

YOLOv8による  
リアルタイム人物除去を組み込んだ  
TurtleBot3 + Raspberry Pi 4 環境用  
SLAMの実現

日本大学文理学部情報科学科-谷研究室

薮島理央 井澤大翔

# 目次

- 自律移動ロボット
- 演習背景
- 演習の目的
- 先行事例
- 演習環境
- 演習概要
- 結果
- 考察・今後の課題



# 自律移動ロボット



- ロボットが自律移動するには環境地図を作成し、自己位置推定をすることが必要なことがある
- 地図作成の手法の1つとして、**SLAM** (Simultaneous Localization and Mapping) が用いられている

# SLAM (Simultaneous Localization and Mapping)

自己位置推定と環境地図作成を同時に行う技術

## SLAMの活用事例

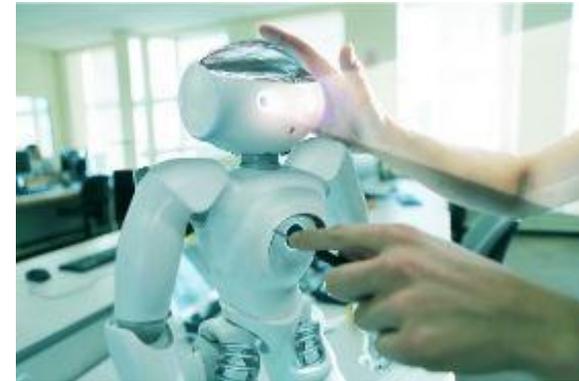
自動運転



AGV (無人配送車)



ロボット技術



# SLAM (Simultaneous Localization and Mapping)

## SLAM 地図作成の基本的な手順

- ① 周囲をセンサでスキャンし, 環境情報を取得
- ② 環境情報から目印となる点 (特徴点) を取得
- ③ 取得した特徴点と既存の地図データを比較
- ④ 自己位置推定
- ⑤ 地図を更新

地図作成



①~⑤をループ

# 目次

- 自律移動ロボット
- **演習背景**
- 演習の目的
- 先行事例
- 演習環境
- 演習概要
- 結果
- 考察・今後の課題



# 演習背景

- サービスロボットの社会実装の進展
  - 医療ロボット, 警備ロボット, 掃除ロボットなど
  - 人が常に存在する**動的環境**での運用が要求される
- 動的環境におけるSLAMの課題
  - 従来のSLAMは環境が**静的**であることを前提として設計されている
  - 人などの動的物体が存在すると地図の歪みが生じる
- 実運用での計算資源の制約
  - 消費電力・コストが高い
  - 計算資源が十分に確保できない場合が多い

# 目次

- 自律移動ロボット
- 演習背景
- **演習の目的**
- 先行事例
- 演習環境
- 演習概要
- 結果
- 考察・今後の課題



# 演習の目的

動的環境下・計算資源制約下におけるリアルタイムでの人物除去機能を搭載したSLAMシステムの構築

人物除去によるSLAM精度の向上

- YOLOv8で検出した人物を点群化し, 近傍点探索によりリアルタイムで地図上から除去する

実運用を想定し, 計算資源制約下で動作するかを検証する

- TurtleBot3に搭載しているRaspberry Pi 4 を使用

# 目次

- 自律移動ロボット
- 演習背景
- 演習の目的
- **先行事例**
- 演習環境
- 演習概要
- 結果
- 考察・今後の課題



# 先行事例

## 中央大2021：YOLO を用いた人除去処理による ICP-SLAM の性能向上

### 研究内容

- 画像 (YOLO) と距離情報 (Depth・LiDAR) の融合
- YOLOによる人物検出を用いて点群中の人物を除去
- クラスタリング + 近傍点探索による人物除去
- 動的物体の影響を低減し、ロバストな地図を構築

