

自律走行ロボットが取得したデータからの リアルタイムモニタリング

日本大学文理学部情報科学科 卒業演習発表(令和6年2月7日)

谷聖一研究室 川口聖人

目次

- 1 はじめに

- 1.1 背景
- 1.2 動機・目的
- 1.3 先行事例
- 1.4 演習概要

- 2 演習内容

- 2.1 演習内容
- 2.2 演習環境
- 2.3 ロボットの自律走行
- 2.4 RealSense
- 2.5 点群データ処理
- 2.6 可視化

- 3 おわりに

目次

- 1 はじめに

- 1.1 背景
- 1.2 動機・目的
- 1.3 先行事例
- 1.4 演習概要

- 2 演習内容

- 2.1 演習内容
- 2.2 演習環境
- 2.3 ロボットの自律走行
- 2.4 RealSense
- 2.5 点群データ処理
- 2.6 可視化

- 3 おわりに

1.1 背景

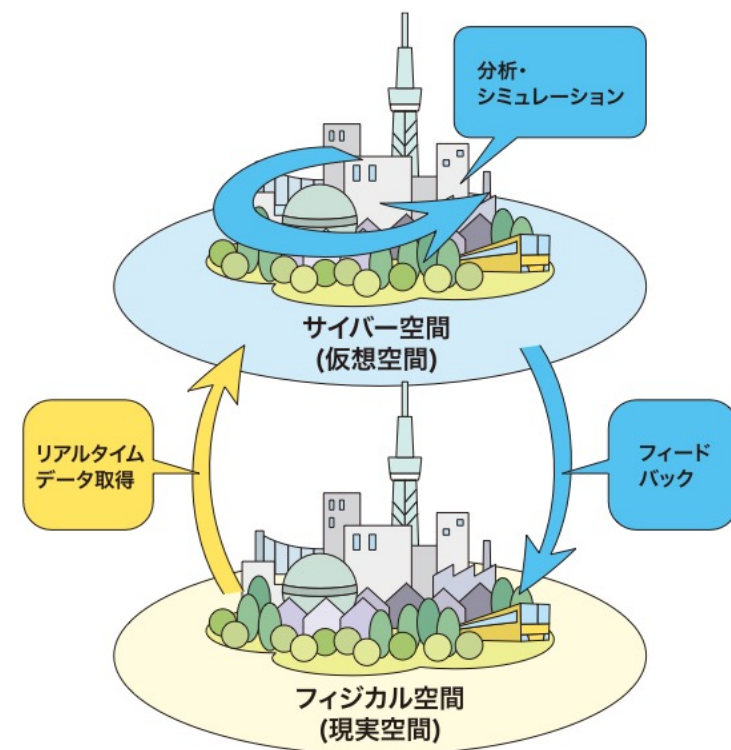
- 仮想空間と現実空間を高度に融合させたシステムにより、経済発展と社会的課題の解決を両立する、人間中心の社会 Society 5.0を実現させる基盤



- IoT等の技術を活用して現実空間の情報を取得し、仮想空間内に現実空間の環境を再現する、デジタルツインが注目

1.1.1 デジタルツインとは

- 2002年にマイケルグリーブスによって広く提唱された概念
- 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



画像引用： https://www.soumu.go.jp/hakusho-kids/use/economy/economy_11.html

デジタルツインの現状に関する調査研究の請負成果 報告書，研究開発戦略センター
https://www.soumu.go.jp/johotsusintokei/linkdata/r03_06_houkoku.pdf

1.1.1 デジタルツインとは

- デジタルツインによるメリット
 - 現実空間の状態をリアルタイムモニタリング
 - デジタルアーカイブ化による保守性の向上
 - 過去のデータと現在のデータの比較し変化の計測
 - 仮想空間上でのシミュレーションによるコストの削減
 - 町の浸水シミュレーション
 - AIの強化学習

