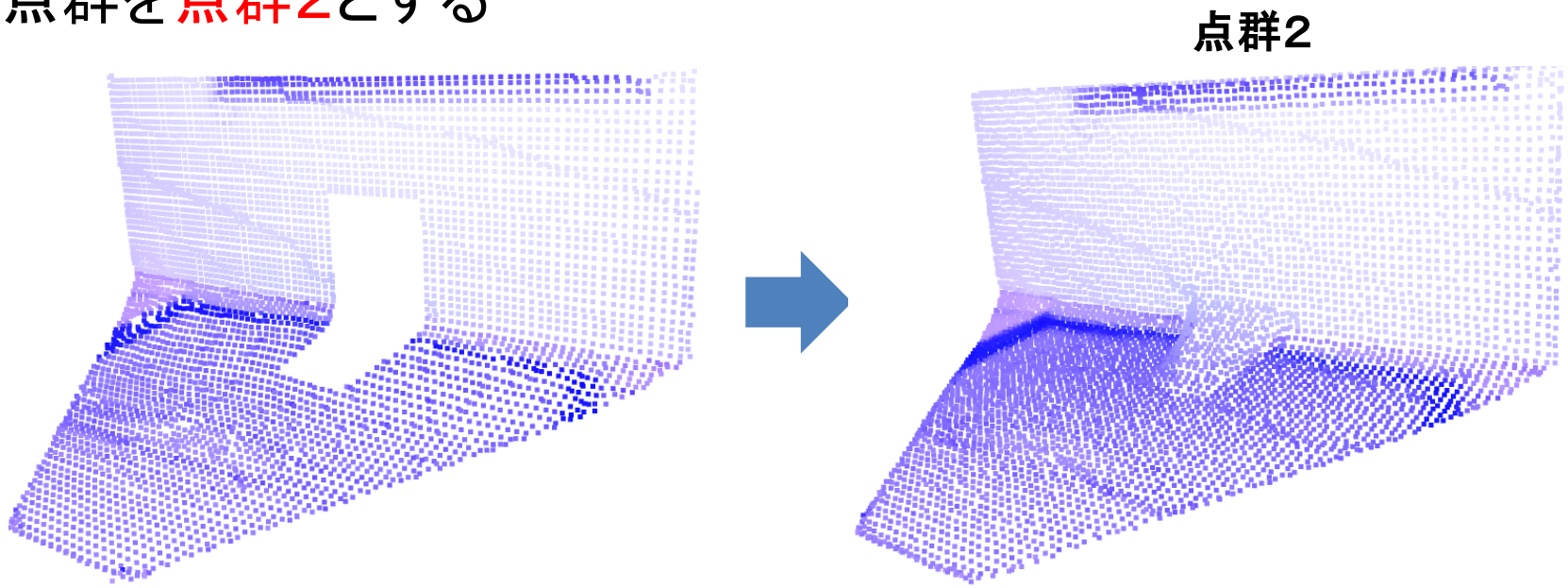
3.2 演習方法



メッシュ化（ドロネーの三角形）

ドロネーの三角形を用いてメッシュ化を行う
→切り取り部分にメッシュを貼ることで補完

得られた点群を点群2とする



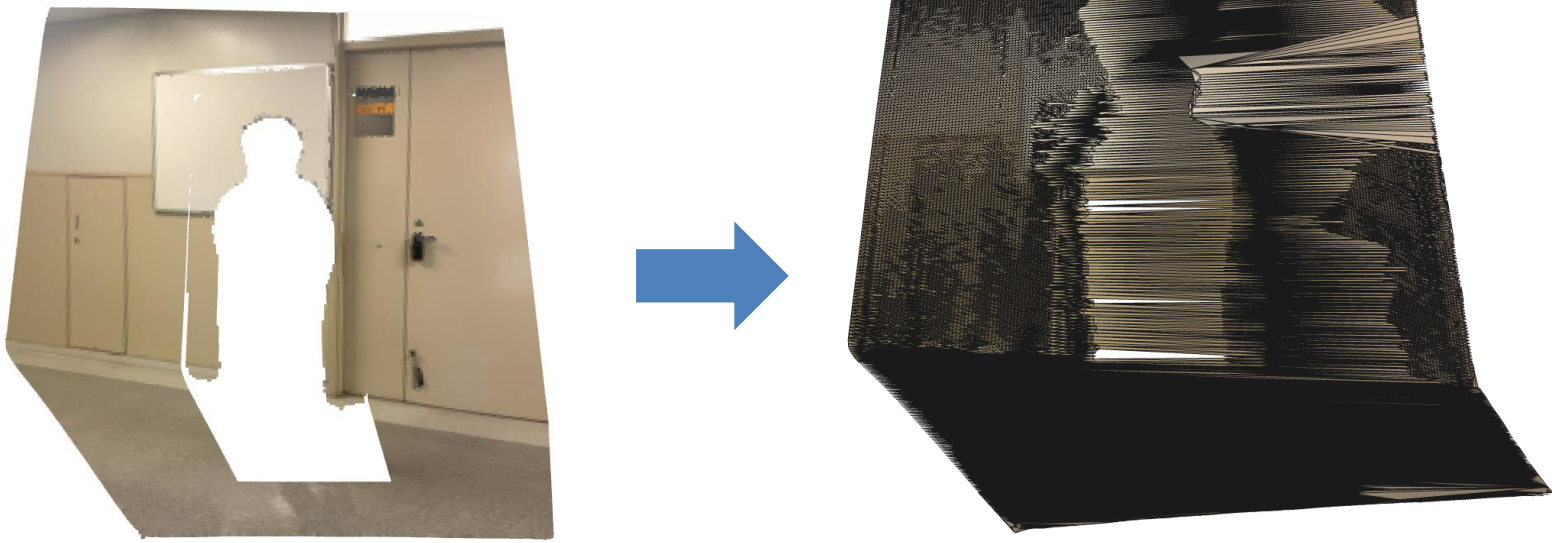
3.2 演習方法

撮影
↓
物体除去
↓
点群化
↓
メッシュ化

