



# Occupancy Networksを用いた 都市3DモデルのLOD1からLOD2への超解像化

日本大学文理学部情報科学科-谷研究室

5422057 川元瑛仁

# 目次

---

- はじめに
- 目的
  - 動機・背景
  - 先行研究
  - 本演習の目標
- 演習手順
- 準備
- 演習結果
- 考察・今後の展望

## 概要

---

卒業演習のテーマ

### Occupancy Networksを用いた都市3Dモデルの LOD1からLOD2への超解像化

- ❖ 演習に用いる都市3Dモデルについて
- ❖ LODについて
- ❖ Occupancy Networksとは

## 演習に用いる都市3Dモデルについて



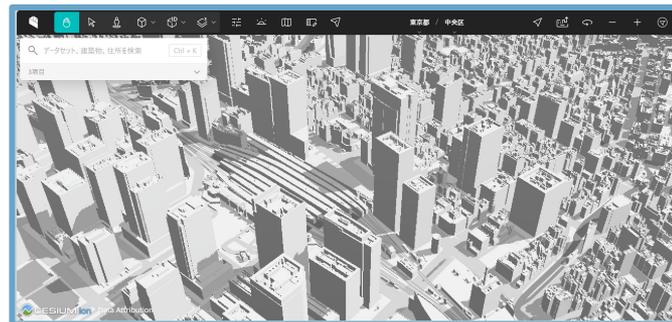
**PLATEAU** : 国土交通省が主導する, 日本全国の3D都市モデルの整備・活用・オープンデータ化のプロジェクト

データ形式: [CityGML](#)

**CityGML** : 建築物や土地利用, 交通といった単位でモジュール化された仕様を必要に応じて取り込んで利用できる拡張可能なフォーマット

PLATEAUの3D都市モデルのデータは,  
「G空間情報センター」の「3D都市モデル (Project PLATEAU) ポータルサイト」  
から入手できる。

<https://www.geospatial.jp/ckan/dataset/plateau>



## LODについて

### LOD (Level Of Detail): 3D都市モデルの詳細度のこと

PLATEAUでは、データのある地域では必ずLOD0と1のデータが存在していて、一部地域にのみLOD2, さらに限られた建物にのみLOD3のデータが存在



図 1-7 LODによる詳細度の違い

LOD	提供されるエリア	解説
LOD0	対象のすべてのエリア	平面に投影したもの。高さ情報がない
LOD1	対象のすべてのエリア	直方体の組み合わせで構成されたもの。箱モデル
LOD2	特定のエリア	屋根や壁などを再現したモデル
LOD3	限定されたエリア	LOD2をさらに詳細に表現し、開口部や立体交差などを表現できるようにしたモデル
LOD4	一部の建築物	建物の内部までモデル化したもの

表 1-2 建築物のLOD

# Occupancy Networksとは

Lars Mescheder, Michael Oechsleらによって2019年に発表された  
ニューラルネットワークを用いて3D物体やシーンの形状を連続的な関数として表現する手法

3D空間上の任意の位置  $p=(x,y,z)$   $p=(x,y,z)$  に対して  
その点が形状内部にあるか(=占有されているか) を  
0~1 の確率(Occupancy)として予測する関数  $f_\theta(p)$   $f_\theta(p)$  として表現

## Convolutional Occupancy Networks

$$f_\theta : \mathbb{R}^3 \rightarrow [0, 1]$$

- 直接的な上位互換モデル
- 畳み込みエンコーダを採用し、点群のクエリ時に近隣の特徴もデコーダに渡すことで、より高解像度かつ複雑な形状の再構成を行えるようになった