### 本演習との違い

- リアルタイム性の検証は行っていない
- 計算資源制約を前提とした検証ではない

# 目次

- 自律移動ロボット
- 演習背景
- 演習の目的
- 先行事例
- **演習環境**
- 演習概要
- 結果
- 考察・今後の課題



# 演習環境

## ロボット構成

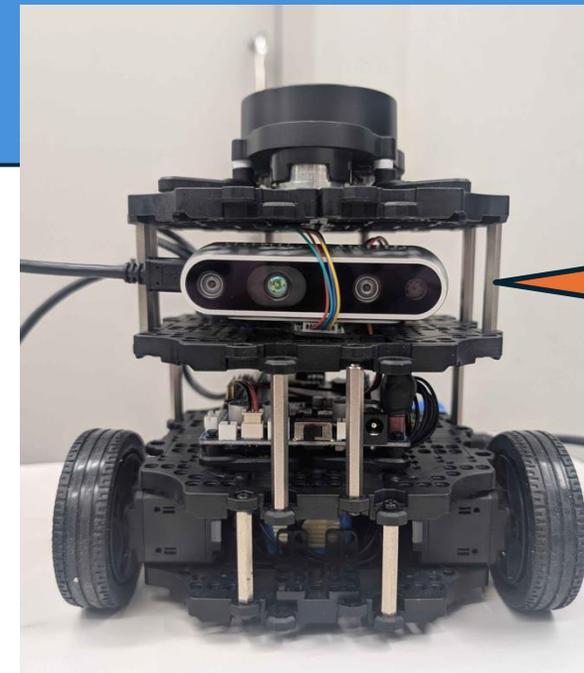
- TurtleBot3 (Burger)
- Raspberry Pi 4B (8GB)
- LiDAR:LSD-02
- Depth Camera:RealSense D435

## PCスペック

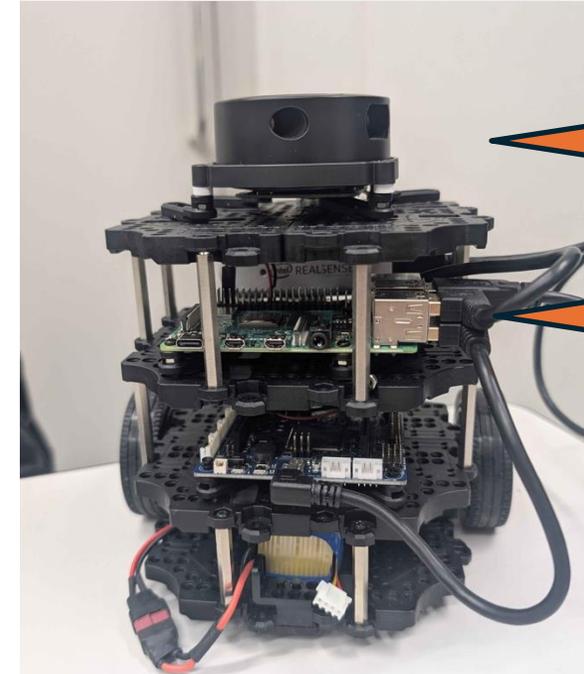
- CPU : Intel(R) Core(TM) i7 – 13620H 2.40 GHz
- RAM : 32 GB
- GPU : NVIDIA GeForce RTX 4050 Laptop

## ソフトウェア

- Ubuntu 22.04 LTS
- ROS 2 Humble
- YOLOv8



RealSense



LiDAR

Raspberry Pi 4

# TurtleBot 3

- ROS入門のためのオープンソースの移動ロボットプラットフォーム
- 2D LiDARとシングルボードコンピューターであるRaspberry Piが搭載
- Raspberry Pi上のUbuntuでROSが動作
  - ROSがLiDARやモーターを制御



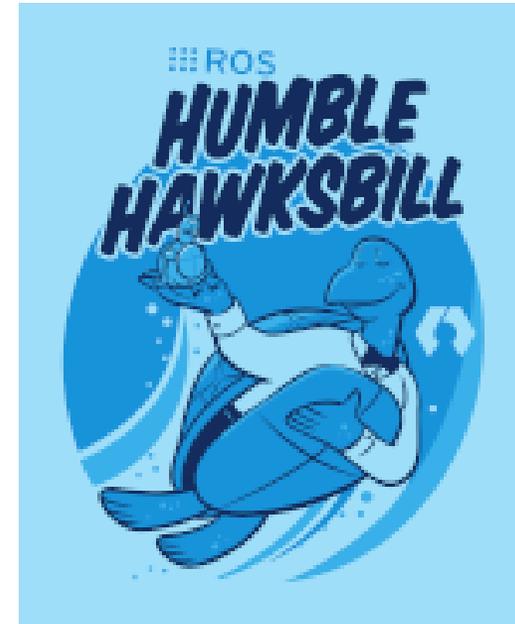
参考 : <https://e-shop.robotis.co.jp/product.php?id=352>