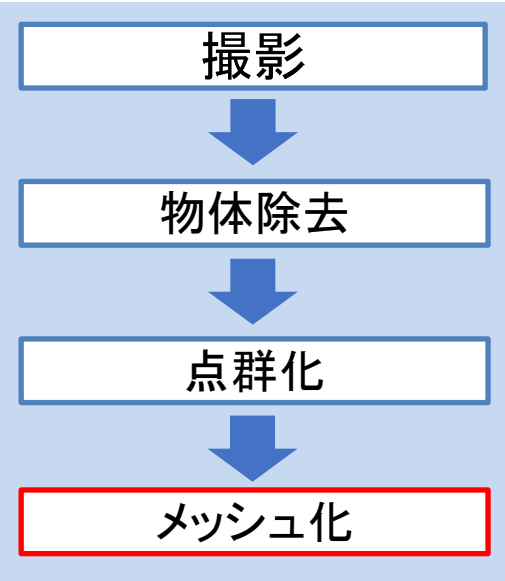
メッシュ化

点群データ内の頂点同士を結んで三角形を作り
面を貼ることで3次元曲面を近似

本演習ではドロネー三角形分割を用いて点群をメッシュ化



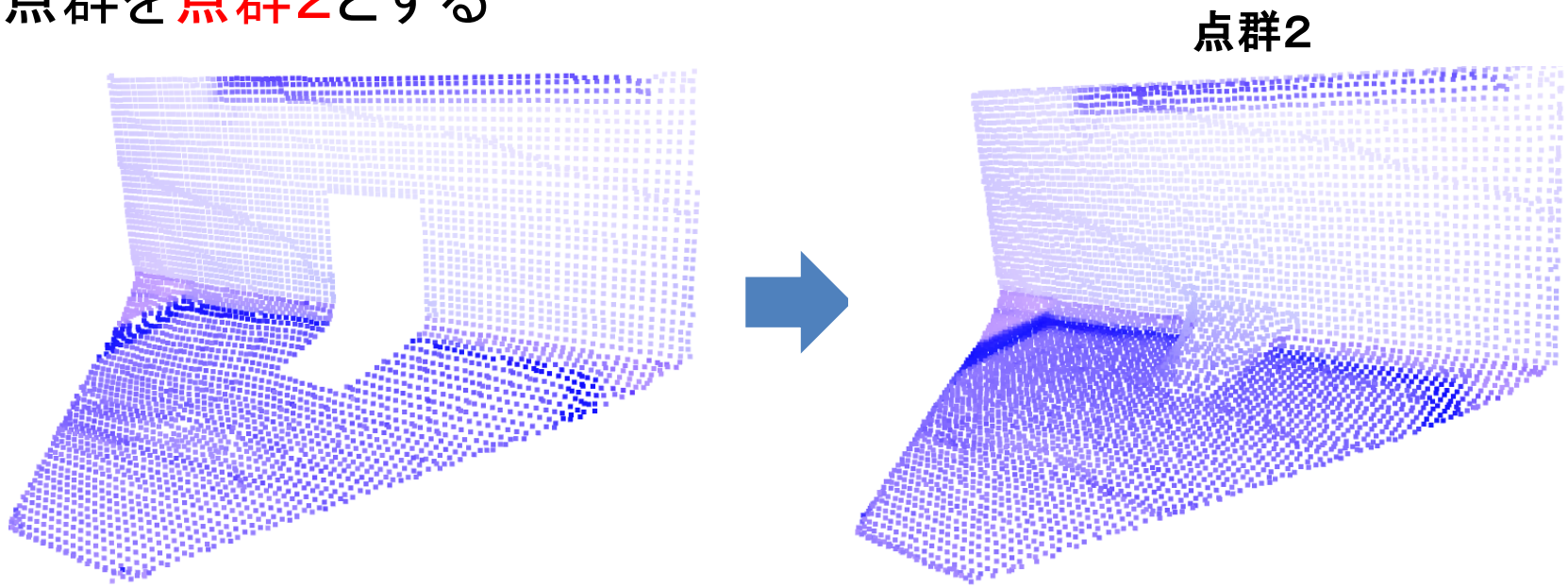
3.2 演習方法



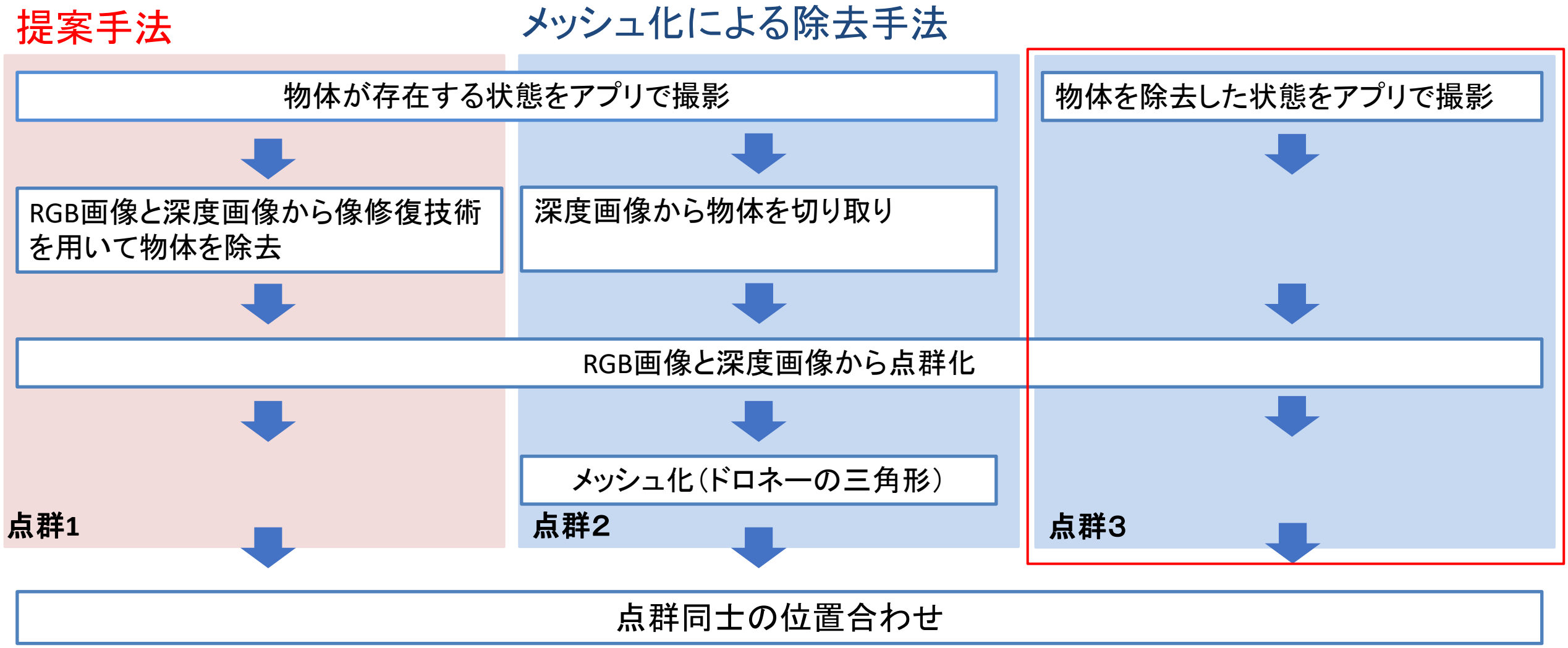
メッシュ化（ドロネーの三角形）

ドロネーの三角形を用いてメッシュ化
→切り取り部分にメッシュを貼ることで補完

得られた点群を点群2とする



3.2 演習方法


提案手法
メッシュ化による除去手法
物体が存在する状態をアプリで撮影
物体を除去した状態をアプリで撮影
RGB画像と深度画像から像修復技術を用いて物体を除去
深度画像から物体を切り取り
RGB画像と深度画像から点群化
メッシュ化（ドロネーの三角形）
点群1
点群2
点群3
点群同士の位置合わせ