## 動機・背景

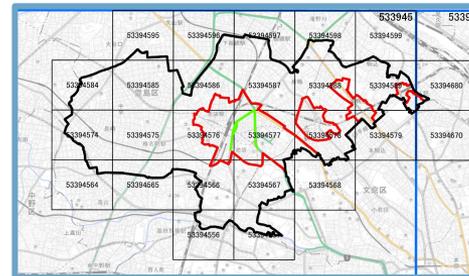
現在, PLATEAUでLOD2のデータが存在する地域は非常に限られている

→自分が欲しい建物のLOD2のデータがない場合がある

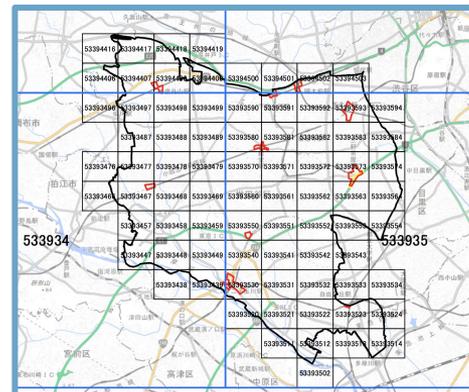


任意の建物に対してLOD1モデルから新たなLOD2モデルを生成するシステムを機械学習技術を用いて実現することができれば, この問題を解決することができるのではないかと

豊島区



世田谷区



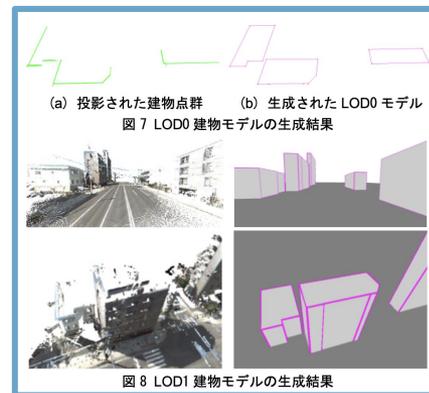
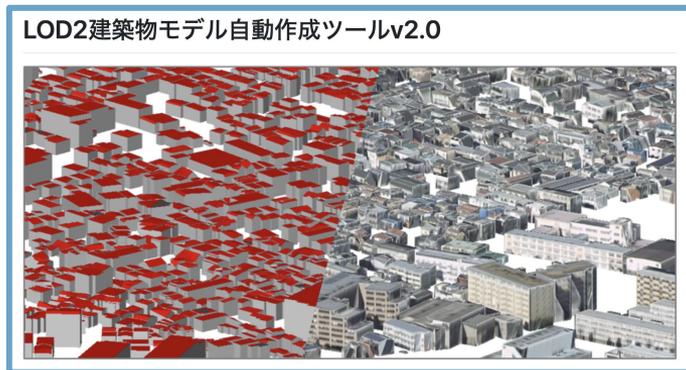
# 先行研究

## LOD2建築物モデル自動作成ツールv2.0

<https://github.com/Project-PLATEAU/Auto-Create-bldg-lod2-tool>

市街地 MMS 点群からの機械学習手法によるセグメンテーションと 建物 LOD モデリング

[https://www.istage.jst.go.jp/article/pscjspe/2016S/0/2016S\\_779/\\_pdf](https://www.istage.jst.go.jp/article/pscjspe/2016S/0/2016S_779/_pdf)



## 本演習の目標

---

### 無期限無制限での大きな目的

AIにPLATEAUのLOD1データを学習させ、  
LOD2の3Dモデルが未実装のエリアに対して、LOD2を作成(生成)

