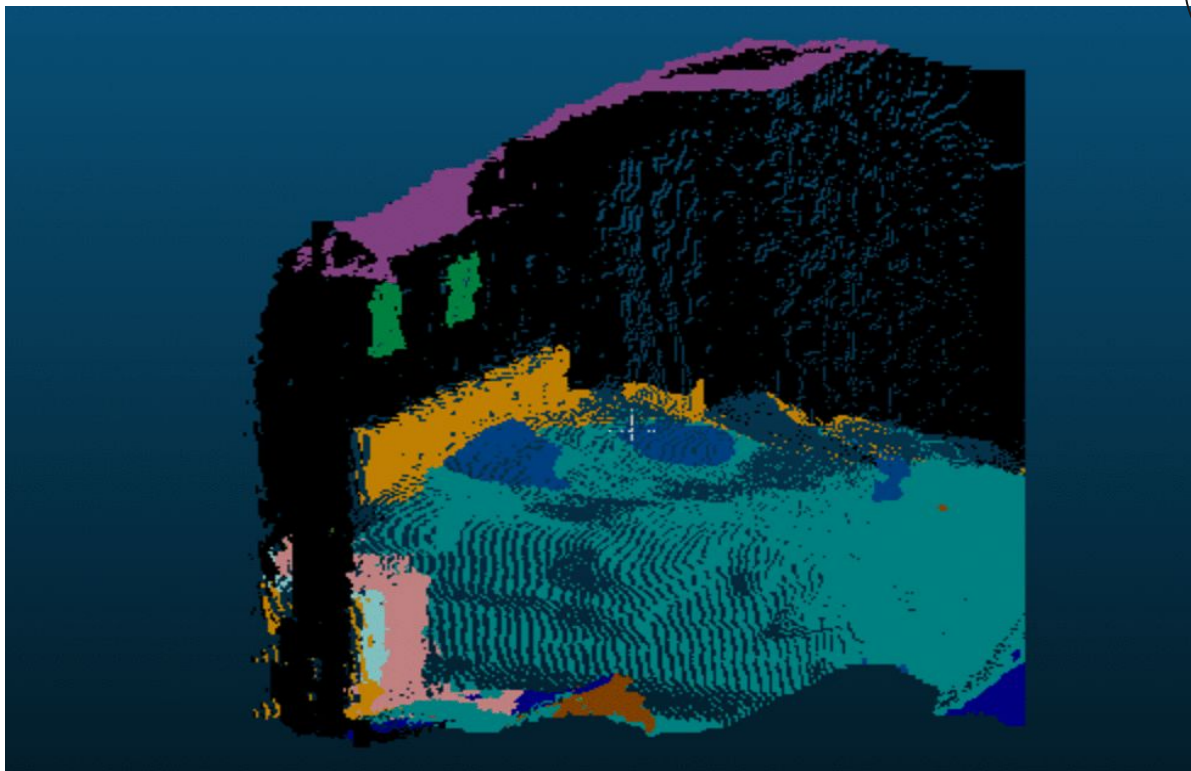




2D セグメンテーションと深度情報を活用した リアルタイム 3D セマンティックセグメンテーションの試み

株式会社Tengun-label 岩澤 秀樹との共同研究
日本大学文理学部情報科学科 谷研究室
5418052 オウ イチホウ

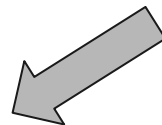
2月14日宮田研・谷研・大澤研 合同発表会



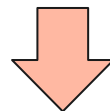
何のデータ？
画像？...

セマンティックセグメンテーション

どんな特徴ある？



ベッド、枕、布団など、全てのものが種類に応じて違う色で表現されている。



すべての物が分類されている!!!

しかし、さっきのデータと比べて何かが違う...



できるかもし
れない！

2D画像

?

3Dデータ

「距離」を2D画像のZ軸とする

もし、2D画像中のすべてのピクセルに対して、
【カメラ】<—>【撮影物】の距離が分かれば...

Depthデータ



Depth画像

左の画像は実際にセンサーで取得した**Depthデータ**と呼ばれる形式のデータで、各ピクセルが3Dカメラ(深度センサー)の平面から撮影対象物までの距離を示しており、その値を濃淡で**画像**として表示している。

色の濃い点が近くにある点、淡い点が遠くにある点として表現されている。

点群



点群で表現されている兎

点群とは

点の集まりのことをいう。
ポイントクラウドと呼ばれることもある。

点群データは以下の情報から構成されている。

- ・三次元座標値(X,Y,Z)
- ・色の情報(R,G,B)

