基本的点群技術活用の試行

令和4年度 卒業演習

5418071 田村 彩斗
5419047 鈴木 晴斗

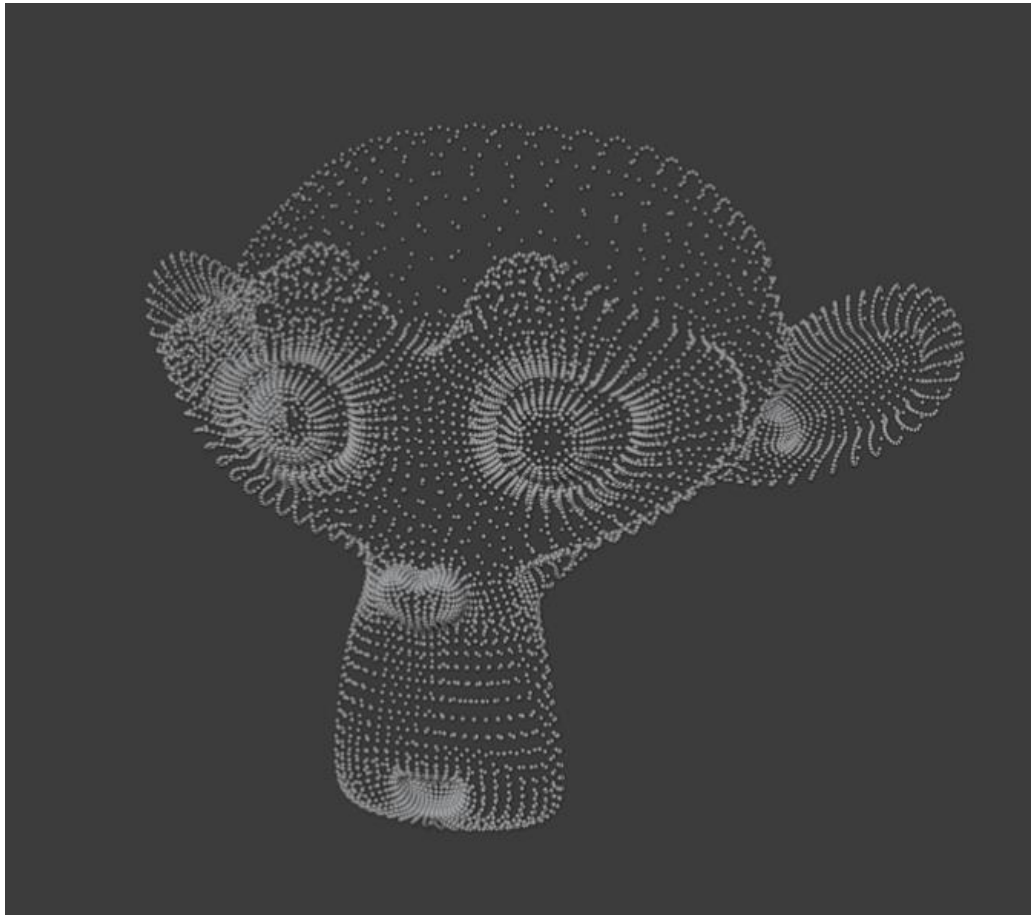
目次

1.1 点群とは

点群？

1.1 点群とは

- 三次元空間内の点の集合
- ポイントクラウドとも呼ばれる



画像引用:Blender/Point Cloud
https://docs.blender.org/manual/en/latest/modeling/point_cloud.html