# ROS2 Humble

ロボット開発に用いるオープンソースミドルウェア  
「ROS2(Robot Operating System 2)」の長期サポート版

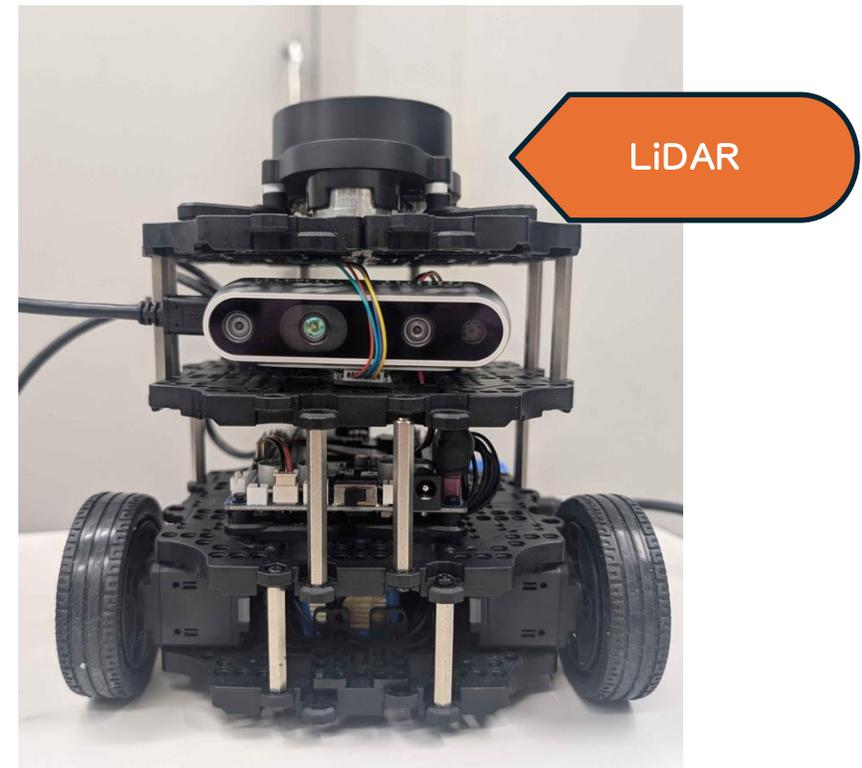
## ROS2 Humble採用理由

- TurtleBot3 が公式に対応
- Ubuntu 22.04 LTSに対応



# LiDAR (Light detection and ranging)

レーザー光を照射し、物体に反射して戻ってくるまでの時間から距離を測定するセンサ

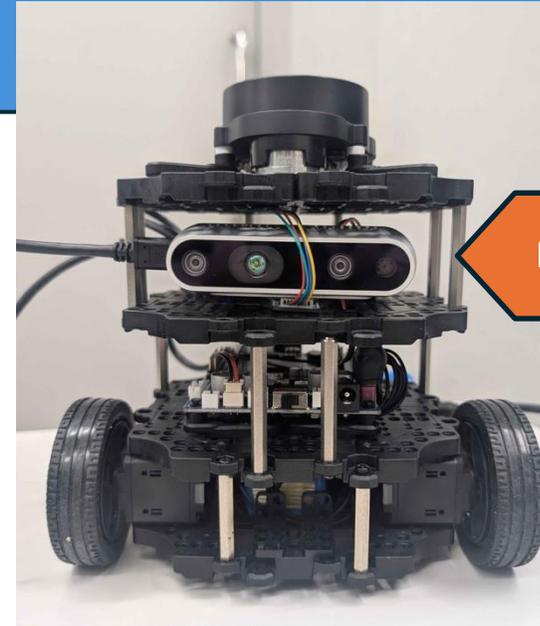


# RealSense D435

- RGB画像と深度画像 (Depth) を取得可能なデバイス
- Intelから, ROSパッケージが提供



画像引用: [https://jp.rs-online.com/web/p/depth-cameras/1720981?srsId=AfmBOoqKpQiWp20DrdhUOK2fAUuqyWTIIvfHuyT\\_u362ke69IG3nqPKK](https://jp.rs-online.com/web/p/depth-cameras/1720981?srsId=AfmBOoqKpQiWp20DrdhUOK2fAUuqyWTIIvfHuyT_u362ke69IG3nqPKK)