つまり、一つの点には**複数の属性**を持っている。

応用

- ・高速道路、トンネルなどの劣化の点検
- ・災害による被害確認
- ・重要文化財、公共施設のデジタルアーカイブ
など、様々な用途に活用されている。

目次



1. はじめに

1.1 セマンティックセグメンテーションとは

1.2 2D・3Dセグメンテーションについて

1.3 問題意識

1.4 先行研究

1.5 演習の動機・目的

2. 演習

2.1 演習方法

2.2 演習結果

3. 終わりに

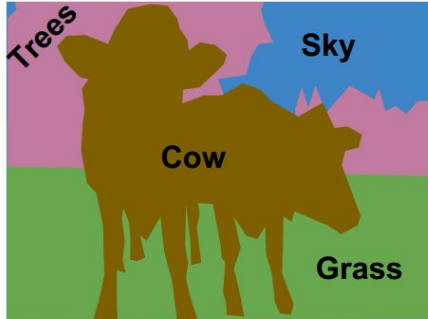
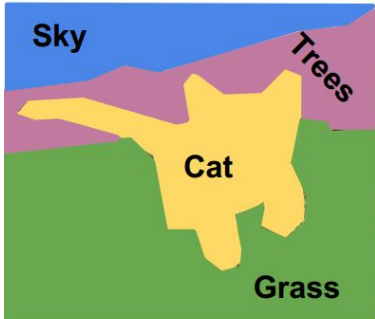
3.1 まとめ

3.2 今後の課題・展望



1. はじめに

1.1 セマンティックセグメンテーションとは



直訳で「**意味による分割**」の意味になる。

画像内の全ピクセルに対して、（事前に定義された）**ラベル**や**カテゴリ**を関連付けることで、深層学習で実現されることが多い。

コンピュータはセマンティックセグメンテーションを用いて、画像の「**意味**」を理解できるようになる。

1.2 2D・3Dセグメンテーションについて

2Dセグメンテーション

対象: カラー画像 (RGB画像)

流れは以下のなる。

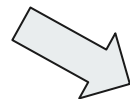
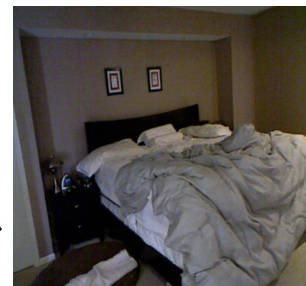
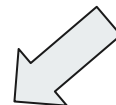
元の画像



正解ラベル、教師データ



深層学習モデル

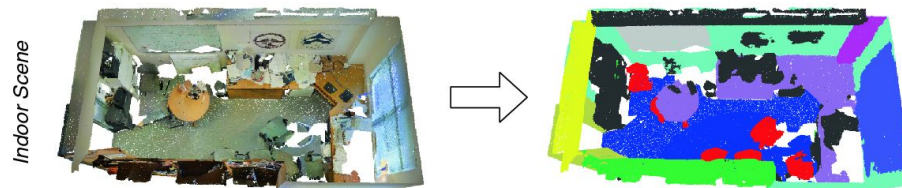
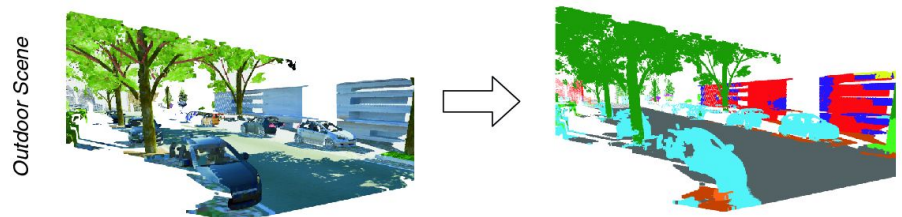


1.2 2D・3Dセグメンテーションについて

3Dセグメンテーション

対象: 点群データ(RGB情報持つ)

流れは2Dとほぼ同じ。



Input: 3D Point Cloud

Output: Semantic Segmentation

3Dの世界で、直接**3Dセグメンテーション**をする手法も存在。

1.3 問題意識

3Dセグメンテーションの欠点

- ・3次元空間における演算は膨大で、計算時間が非常にかかる。実際に、「リアルタイム性」に関する論文などは非常に少ない。

- ・現在では高性能のスペックが必要。

- ・教師データを作るにはコストが増える。

- ・.....