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？

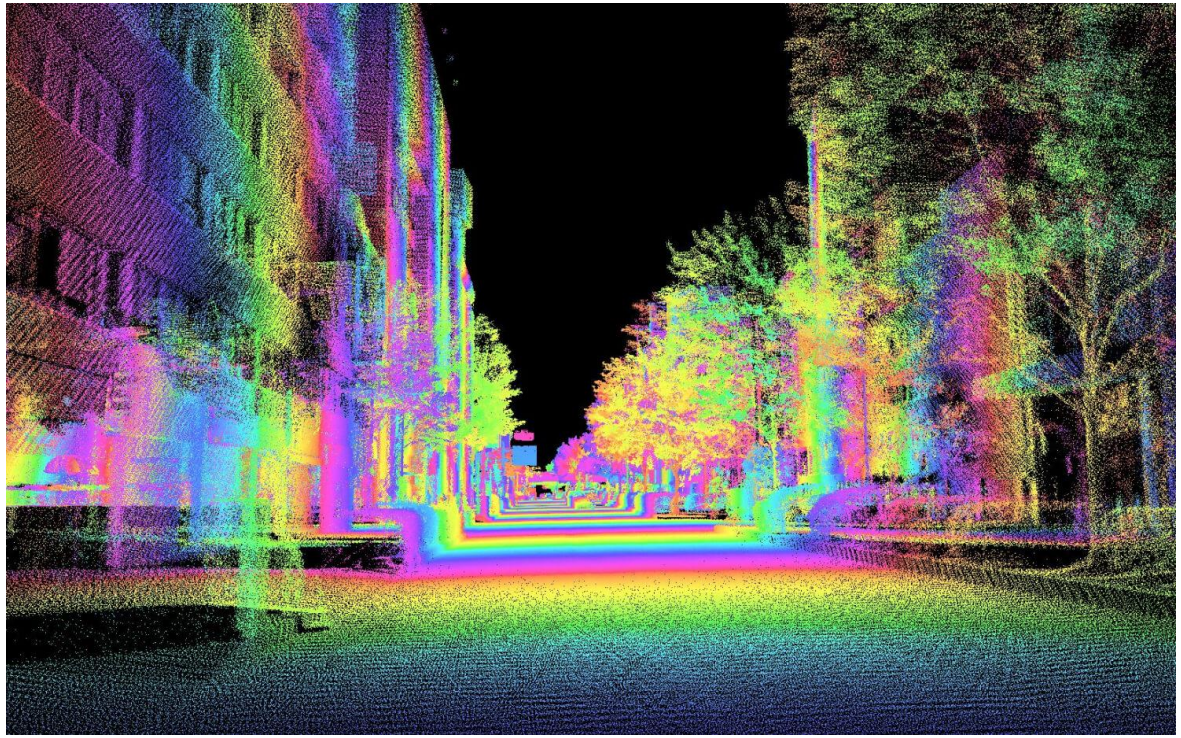
1.2 点群データの活用例 – 車の自動運転

- 車, 歩行者, 建物などの周りの物体をセンサーで読み取り, 外部の情報を立体的に取得する際に点群を活用

1.2 点群データの活用例 – 都市3Dモデル

- 国土交通省が主導する日本全国の3Dモデルの整備・活用・オープンデータ化のプロジェクト

- 都市3Dモデルを作製
  災害時のリスク可視化や
  街づくりなどに活用



画像引用:PLATEAU byMLTU
https://www.mlit.go.jp/plateau/use-case/uc22-044/

1.3 演習概要

本演習では以下の2つを試行

- 室内データの3Dモデル化
- 人除去処理後の補完におけるGLCICと点群データメッシュ化を活用した手法の比較

2.1 点群の表現方法

3次元における位置情報(X, Y, Z)と色情報(R, G, B)の6次元ベクトル

画像引用:Open3D/Geometry/Point cloud
http://www.open3d.org/docs/release/tutorial/geometry/pointcloud.html