1.1.1 デジタルツインの例

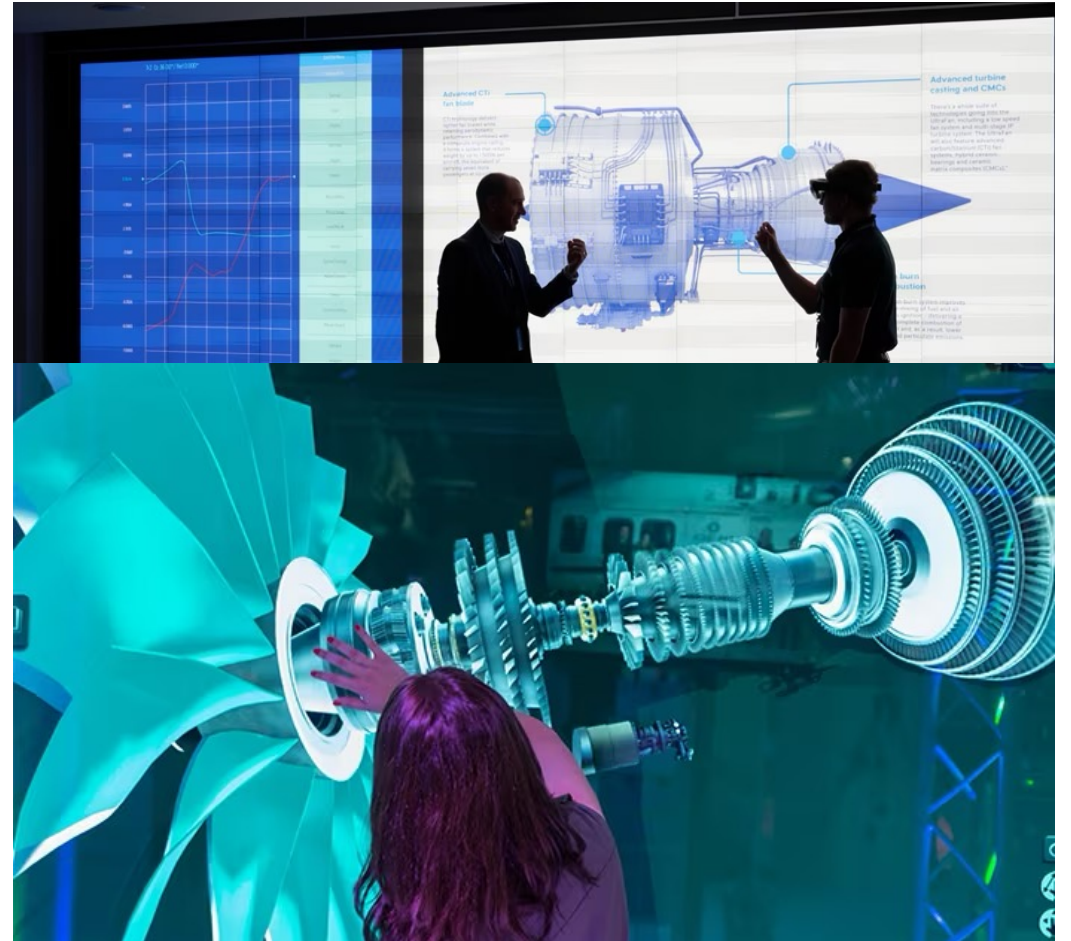
- リアルタイムモニタリング
 - 現実空間の状態を，IoTやセンサーなどの技術を用いて，リアルタイムでデジタルツイン上に反映し，監視・分析する技術

1.1.2 デジタルツインの例

- 狭義のデジタルツインは現実空間と仮想空間のリアルタイムかつ双方向の情報交換により現状の分析と将来予測の機会を与える動的なモデル
 - 例 ロールス・ロイス社の旅客機用ジェットエンジンの開発・保守
- 広義では、データのやり取りを含まない静的な3Dモデル等も含む
 - 例 PLATEAU