RealSense



画像引用: <http://stereo.jpn.org/jpn/stphmkr/makedm/index.html>

# YOLOv8

- Ultralytics社が公開した「物体検出手法YOLO」シリーズのモデル
- 物体検出に加え, セグメンテーションや分類タスクにも対応
- 精度と処理速度のバランスが最適化されており, リアルタイム物体検出に適したモデル

## YOLOv8 採用理由

- 本演習の目的のリアルタイム物体検出に適している
- ROS2 Humble × YOLOv8の事例が既に論文やブログで報告されている



# 目次

- 自律移動ロボット
- 演習背景
- 演習の目的
- 先行事例
- 演習環境
- **演習概要**
- 結果
- 考察・今後の課題



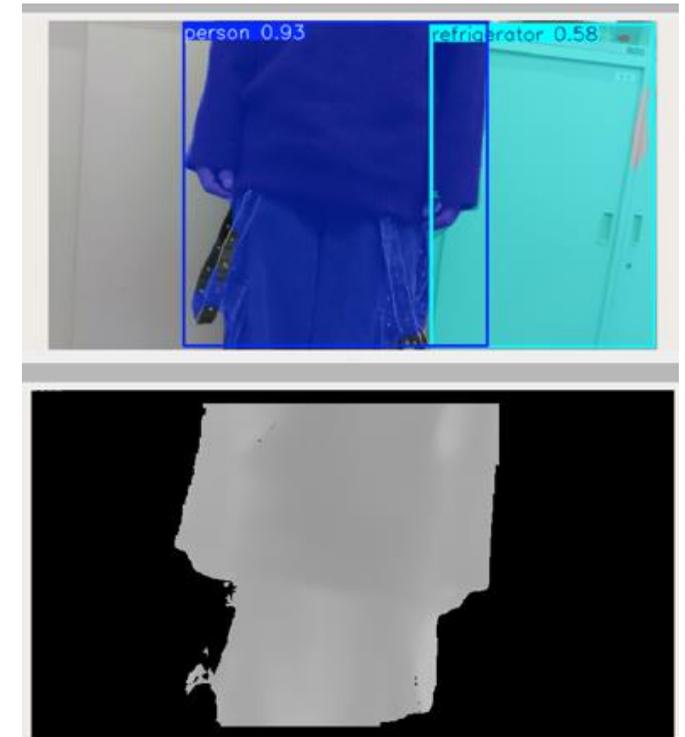
# 演習概要

## マスク処理の流れ

- RealSense から RGB/Depth 画像を取得
- RGB 画像に対し YOLOv8 で人物検出
- 検出領域をDepth画像に対応付け, 人物点群を生成
- 人物点群をSLAM 入力から除去