2.2 点群を扱う流れ

レーザースキャナーや音波測量などで点群データを取得

取得したデータの前処理

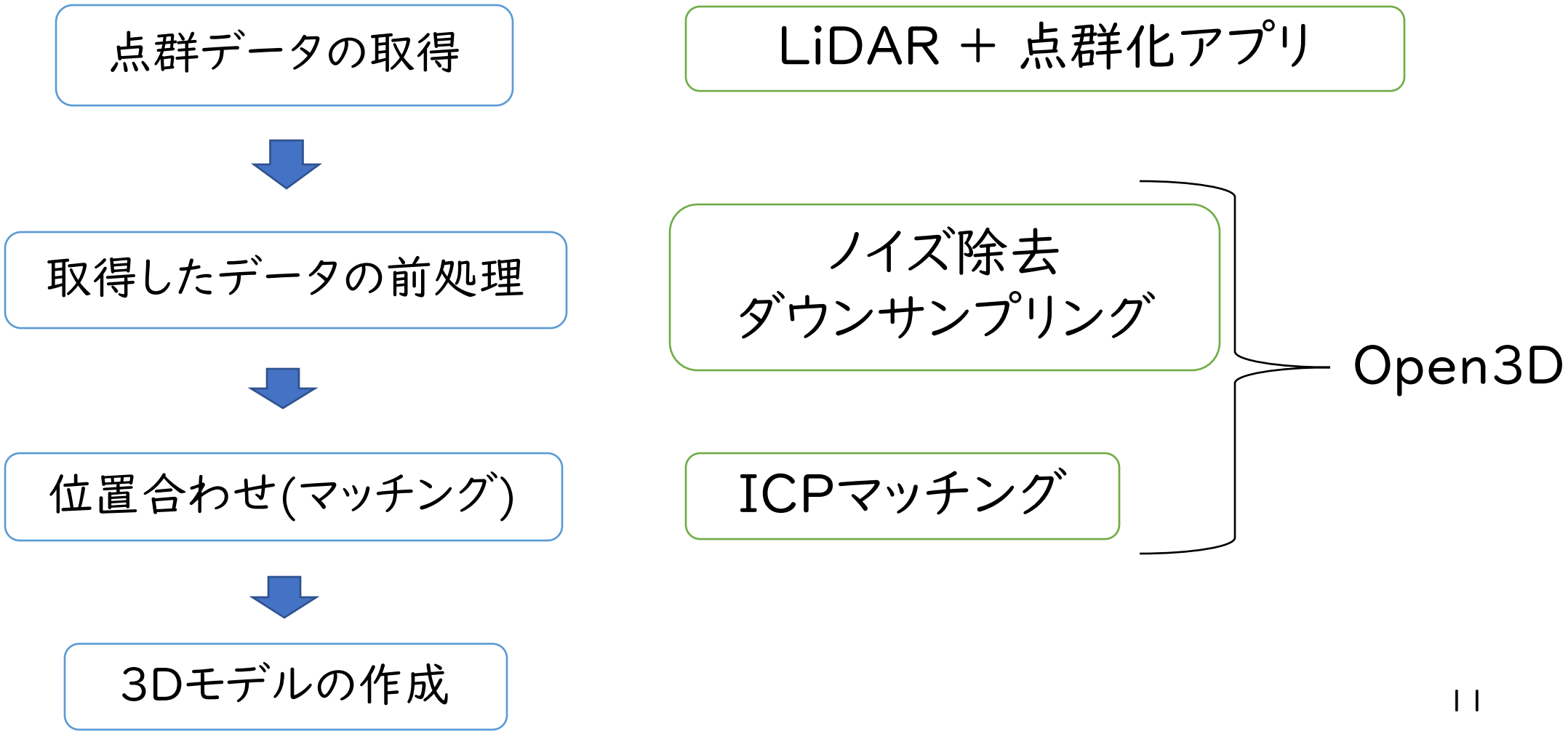
位置合わせ(マッチング)

3Dモデルの作成

活用
- 高速道路,トンネルなどインフラ構造物の点検
- 重要文化財のデジタルアーカイブ
- 映像分野でのVFX
- 都市3Dモデル

⋮

2.2 点群を扱う流れ – 演習



2.3 LiDAR

- Light Detection And Ranging（光による検知と測距）の略称
- レーザー光をパルス状に照射して物体に当て，その光が跳ね返ってくるまでの時間を計測することで物体の位置や形状を測定する技術
- 本演習では，LiDAR 搭載の iPad で Tengun-label が開発した写真を撮影すると点群データに変換するアプリを使用

画像引用:iPhone 12 Proシリーズに搭載の「LiDARスキャナ」って何だ？ カメラとの連動で暗所撮影に力を発揮
https://www.itmedia.co.jp/news/articles/2010/21/news112.html

2.2 点群を扱う流れ – 演習 (再掲)

2.4 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マッチングに利用

画像引用:Open3D: A Modern Library for 3D Data Processing
http://www.open3d.org/docs/release/index.html