ある「リアルタイム」+「点群セグメンテーション」に関する論文から取り出したデータ

	Processing Time (msec)			Speed (fps)
	CNN	kNN	Total	
RangeNet++ [7]	63.51	2.89	66.41	15 Hz
SalsaNet [1]	35.78	2.62	38.40	26 Hz
SalsaNext [Ours]	38.61	2.65	41.26	24 Hz

- ・比較しているモデルの数が少ない

- ・速度について、24FPSは「リアルタイム性」を持つですが、**60万円**ほどのNVIDIA Quadro RTX 6000 - 24GBカードを使用した

リアルタイムとは

語彙

コンピュータで、入力データの処理を即時行なうこと。

—精選版 日本国語大辞典

明確的な定義(速度の基準)

「リアルタイム」には明確的な基準がない！！

補足

人間の肉眼で捉える情報の速度は約**10~12FPS**(フレーム毎秒)
以上であれば、画面が連続しているように見える。

昔の映画:16FPS, 今の映画:24FPS

今回の演習では、16FPS以上であれば、「リアルタイム性」があると
する。

1.4 先行研究

実は

・「リアルタイムの2Dセマンティックセグメンテーション」についての論文・実装が多数存在している。

・RGB画像とDepth画像を用いて、2Dセマンティックセグメンテーションの精度を上げる論文も存在している。

有名な論文収録ウェブサイト
arxiv.orgにおいて、
「**Real-Time Semantic Segmentation**」を「タイトル」で検索
結果は66項目がある。

**Review on Indoor RGB-D
Semantic Segmentation with
Deep Convolutional Neural
Networks**

<https://arxiv.org/abs/2105.11925>

しかし、「2D セグメンテーションと深度情報を活用したリアルタイム 3D セマンティックセグメンテーション」のような考えの論文が見つからない。

1.4 先行研究(共同研究)

本研究テーマは、株式会社Tengun-label の岩澤 秀樹と議論し、アドバイスを頂いた上で決まったものである。

この研究は、岩澤 秀樹との「共同研究」である。

株式会社

Tengun-label: <https://www.tengun-label.com/>

個人プロフィール

<https://www.facebook.com/hiwa0715/>

1.5 演習の動機・目的

動機

- ・「自動運転」と「ロボティクス」が近年の話題になって、特にセマンティックセグメンテーションはその中に重要な役割を果たしている。
- ・どの領域においても、「xxがリアルタイムでできる」ということが実用性がある。

目的

「2D セグメンテーションと深度情報を活用したリアルタイム 3D セマンティックセグメンテーション」の実現。

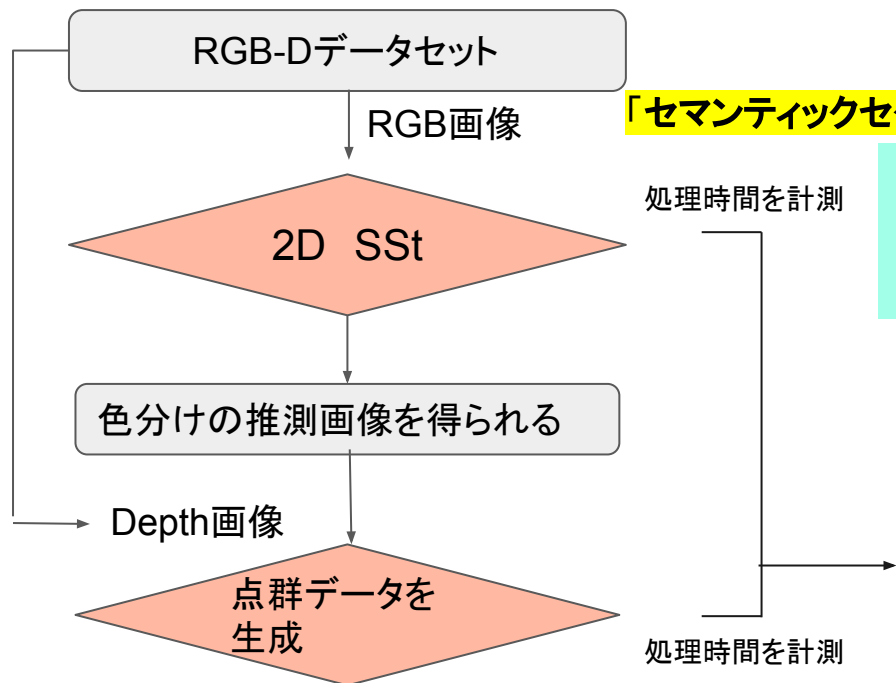
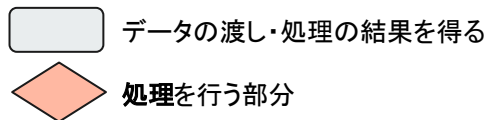
※特に、「リアルタイム性」を重視



2. 演習

2.1 演習方法

演習の流れ

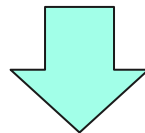


演習に用いたライブラリ

画像: numpy, cv2, PIL
SSt : numpy, torch
時間計測 : time, timeit, ctime

「セマンティックセグメンテーション」を「SSt」と略称

評価:すべての処理がリアルタイムで実行できるか



処理時間の合計値が16FPS以上であるかどうか

2.1 演習方法

環境

インターネット

AWS (Amazon Web Services) の
EC2を利用した

リモートデスク
トップ

仮想環境に接続して, 作業

Amazon Elastic Compute Cloud (EC2) は、Amazonが提供する計算資源を用いてアプリケーションを実行するサービスである。

仮想環境の性能(スペック)