1.1.2 デジタルツインの例

- ロールス・ロイス社：旅客機用ジェットエンジン
 - 旅客機用ジェットエンジンをデジタルツイン上に再現し、現実空間からのリアルタイムなセンサーデータを元に劣化情報をシミュレート
 - 適切なメンテナンスを施すことで費用の削減



画像引用: <https://www.rolls-royce.com/innovation/digital/digital-twin.aspx>

<https://www.rolls-royce.com/media/our-stories/discover/2019/how-digital-twin-technology-can-enhance-aviation.aspx>

1.1.2 デジタルツインの例

● PLATEAU

- 国土交通省が様々なプレイヤーと連携して推進する、日本全国の都市デジタルツイン実現プロジェクト



1.1.2 デジタルツインの例

- 株式会社マクニカ

- 物流倉庫を仮想空間上に再現し、仮想空間上で、自律走行ロボットのAIの強化学習

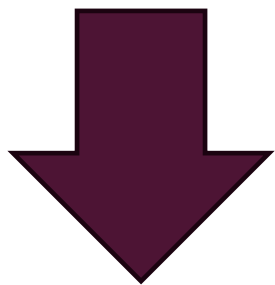


1.1.3 デジタルツインで期待されている技術要素

●IoT 移動ロボット 画像認識 物理シミュレーター

1.2 動機・目的

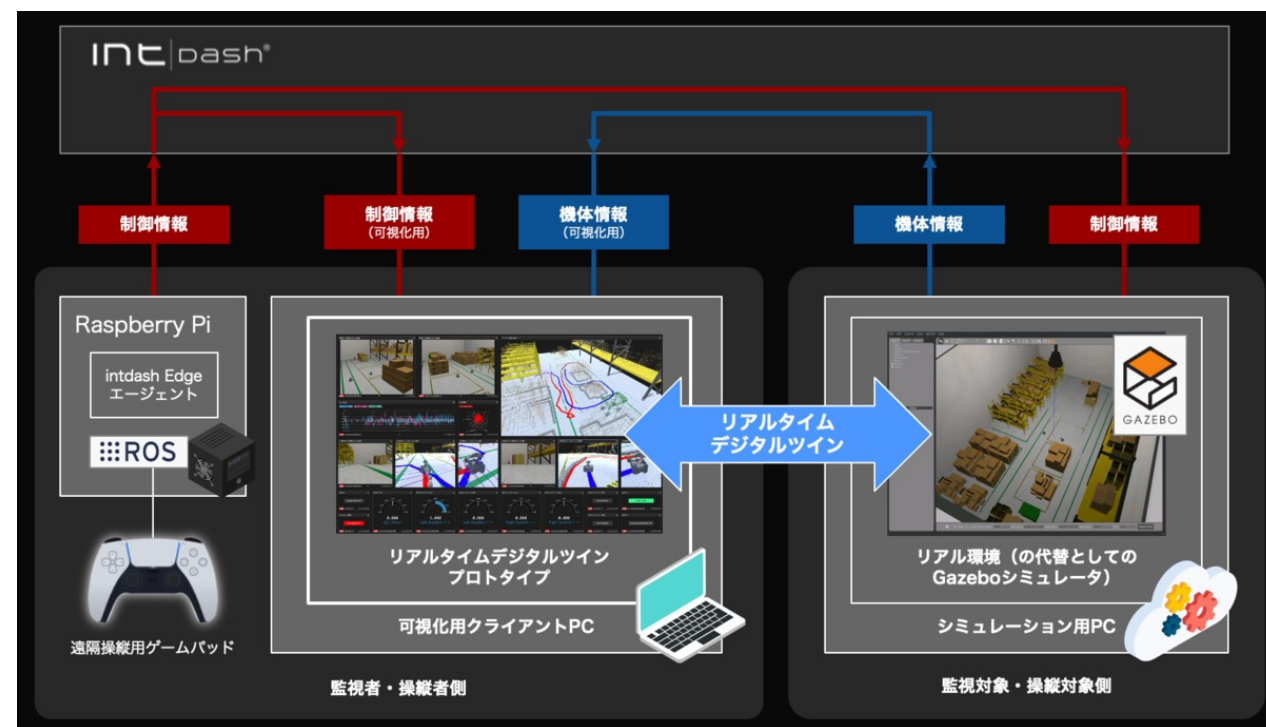
- PLATEAUのような静的な3Dモデルとリアルタイムなセンサーデータを連携し、動的なモデルとして運用するための足がかり