撮影

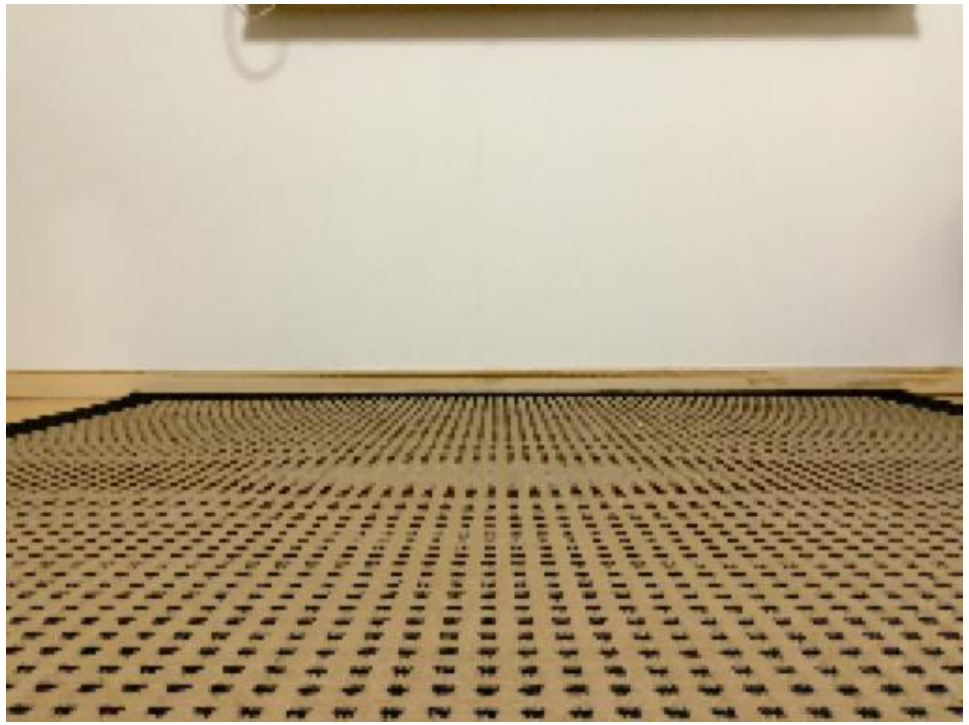
↓

点群化

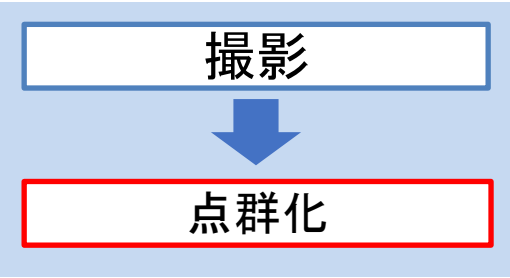
3.2 演習方法

物体を除去した状態をアプリで撮影

RGB画像



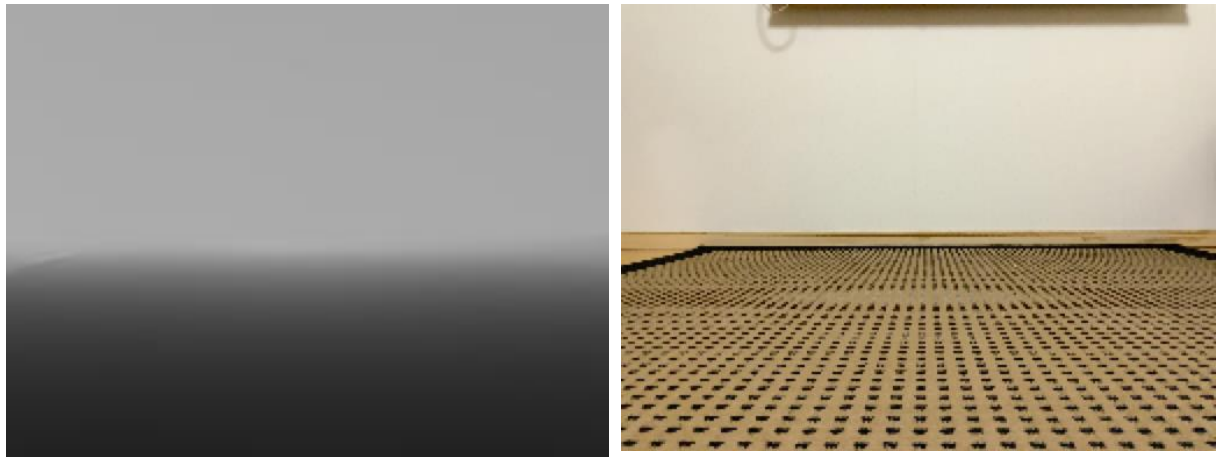
深度画像

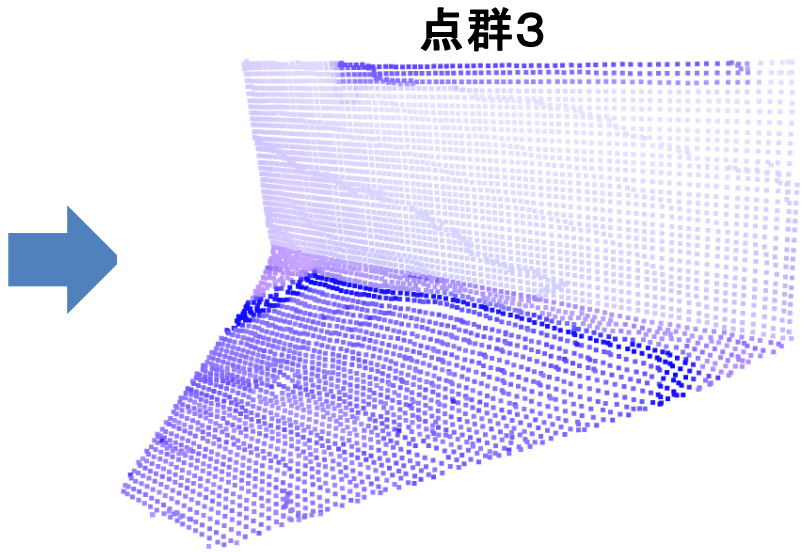


3.2 演習方法

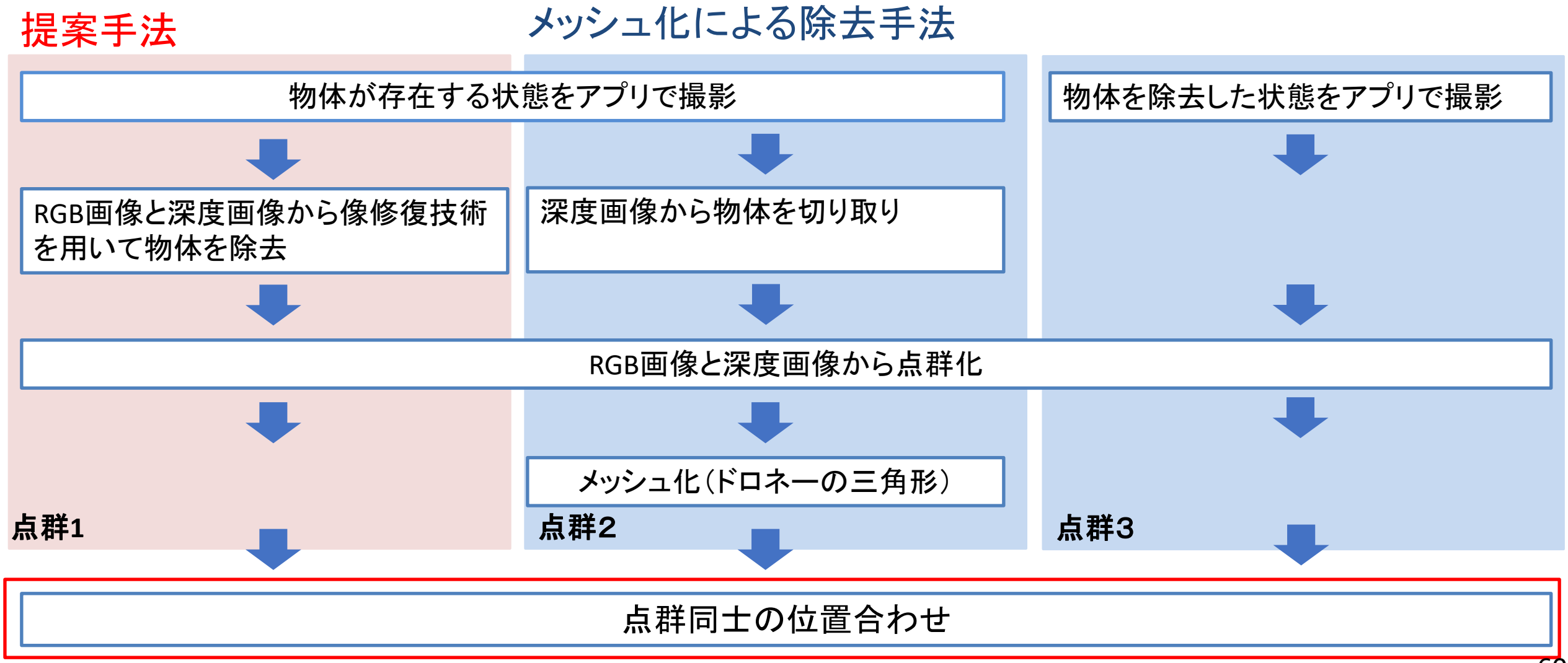