- ・OS: Ubuntu 20.04
- ・CPU: Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz
- ・GPU: Nvidia Tesla T4

2.1 演習方法

カメラとは 3次元空間内の点を, 2次元画像上の点に対応させる装置である。



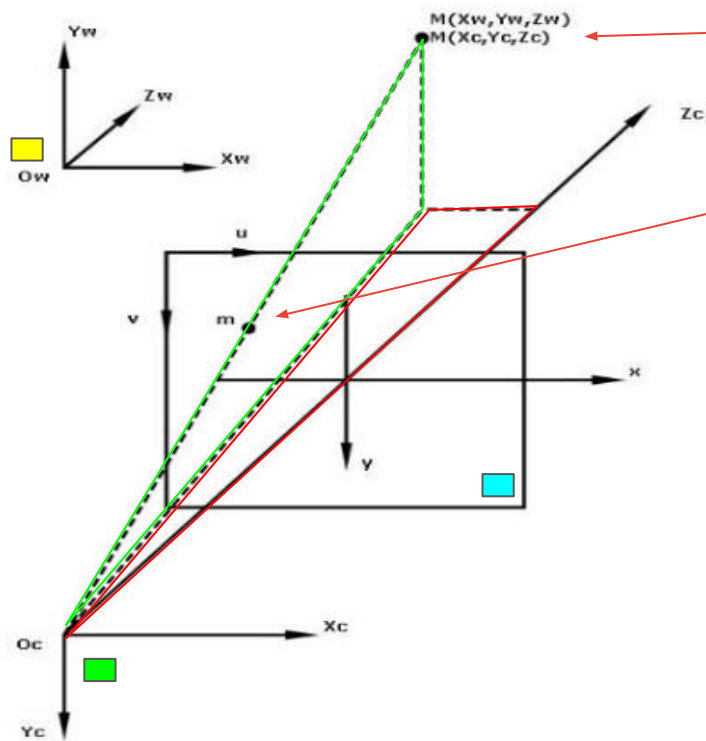
(カメラ)キャリブレーションとは

カメラモデルの内部パラメータ(焦点距離, 光学中心など)と外部パラメータ(カメラの位置)を推定する手法である。

「キャリブレーション済み」=> 内部・外部パラメータがわかる



焦点距離を得られる。



ある空間中の点 $M(X_w, Y_w, Z_w) = M(X_c, Y_c, Z_c)$ が
 (ワールド座標の原点とカメラ座標の原点が同じとする)
 画像平面に対応する点 $m(X_u, Y_v)$ との座標関係を考えよう。

緑の三角形を見ると、三角関係により

$$f / Z_c = Y_v / Y_w \quad (1)$$

Y_w : 求める値

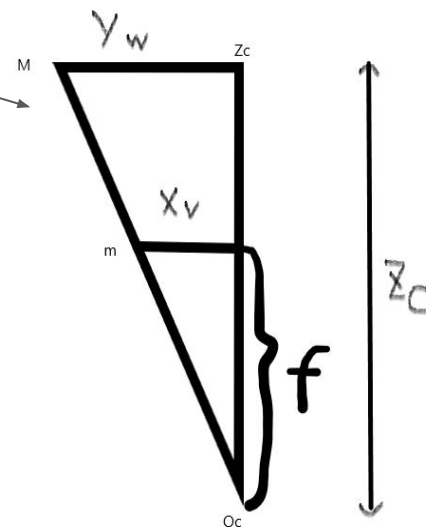
f : 焦点距離

(キャリブレーションで得られた値)

Z_c (Z_w) : Depthデータ(距離)

Y_v : 画像座標系におけるY座標

$$\Rightarrow Y_w = Y_v * Z_c / f \quad (2)$$



まず、三つの座標系がある

・ワールド座標系 : $O_w - X_w - Y_w - Z_w$

・カメラ座標系 : $O_c - X_c - Y_c - Z_c$

・画像座標系 : $x - y$ ($u - v$)