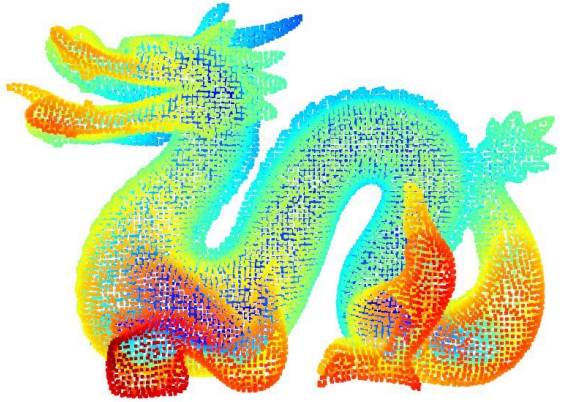
2.5 データの前処理 –ノイズの除去

- センサーで取得した点群データには不要な部分(ノイズ)が存在する場合がある

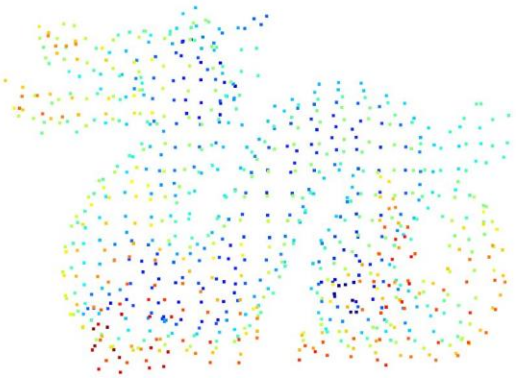
- Open3Dのライブラリ「pcdenoise」などを使用，もしくは手動でデータの編集を行い不要なデータを除去する

2.5 データの前処理 –ダウンサンプリング

- 取得した点群データは膨大な点の数を保存しているため,データをそのまま扱うと処理に時間がかかる
- ダウンサンプリングを行うことで点の数をある程度減らし, 処理にかける時間を減少させる
- 本演習ではOpen3Dのライブラリを使用