RGB画像と深度画像から点群化

点群化にはOpen3Dのライブラリを用いる
得られた点群を点群3とする





3.2 演習方法

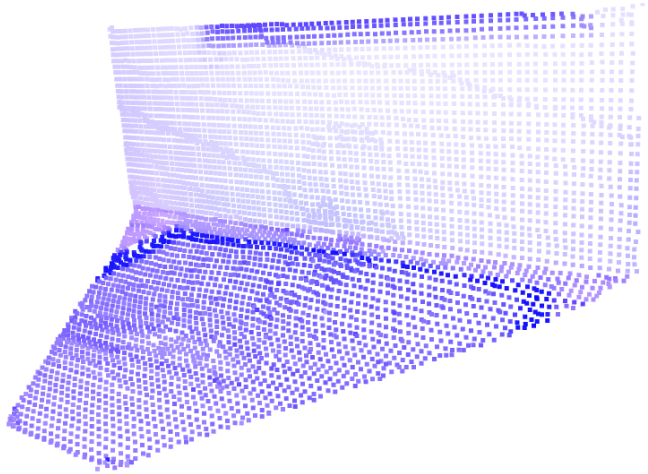

提案手法
メッシュ化による除去手法
物体が存在する状態をアプリで撮影
物体を除去した状態をアプリで撮影
RGB画像と深度画像から像修復技術を用いて物体を除去
深度画像から物体を切り取り
RGB画像と深度画像から点群化
メッシュ化（ドロネーの三角形）
点群1
点群2
点群3
点群同士の位置合わせ