同じように赤の三角形において、

$$X_w = X_u * Z_c / f \quad (3)$$

$\Rightarrow M(X_w, Y_w, Z_w)$ 座標を求めることができる。

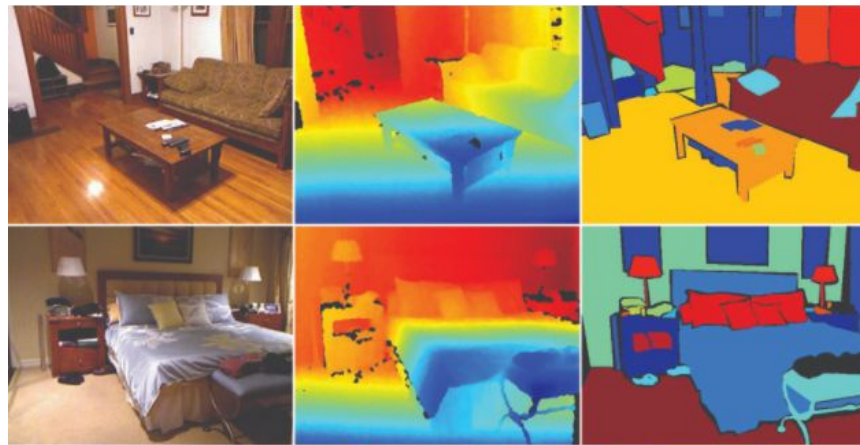
2.1 演習方法

データセット

オープンソースで公開された「**RGB-D**」画像のデータセットの中からニューヨーク大学の「**NYU Depth V2**」というデータセットを使用した。

https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

RGB画像 Depth画像 教師データ



「キャリブレーション済み」のデータセット

2.1 演習方法



演習に利用する2Dセマンティックセグメンテーションツール

Light-Weight RefineNet for Real-Time
Semantic Segmentation

<http://bmvc2018.org/contents/papers/0494.pdf>

<https://github.com/DrSleep/light-weight-refinenet>

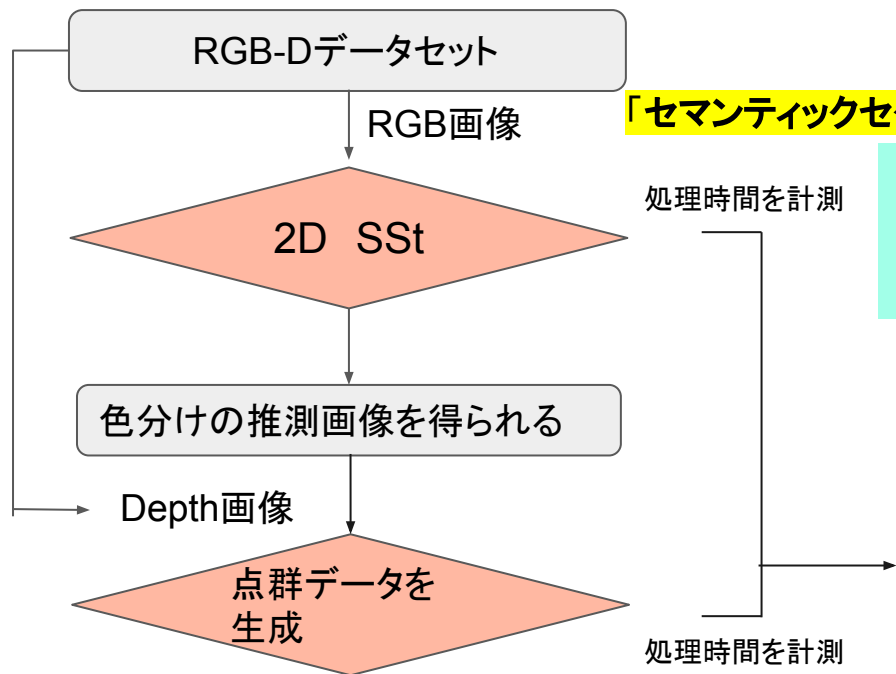
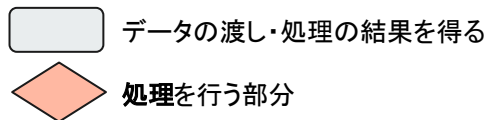
https://pytorch.org/hub/pytorch_vision_resnet/

2.1 演習方法(再掲)

演習に用いたライブラリ

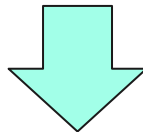
画像: numpy, cv2, PIL
SSt : numpy, torch
時間計測 : time, timeit, ctime