- 自律走行ロボットが取得したデータからの点群データとしてのリアルタイムな可視化

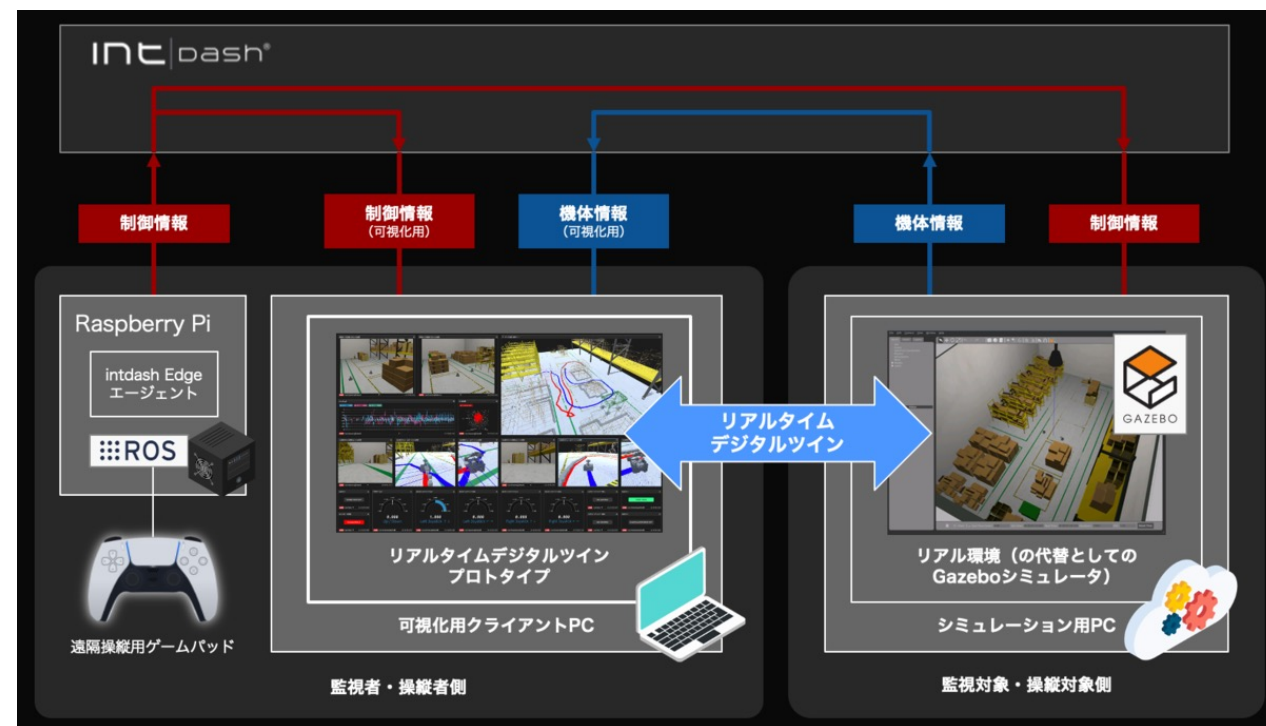
1.3 先行事例

- 株式会社アプトポッド
 - 物流倉庫を仮想空間上に再現
 - 現実世界で動くロボットの情報と障害物を仮想空間上でリアルタイムモニタリング



1.3 先行事例との相違点

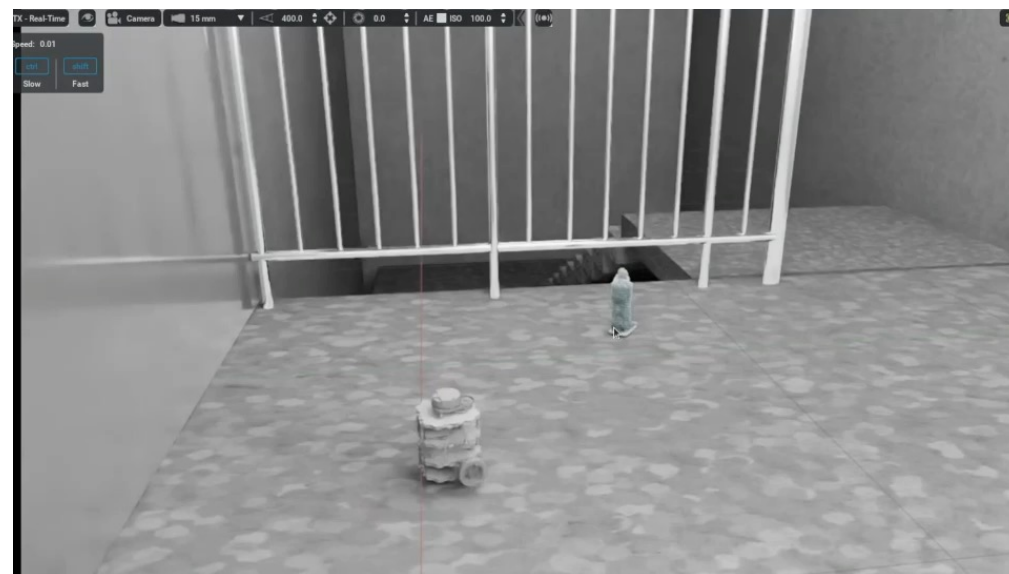
- 使用した物理シミュレーター
 - Unity → Isaac Sim
- 可視化する物
 - 2次元色無し点群→3次元色付き点群
(x,y) (x,y,z,r,g,b)



1.4 演習概要

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化

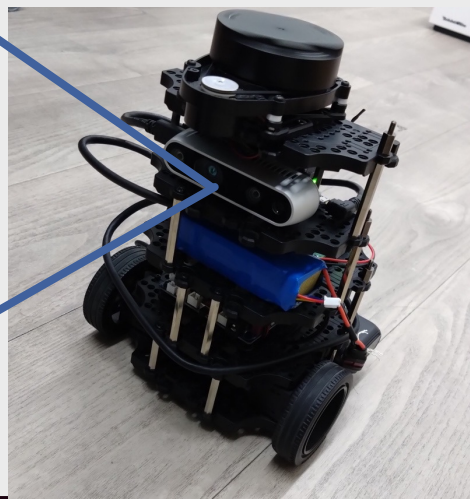
- 仮想空間上に3次元点群データとして可視化するメリット
 - 自由な視点からのモニタリングが可能