3.2 演習方法

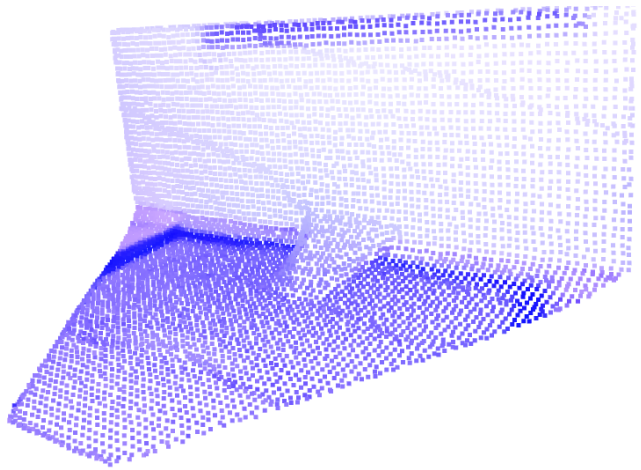
点群同士の位置合わせ

ICPを用いて点群同士を位置合わせ
位置合わせした結果を指標を用いて評価

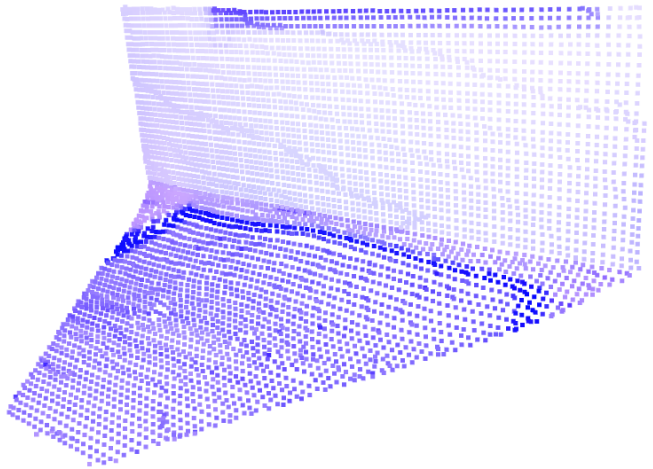
点群1
提案手法



点群2
メッシュ化による除去手法



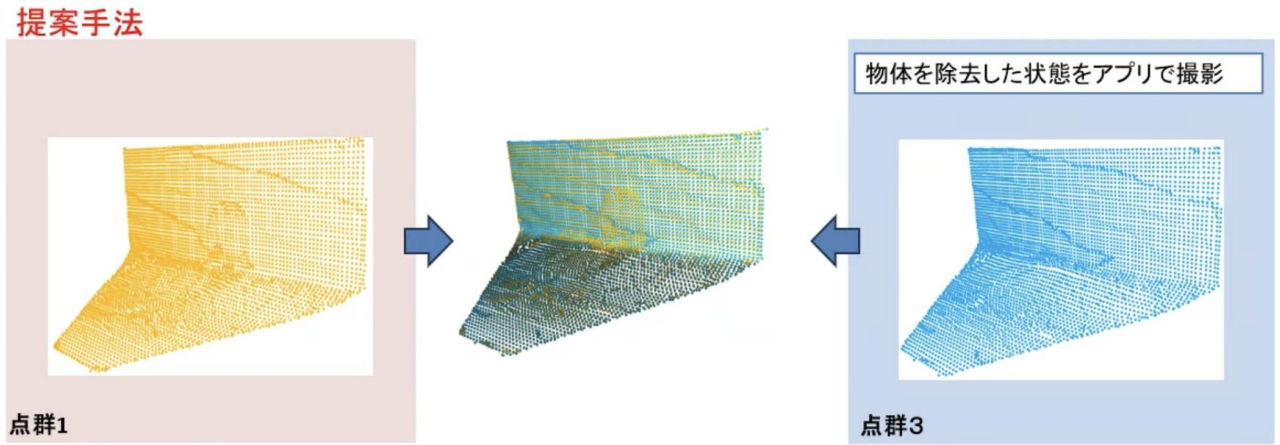
点群3



3.2 演習方法

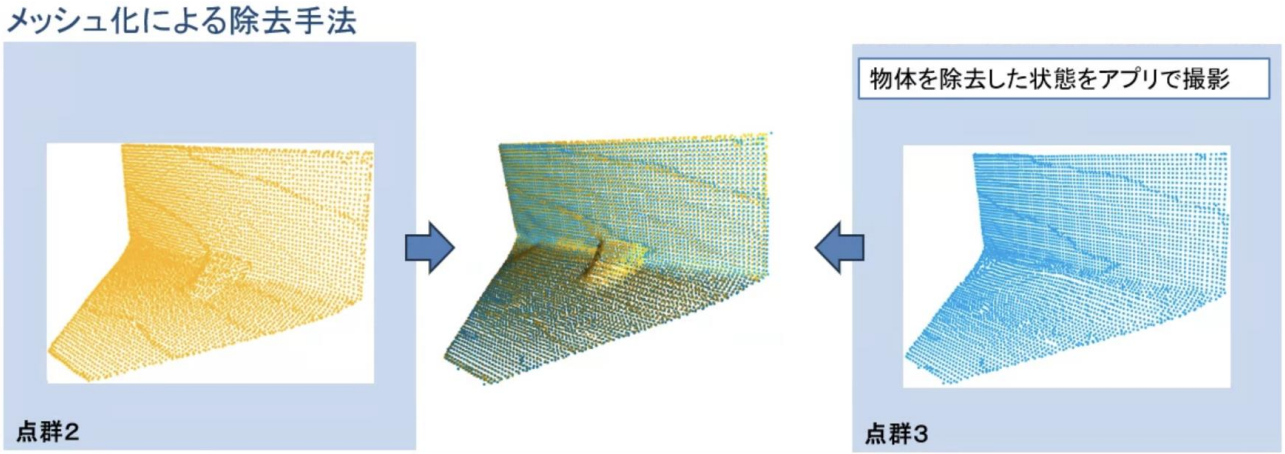
位置合わせ1