演習の流れ



「セマンティックセグメンテーション」を「SSt」と略称

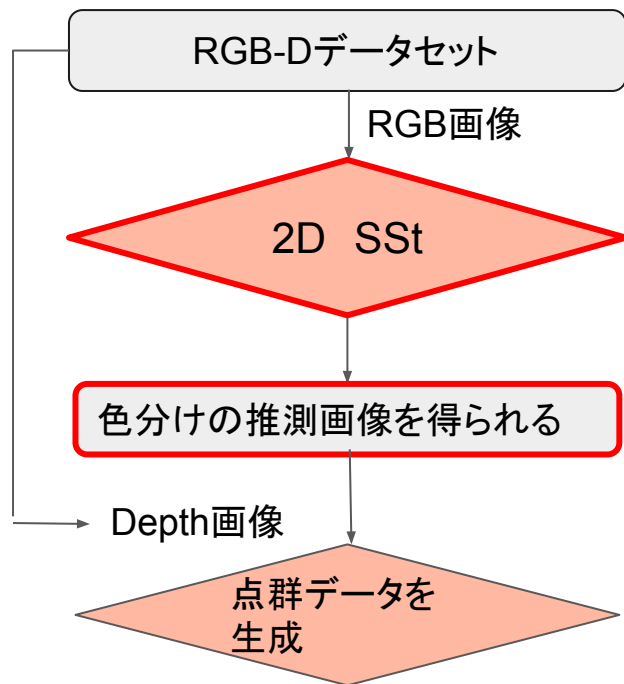
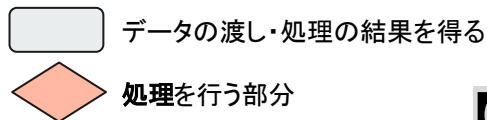
評価:すべての処理がリアルタイムで実行できるか



処理時間の合計値が16FPS以上であるかどうか

2.1 演習方法

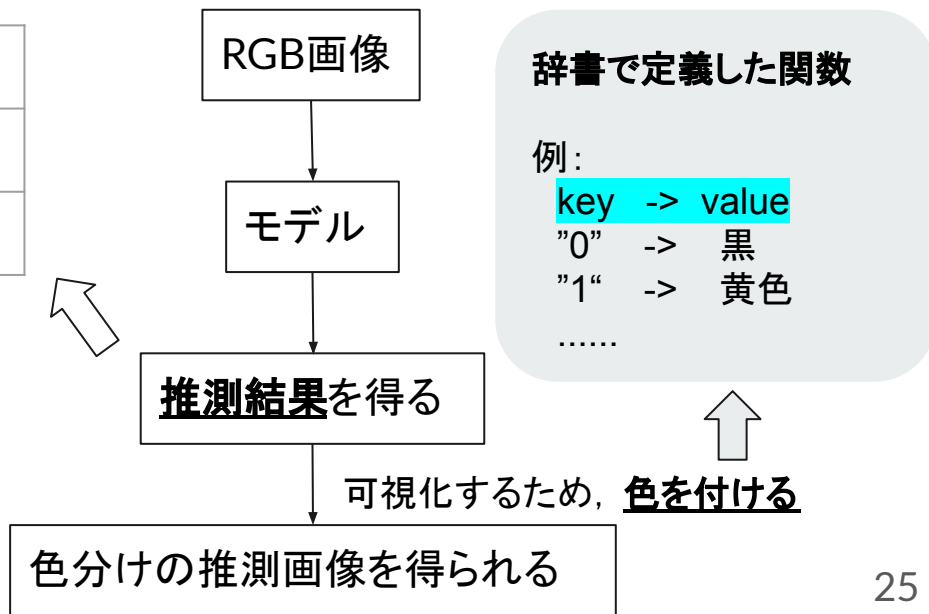
演習の流れ



ラベル番号で構成された画像

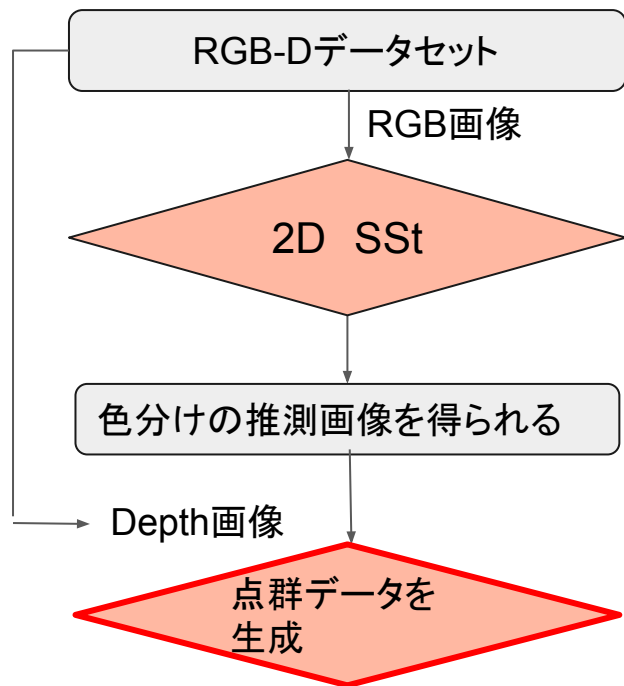
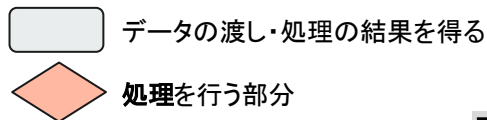
0	0	1
0	0	1
2	2	2