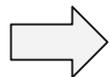


### 卒業演習での目標

LOD1→LOD2の推定変換と生成を行うAIを用いて、  
LOD2の正解データと生成されたデータの比較を行い、その精度を調べる

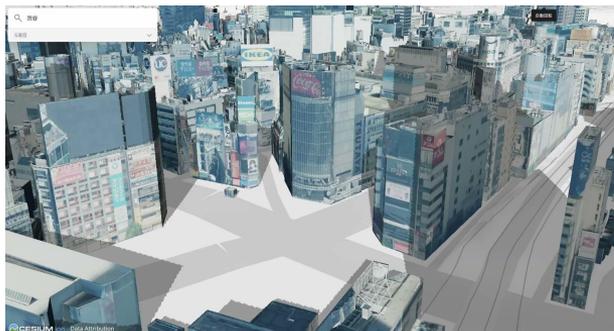
## LOD2を生成するメリット

LOD1からLOD2になることで、建物を単なる直方体ではなく、実際の建築構造に近い形として扱えるようになる

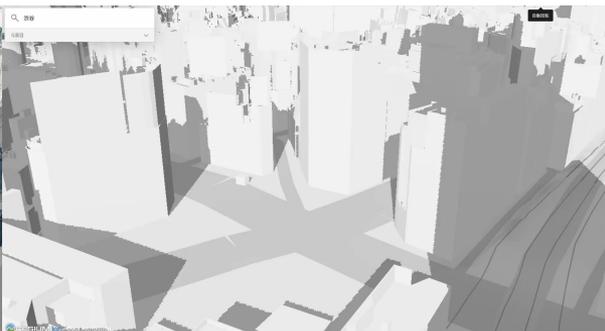


- ❑ 都市をより“現実に近い形”でシミュレーションできる
- ❑ ゲームや映像作品などで使うと、没入感が増す

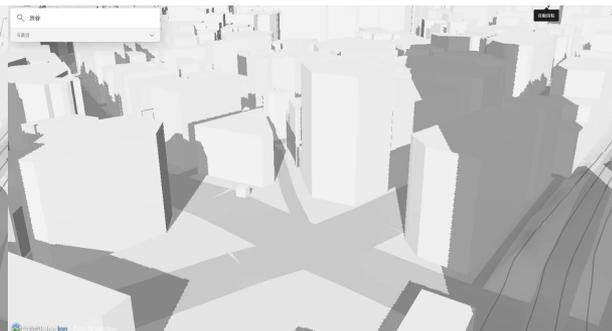
LOD2(テクスチャ有)



LOD2(テクスチャ無)



LOD1



# 演習手段

---

1. G空間情報センター内のPLATEAUのポータルサイトからCityGMLデータをダウンロード
2. ダウンロードしたCityGMLデータを建物ごとにLOD1とLOD2のデータを取り出して、OBJに変換
3. データをOccupancy Networksに学習させるための前処理として、OBJデータから以下の3つのデータを生成
  - inputs.npz: エンコーダへの入力データ
  - points.npz: デコーダへの入力データと教師データ
  - pointcloud.npz: デバック・確認用データ
4. Occupancy Networksに学習させる
5. 学習済みのモデルを使って3Dモデルを生成させ、それを可視化

# 準備

---

- 学習データの用意
- 学習させるAIの準備
- 学習データの前処理

- 取得
- LOD1とLOD2の抽出
- 変換 (CityGML→CityJSON)
- 建物の抽出
- 変換 (CityJSON→OBJ)

**CityGML-tools** : CityGMLファイル进行处理・変換・解析するためのコマンドラインツール集

**cjio** : CityJSON ファイルの編集・変換・検証・軽量化などを行う専用のコマンドラインツール集

# 学習データの用意

## LOD1とLOD2の抽出

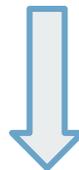
CityGML-toolsのfilter-lodsコマンドを用いて、特定のLODのデータのみを抜き出す