点群1と点群3を位置合わせ
→提案手法の補完精度を評価



位置合わせ2

点群2と点群3を位置合わせ
→メッシュ化による手法の補完精度を評価



3.3 評価方法

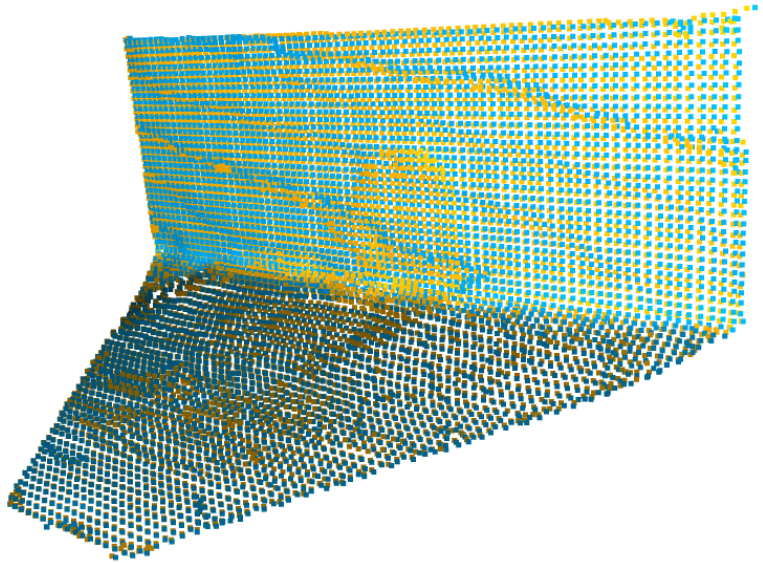
位置合わせを行った結果を2つの評価指標を用いて評価する

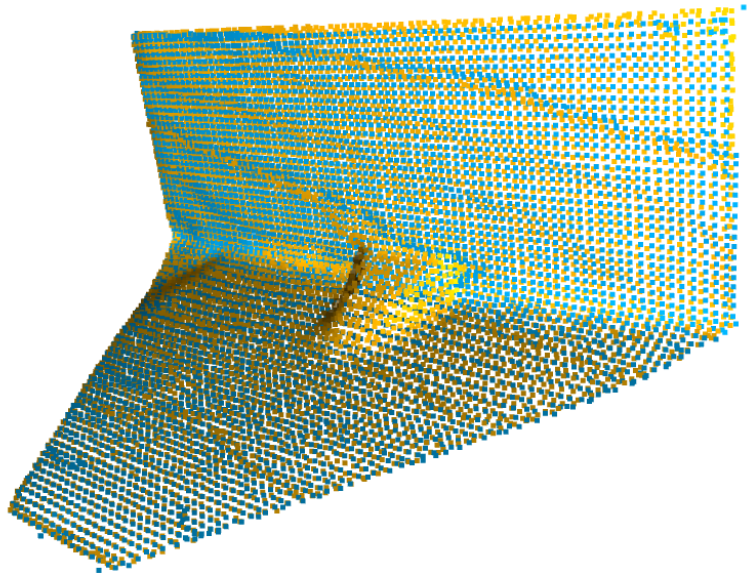
- **fitness**
  対応付けされた点数をターゲット点群内の点数で割ったもの
  値が大きいほど点群同士の重複領域が大きいことを示す
  本演習では補完の精度を意味する