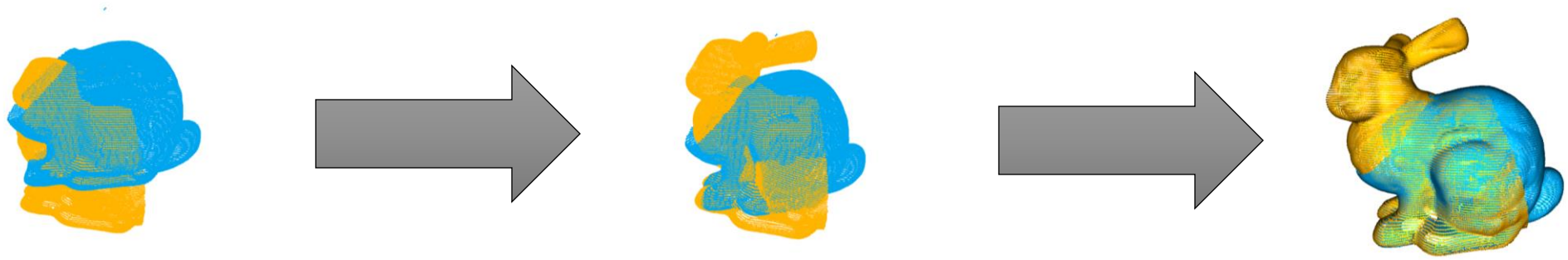


ダウンサンプリング処理



画像引用:技術的特異点
https://tecsingularity.com/open3d/downsample/

2.6 点群データのマッチング



- 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合成する処理
- 点群データの形,大きさは同じ

2.6 点群データのマッチング

- マッチングする際の点群データの形,大きさは同じ

⇒マッチングは回転と平行移動で表現可能

3次元の回転

x 軸周り $\begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{pmatrix}$

y 軸周り $\begin{pmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{pmatrix}$

z 軸周り $\begin{pmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$

３次元の平行移動

$$\begin{pmatrix} T_x \\ T_y \\ T_z \\ 1 \end{pmatrix}$$

2.6 点群データのマッチング

- ソース点群 Ps をターゲット点群 Pt に位置合わせする写像を決定
- この写像は回転移動行列と平行移動行列で表現される