## 計算構成

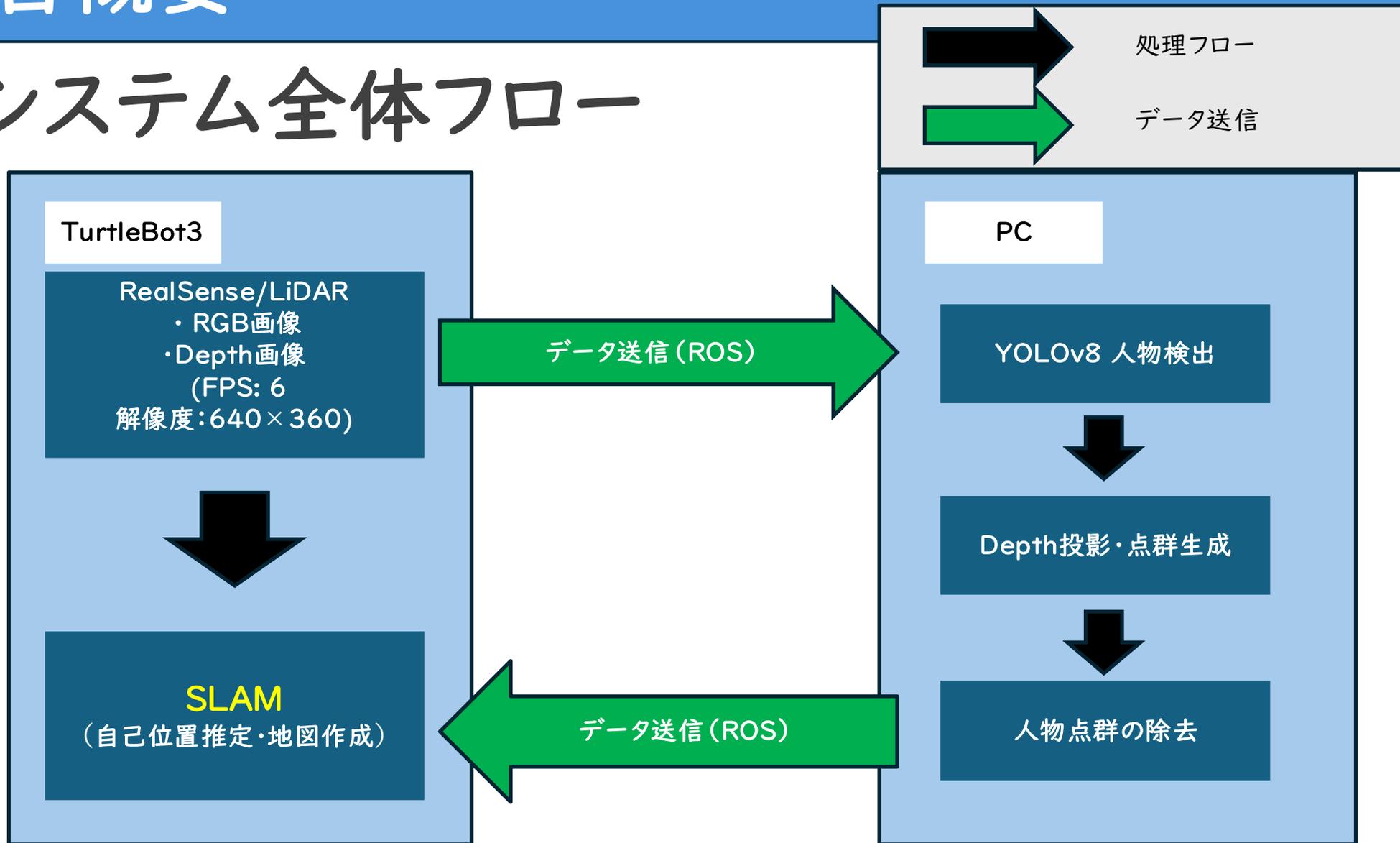
- 人物検出(YOLOv8) : PC (GPU) 側で実行
- SLAM : TurtleBot3側で実行  
→ 処理を分離した