- **inlier_rmse**
  対応付けされた点同士のユークリッド距離の二乗平均平方根誤差
  本演習ではマッチングの精度を意味する

3.4 演習結果

	位置合わせ1（提案手法）	位置合わせ2（メッシュ化による除去手法）
fitness	9.950080e-01	9.370173e-01
Inlier_rmse	4.910894e-05	5.223976e-05

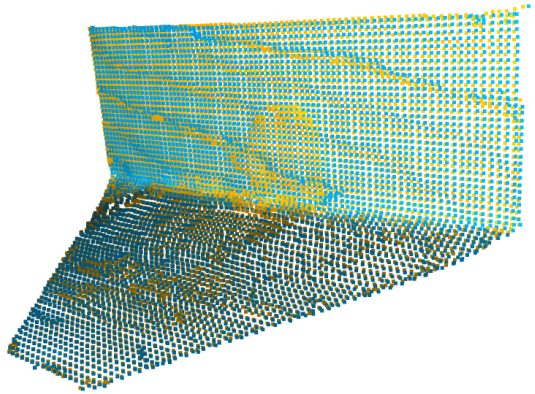


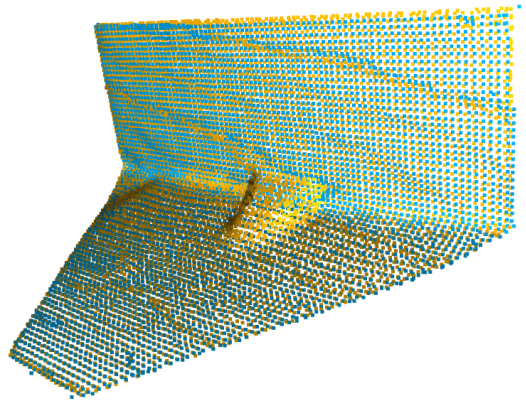


目次

4.1 考察

	位置合わせ1（提案手法）	位置合わせ2（メッシュ化による除去手法）
fitness	9.950080e-01	9.370173e-01
Inlier_rmse	4.910894e-05	5.223976e-05

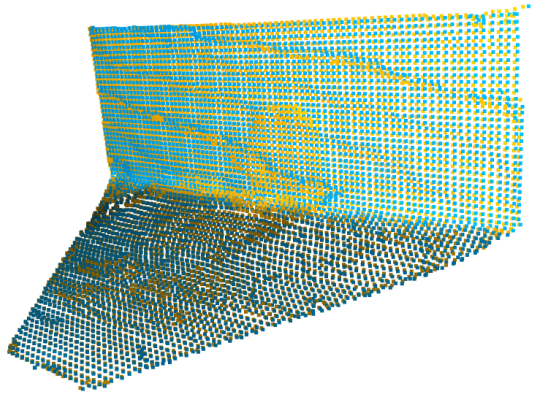


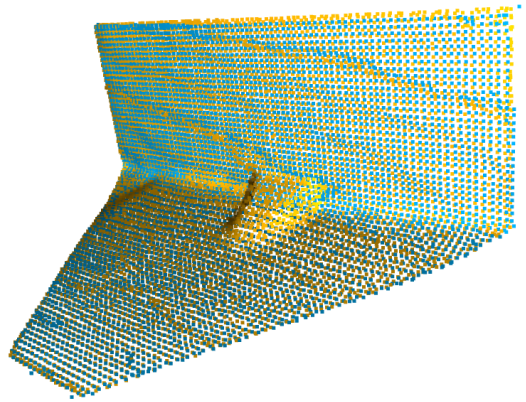


メッシュ化による除去手法と提案手法で大きな違いが見られない
→ICPのマッチング精度はあまり変わらない

4.1 考察

	位置合わせ1（提案手法）	位置合わせ2（メッシュ化による除去手法）
fitness	9.950080e-01	9.370173e-01
Inlier_rmse	4.910894e-05	5.223976e-05





提案手法の方がメッシュ化による除去手に比べてfitnessの値が大きい
→補完した部分の対応点がうまく見つかっている

物体除去後の補完精度が高い

4.2 今後の課題

本演習では実験パターンが不十分

- オブジェクトの種類や背景情報などを変えて実験

→提案手法がどのような場面を得意とするのか調査