- 本演習ではICPマッチングアルゴリズムの　Point to plane 手法をOpen3Dのライブラリで使用

3 演習 I

3D都市マップ作製の際,建物の3Dモデルなどを作成などに点群を利用
本演習では点群を利用した3Dモデルの作成までを試行

室内データの一部をマッチングし, 3Dモデルを作成

3 本演習での室内データ3Dモデル化の流れ

室内をLiDAR搭載カメラで14枚写真測定

↓

変換した点群データの前処理

↓

点群データのマッチングを行い3Dモデル化

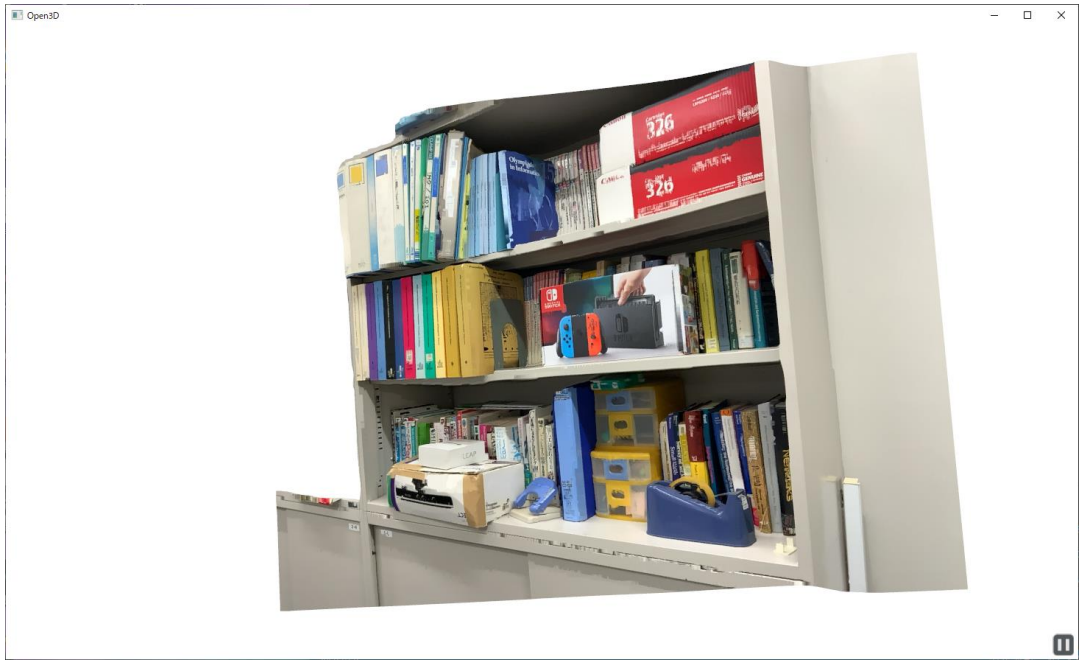
3 演習1 - 演習環境

- MacOS
- Python3.9
- Open3D:0.15.1

3.1 本演習での点群データの取得

撮影した画像

点群データに変換後



Tengun-labelが開発したアプリを使用
LiDARで得た奥行き情報などを撮影した写真と合わせることで
約280万点からなる点群データを作成

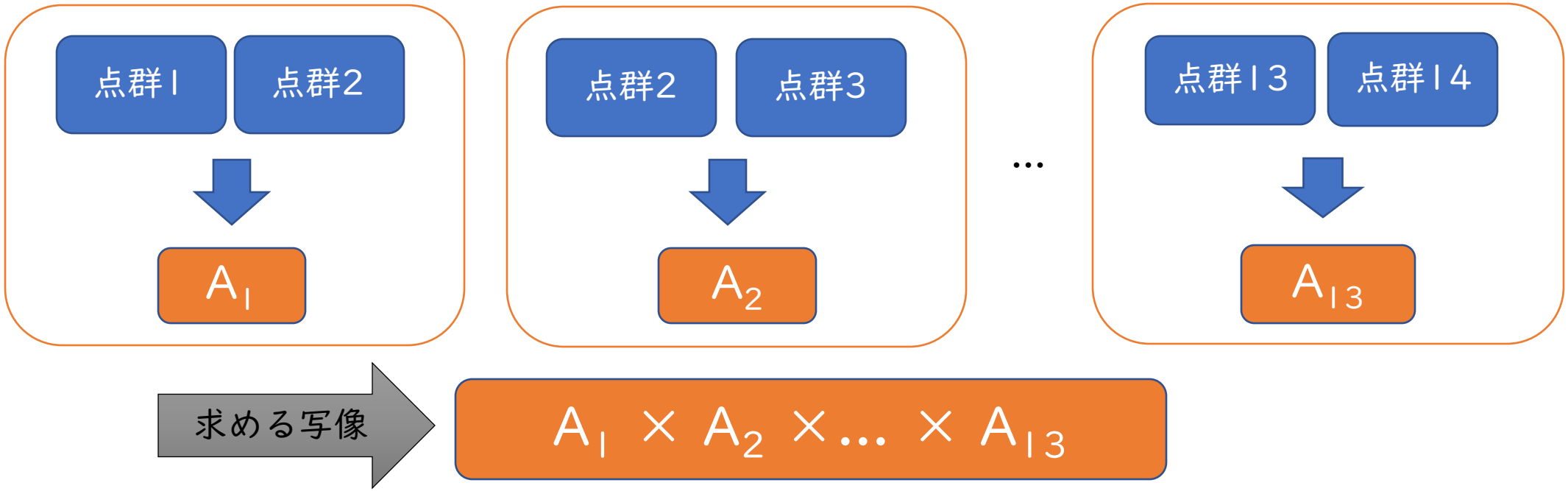
3.2 本演習でのマッチング処理

取得した14個のデータのノイズを手動で除去

すべてのデータをダウンサンプリング
約280万点⇒約10万点

Point to plane で隣接データをマッチング

3.2 本演習でのマッチング処理 – 手順



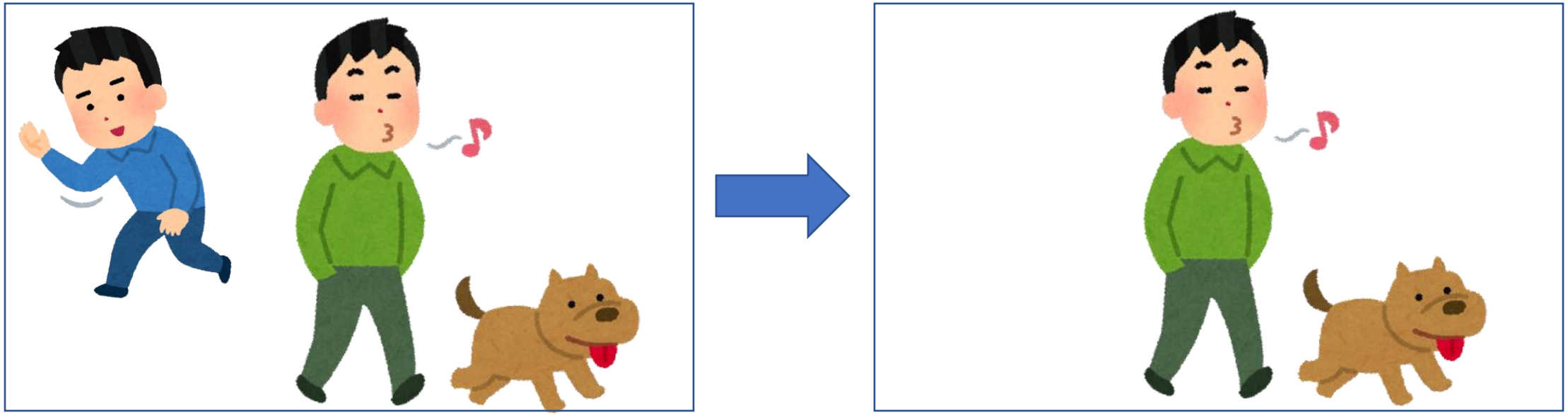
- A_i: 点群 i と点群(i+1)をマッチングする写像(行列)