【2D SSt】と【色付け処理】



2.1 演習方法

演習の流れ

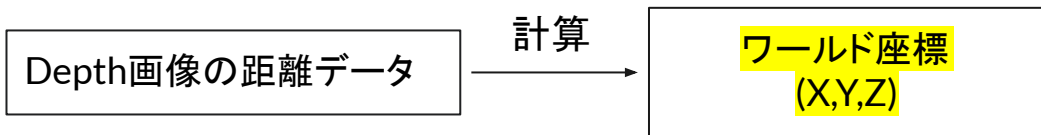


【点群データの生成】

点群データとは,

R, G, B, X, Y, Z 複数の属性を持つ点の集まり。

【色情報】は推測画像から得られる。



以下を参照

<https://github.com/gllmp/rgbd2pcl>

2.2 演習結果

【2D SSt】の時間計測

言語: Python

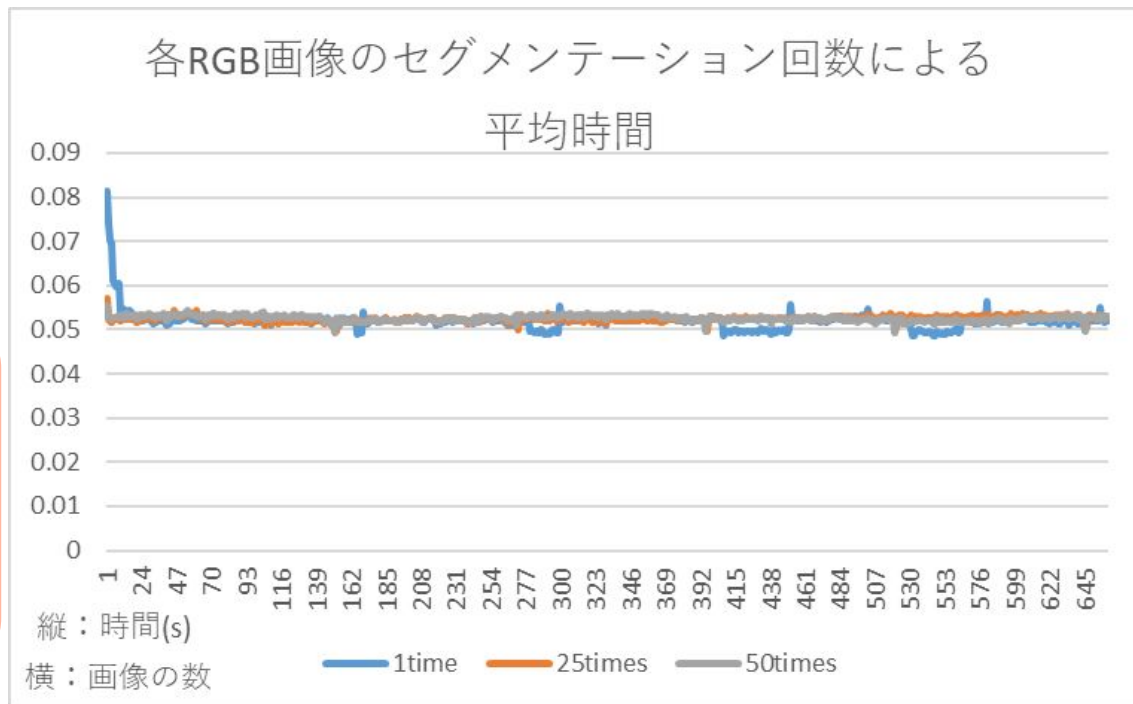
対象: 660枚画像

画像サイズ: 625*468

ライブラリ: numpy, time, torch

・1回の場合、最初に時間がかかったけれども、10枚目ぐらいから安定してきた。

・全体的に**約0.052s**に収まっている。



2.2 演習結果

【点群データを生成】の時間計測

最初に, Python環境で一枚画像の点群データを生成する時間の計測をしたが... (timeitライブラリを使った)

結果

```
%timeit -n 20 -r 3 get_points(depth,rgb)
```

```
1.32 s ± 6.45 ms per loop (mean ± std. dev. of 3 runs, 20 loops each)
```

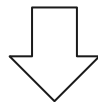
既に1s以上かかったので,「リアルタイム性」がない.....