✓ YOLO8での検出と, 人物領域を抽出したもの

# 演習概要

## システム全体フロー



## 演習方法

実験場所の地図安定後に、

- 人物除去なしの通常SLAM
- YOLOv8を用いた人物除去SLAM

を行うことで人物が地図にどう影響するかを確認する

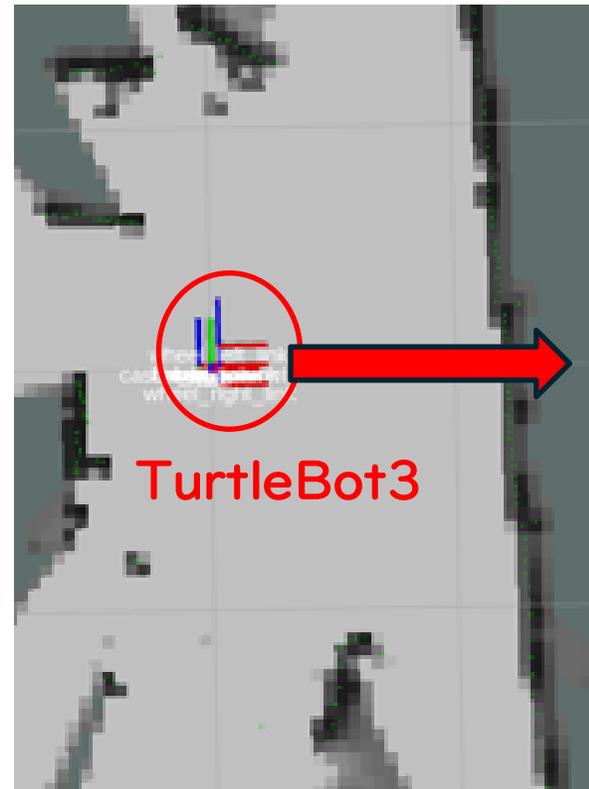
条件として、

- 人物が目の前で立ち止まる場合
  - 人物が複数回通過する場合
- の二種類のパターンで比較する

## 評価方法

実験で得られた地図を視覚化して比較する

安定した状態の地図



# 目次

- 自律移動ロボット
- 演習背景
- 演習の目的
- 先行事例
- 演習環境
- 演習概要
- **結果**
- 考察・今後の課題



# 結果

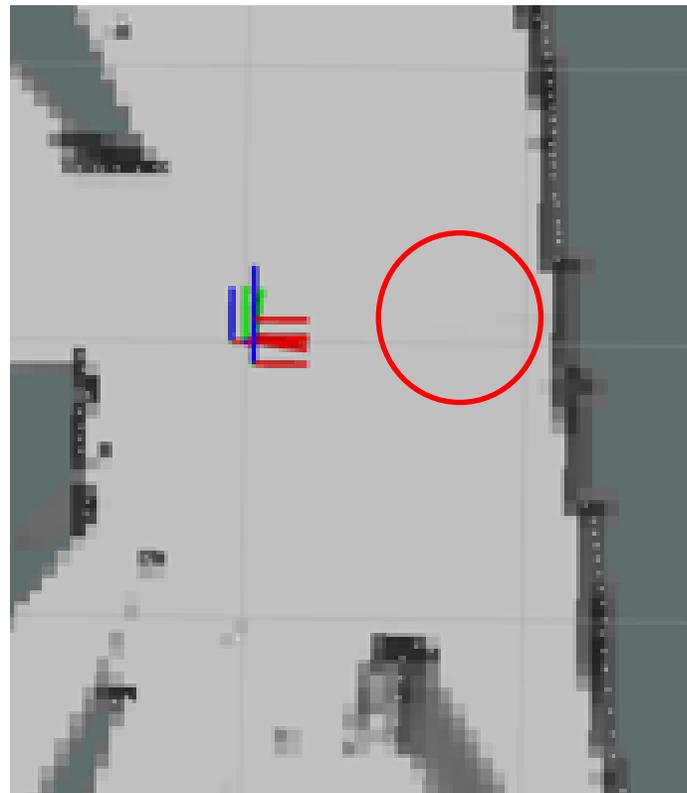
## 人物が目の前で立ち止まる場合

通常SLAM (YOLOv8なし)