- 点群1と点群14をマッチングする写像: $A_1 \times A_2 \times \ldots \times A_{13}$

3.3 3Dモデル

4 演習2

- 不要な部分を選択することでその部分を消去し，周りの情報を基に空白を補完する機能を持つスマートフォンアプリが存在
- 本演習では「画像の人物の除去→空白の補完」を以下の2通りの方法で試行
  - A: 人物領域を隠しGLCICで補完
  - B: 点群データに変換後，人物領域を切り取りメッシュ化



4 演習2 - 演習環境

- Windows11
- Ubuntu 20.04
- Python 3.8.10

4.1 演習2の流れ

A GLCIC
1. 写真を撮る
2. 人物領域をマスク
3. GLCICで補完

B メッシュ化
1. 写真を撮る
2. 点群データに変換
3. 人物領域の点群を除去
4. メッシュ化で補完

4.2 GLCIC

- 2017年に飯塚らが開発したディープネットワークによる画像補完技術
  - 畳み込みニューラルネットワークを用いてシーンの大域的かつ局所的な整合性を考慮した画像補完手法

4.2 GLCIC

元画像

一部をマスクした画像

画像引用:ディープネットワークによるシーンの大域的かつ局所的な整合性を考慮した画像補完
http://iizuka.cs.tsukuba.ac.jp/projects/completion/ja/

4.2 GLCIC

元画像

補完後の画像

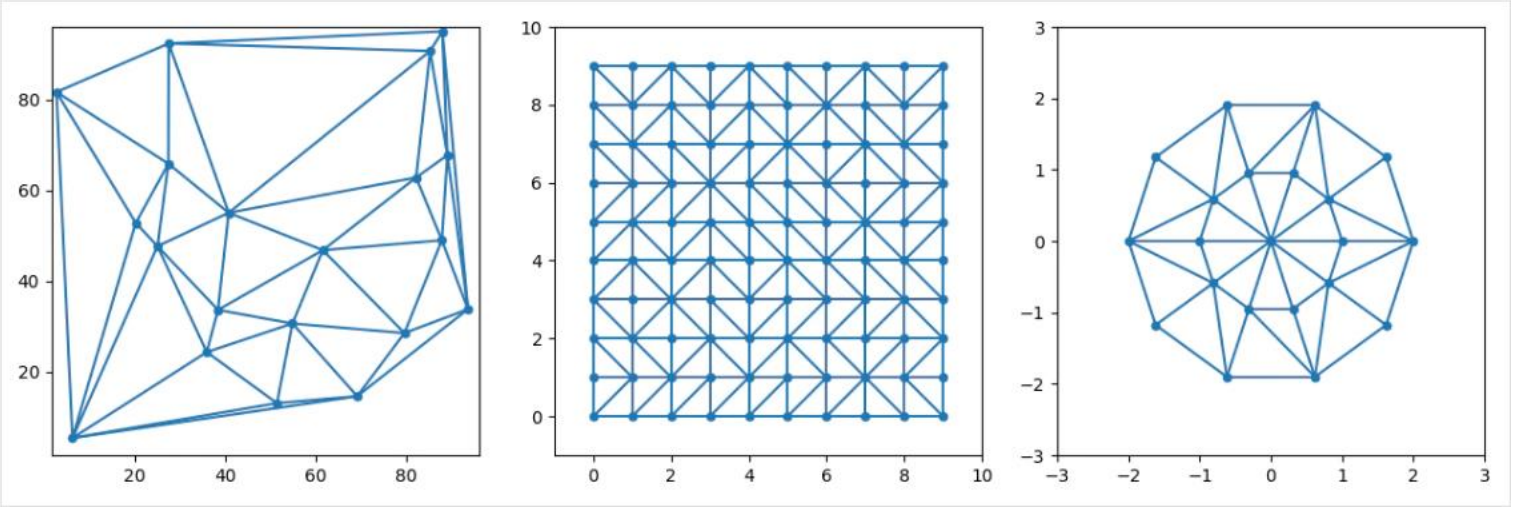
4.3 点群除去

点群データに変換

人物領域の点群を除去

4.4 メッシュ化

- 点群データのそれぞれの点を頂点とし他の頂点と繋ぎ，二次元の三角形や四角形で3次元曲面を近似
- 本演習ではドロネー三角形分割と呼ばれる散らばった点同士を繋いで三角形で敷き詰められた状態にするアルゴリズムを使用