Pythonの処理速度が遅い！

2.2 演習結果

【点群データを生成】の時間計測

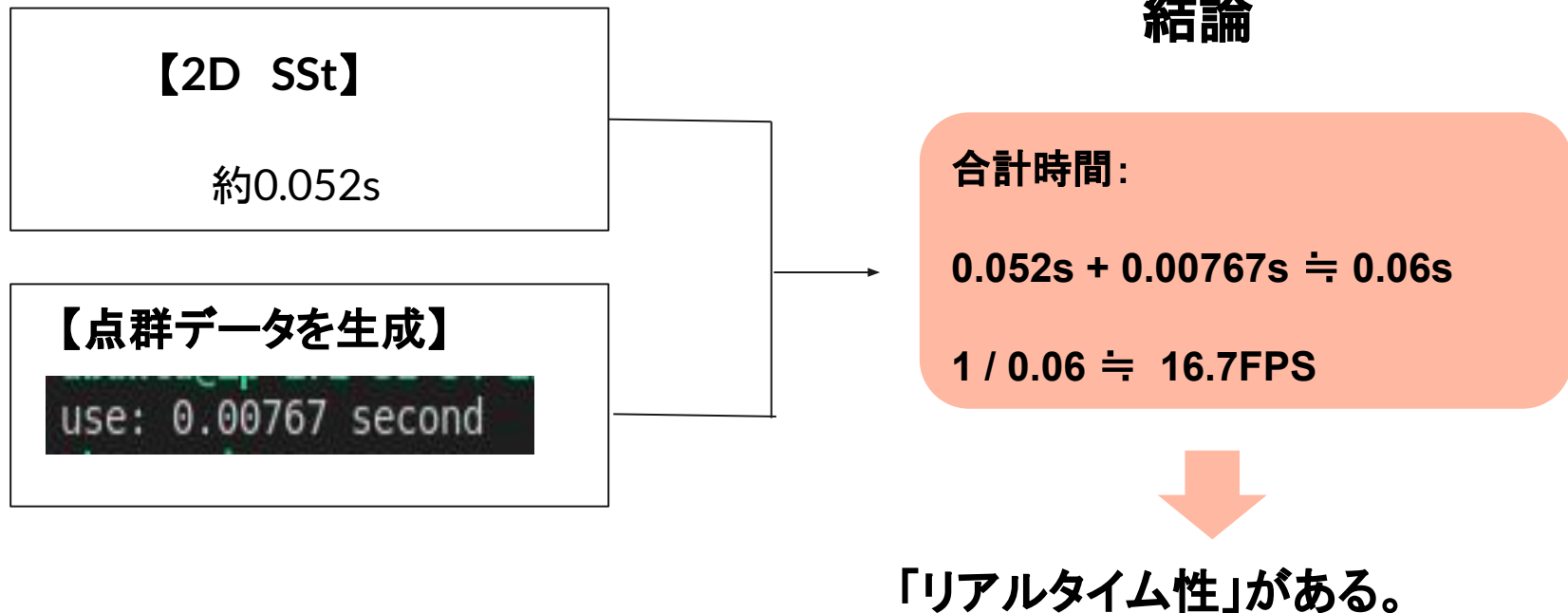
【点群データを生成】部分のコードをC++で書き直し、
同じデータを与えて、時間計測！！



結果

```
use: 0.00767 second
```

2.2 演習結果



2.2 演習結果

元画像

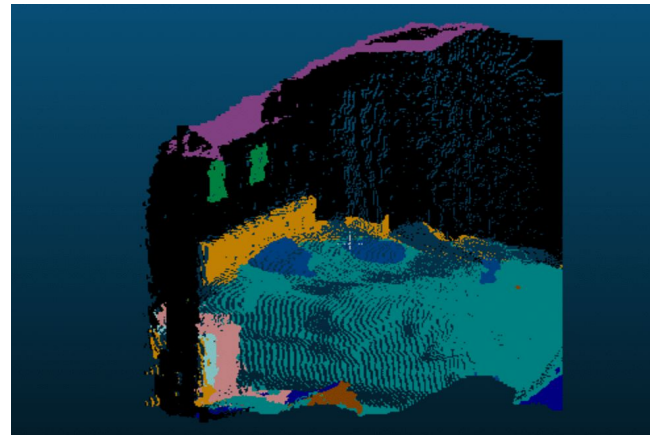


2Dセグメンテーション



Depth
データ

点群にした画像





3. 終わりに

3.1 まとめ



- 結果としては、ギリギリ**16FPS**に達成したが、満足できる結果ではない。
まだ、改善できることが多く見られる。
- 「**可視化の結果**」は、2Dセマンティックセグメンテーションの**予測精度**に関連している。

3.2 今後の課題・展望

今後の課題

- 処理速度の上昇
- 2D SStの精度上昇
- 「2D SSt」部分に対して、最初の不安定を解決
- キャリブレーションの精度向上

今後の展望

- 「ロボティクス」「自動運転」などの領域での応用。
e.g.) 即時衝突回避, 対象物の追従ロボット
- SLAM・VR・AR分野の応用
SLAM(スラム)とは、自己位置推定と環境地図作成を同時に行うことを言う
- 使用の目的に応じて、スマートフォン向けのアプリケーションを作る。