人物点群

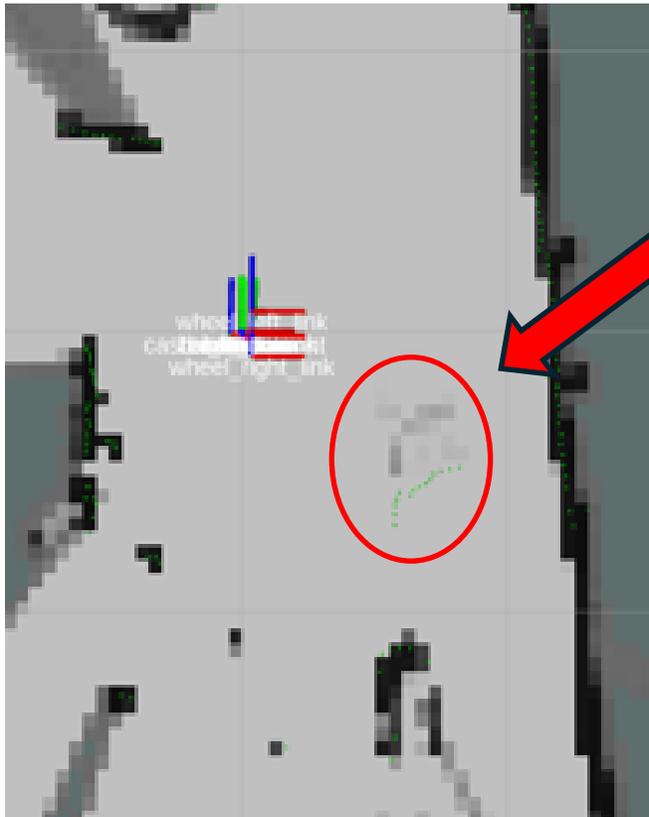
人物除去SLAM (YOLOv8あり)



# 結果

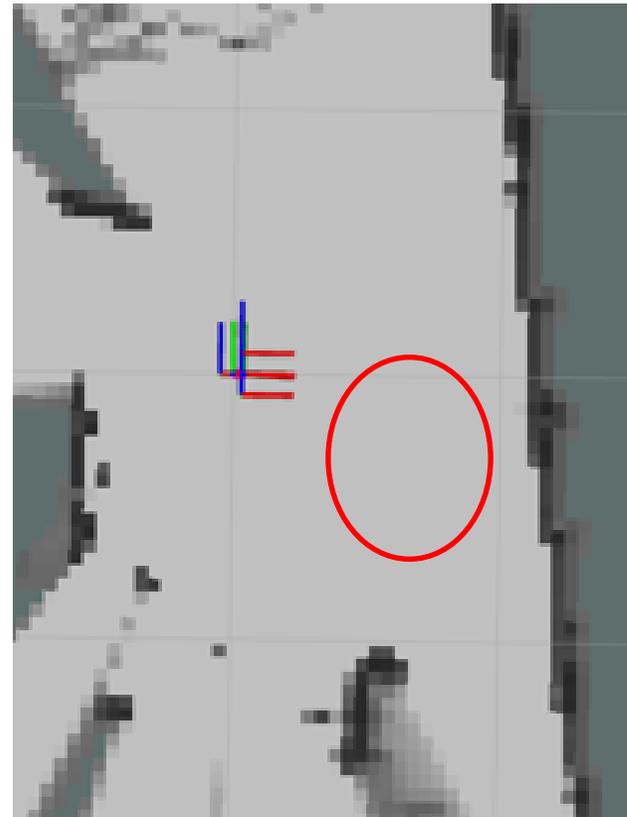
## 人物が複数回通過する場合

通常SLAM (YOLOv8なし)