画像引用:**ドロネー三角形分割の期待最速アルゴリズム**
https://qiita.com/tsukasa__diary/items/a835c17e5bf4935636c4

4.5 結果 – GLCIC

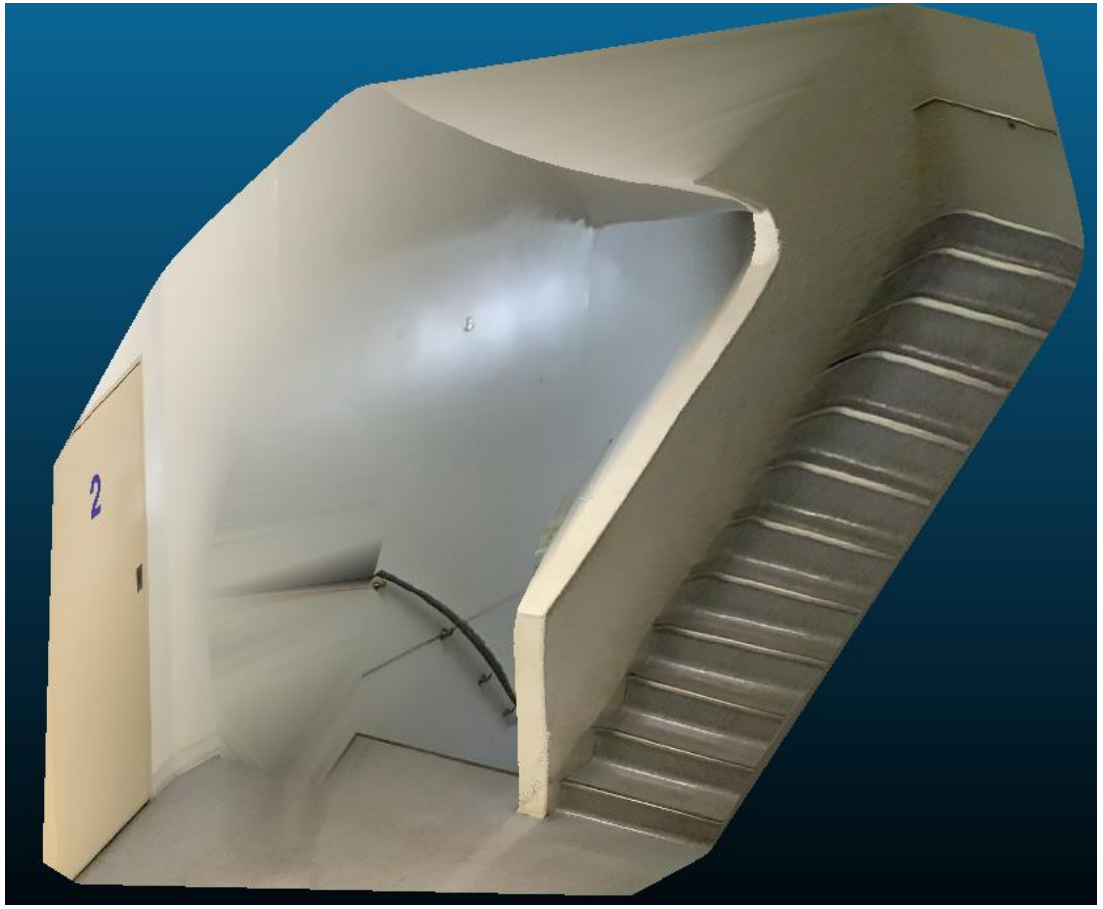
取得した写真

補完後の画像

4.5 結果 – メッシュ化

点群データに変換

メッシュ化



5 最後に

- 演習1: 室内データの3Dモデル化を扱い，自動運転や都市3Dモデル化などでの基本的点群技術を確認

- 演習2: 得意なシーンに違いがありそう
  - GLCIC⇒背景が複雑な場合
  - メッシュ化⇒背景が単調な場合

- 本演習では基本的点群技術の試行

⇩

今後，発展させたいこと:
- 演習1: より広範囲での3Dモデル構築
- 演習2: 高精度な補完処理への応用