## 建物の抽出

cjioのsubsetコマンドを用いて、建物IDを指定して特定の1棟だけを抜き出す

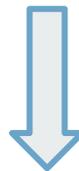
## ファイル形式の変換

CityGML



CityGML-toolsのto-cityjsonコマンドを用いて変換

CityJSON



cjioのexportコマンドを用いて、CityJSONからOBJメッシュを直接生成

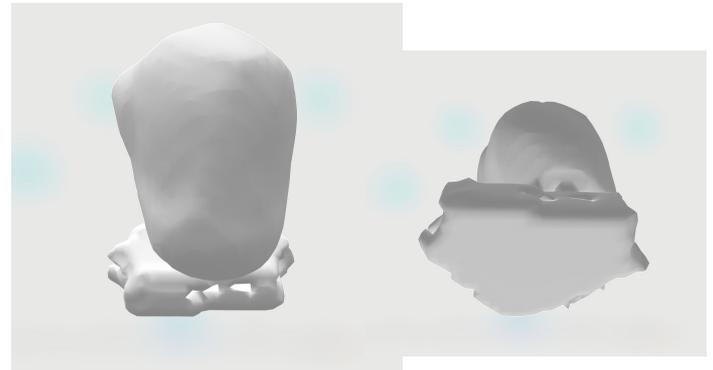
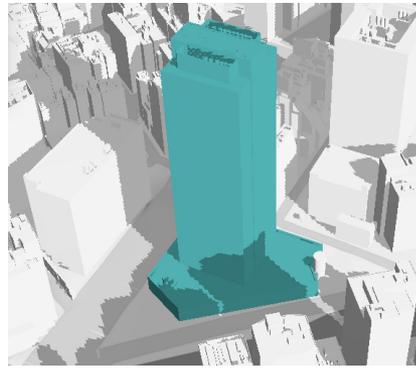
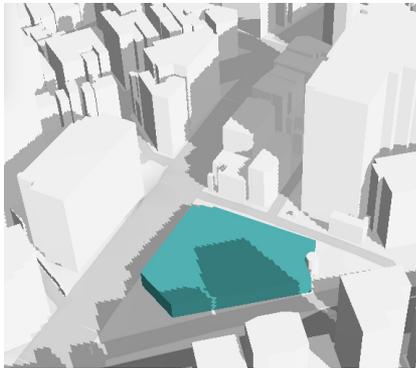
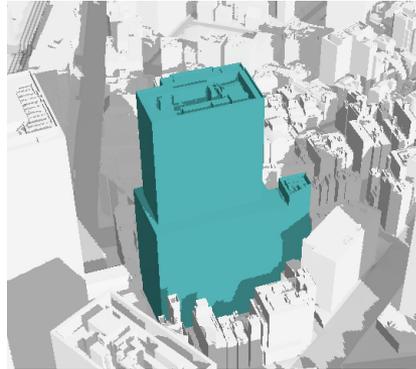
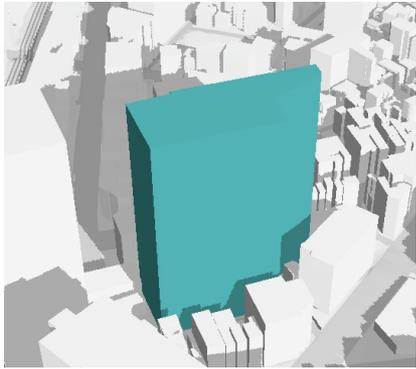
OBJ

## 学習データの前処理

**npz** : NumPy専用の複数配列をまとめて保存できる圧縮ファイル形式  
対応関係を壊さず保存できる

- **inputs.npz** : LOD1から作ったvoxel(占有グリッド)  
ConvONetのエンコーダへの入力データ
- **points.npz** : LOD2の周囲からランダムにサンプリングされたクエリ点の座標と、  
その点がLOD2の内部か外部かを示す正解ラベル(0または1)  
デコーダへの入力データ(クエリ点の座標)  
教師データ(正解データ)
- **pointcloud.npz** : LOD1から得た点群の座標  
LOD1のメッシュ表面から、均一な数の点群を  
サンプリングしたもの  
~~エンコーダへの入力データ(ONet時点での)~~  
確認・デバック用データ

# 演習結果



# 考察・今後の展望

---

## 考察

本演習での学習データの与え方や学習手法では、建物らしい平らな面や直角の構造などを十分に学習できず、モデルは「建物としての構造」をうまく生成させることができなかった

## 今後の展望

- ❑ LOD 変換に必要な建物の特徴がより分かる形で情報を与えると改善するか
- ❑ Convolutional Occupancy NetWorksよりも建物の生成に適した学習モデルを見つけ、それを用いて演習を行う