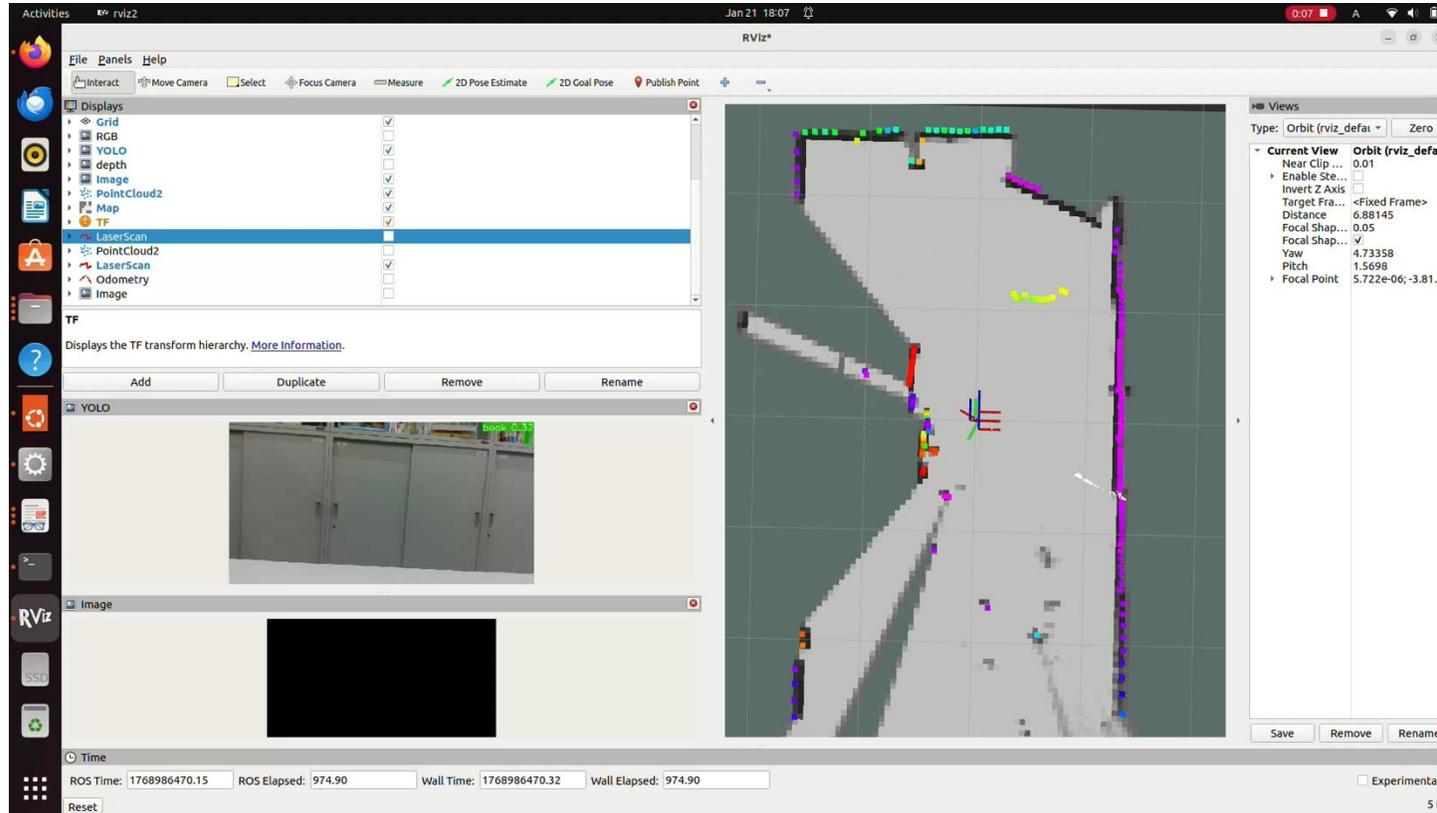


人物点群

人物除去SLAM (YOLOv8あり)



# 結果



- カラーの点群：LiDARから受け取った生の点群
- 白い細かな点群：検出した人物点群
- 白の点群：人物除去処理後の点群

# 結果

- 人物除去SLAMは地図精度の低下を抑制
- 動的環境でもSLAM精度の向上を確認

 人物の歩く速度に制限があった

# 目次

- 自律移動ロボット
- 演習背景
- 演習の目的
- 先行事例
- 演習環境
- 演習概要
- 結果
- 考察・今後の課題



# 考察・今後の課題

## ■ 処理性能と通信遅延の影響

- Raspberry Pi 4 では YOLOv8 推論と点群マスク処理の負荷が大きく、RealSense は FPS $\approx$ 6・解像度 640 $\times$ 360 が上限
- 人物検出 $\rightarrow$ マスク処理 $\rightarrow$ SLAM反映が地図更新に追従できず、人物位置の時間ずれにより地図へ誤反映が発生

- Raspberry Pi 5 などの高性能計算環境や、Ubuntu バージョンの更新などによる処理能力向上により、人物除去処理を含めた SLAM のリアルタイム性の向上が期待される