1.4 演習概要

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化

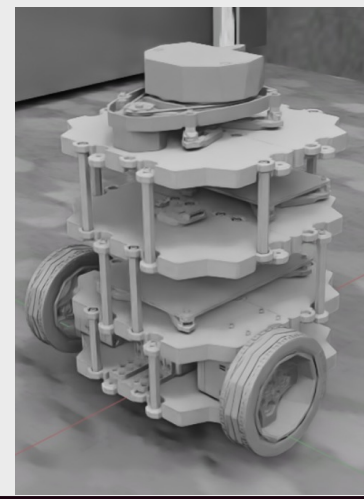
現実空間



センサー
データ

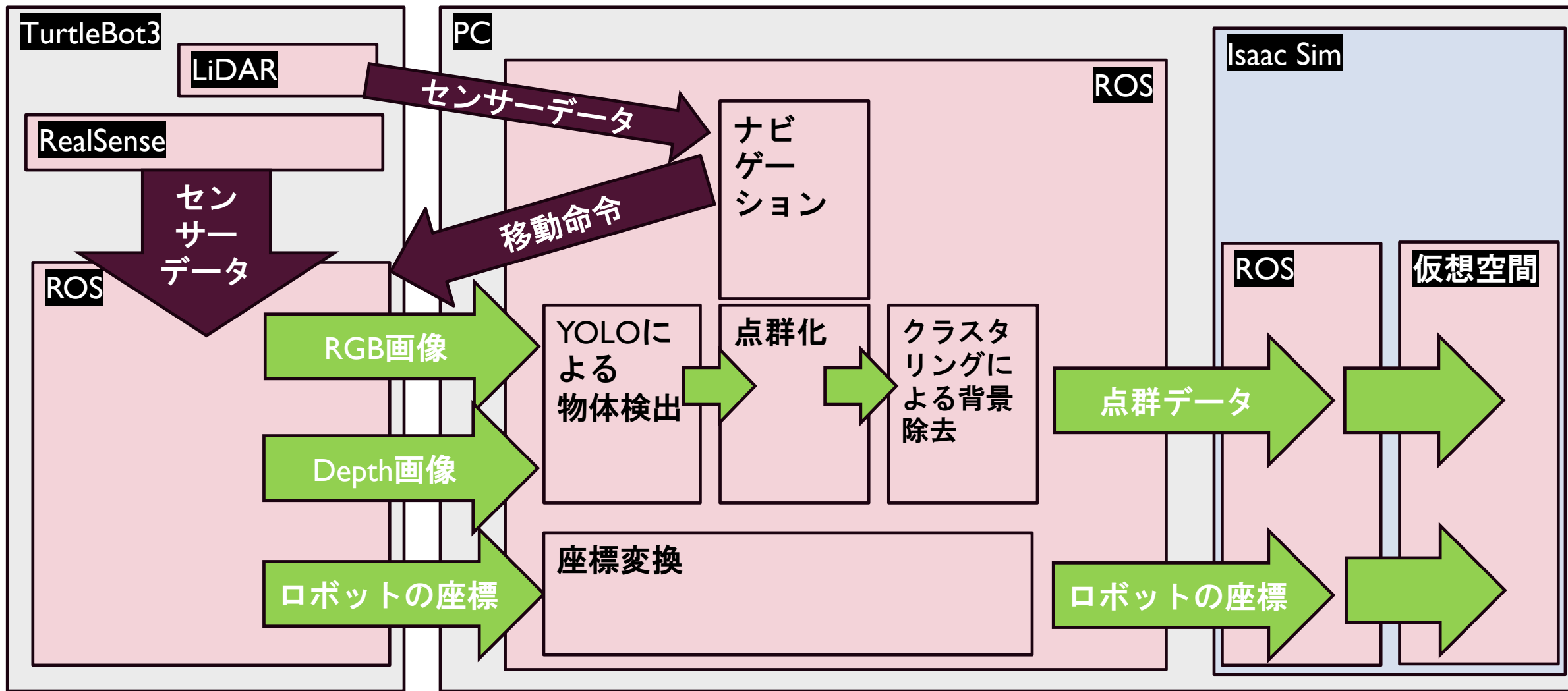


仮想空間

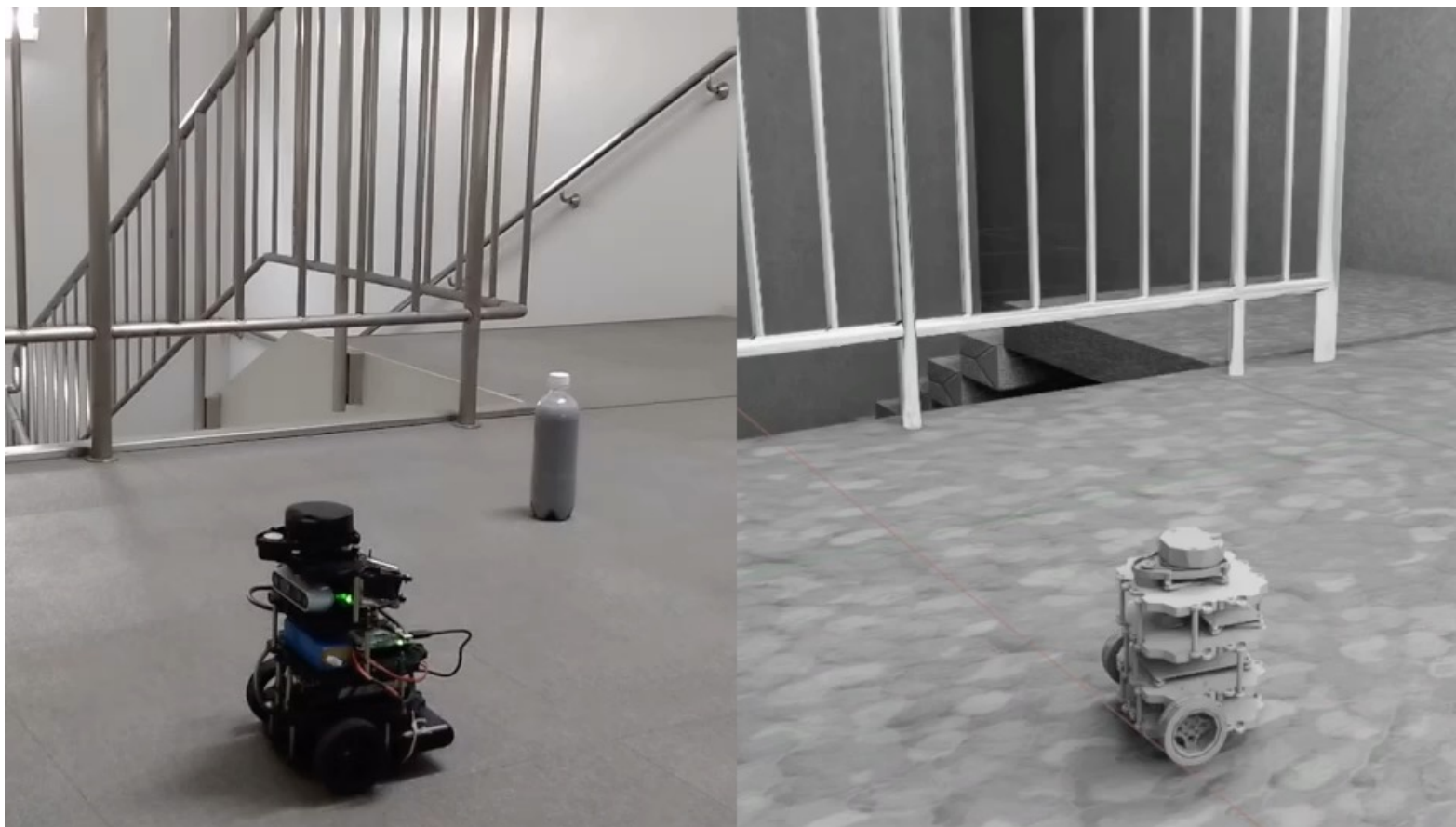


全体構成とデータの流れ

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化



1.4 演習概要



目次

- 1 はじめに

- 1.1 背景
- 1.2 動機・目的
- 1.3 先行事例
- 1.4 演習概要

- 2 演習内容

- 2.1 演習内容
- 2.2 演習環境
- 2.3 ロボットの自律走行
- 2.4 RealSense
- 2.5 点群データ処理
- 2.6 可視化

- 3 おわりに

2.1 演習内容

- 本演習では，以下の2点を実施
 - 1.TurtleBot3にRealSenseを搭載し，
室内を自律走行しながら撮影
 - 2.TurtleBot3の走行の様子と，撮影した物体を
事前に作成した部屋の仮想空間上に再現

2.2 演習環境

- TurtleBot3

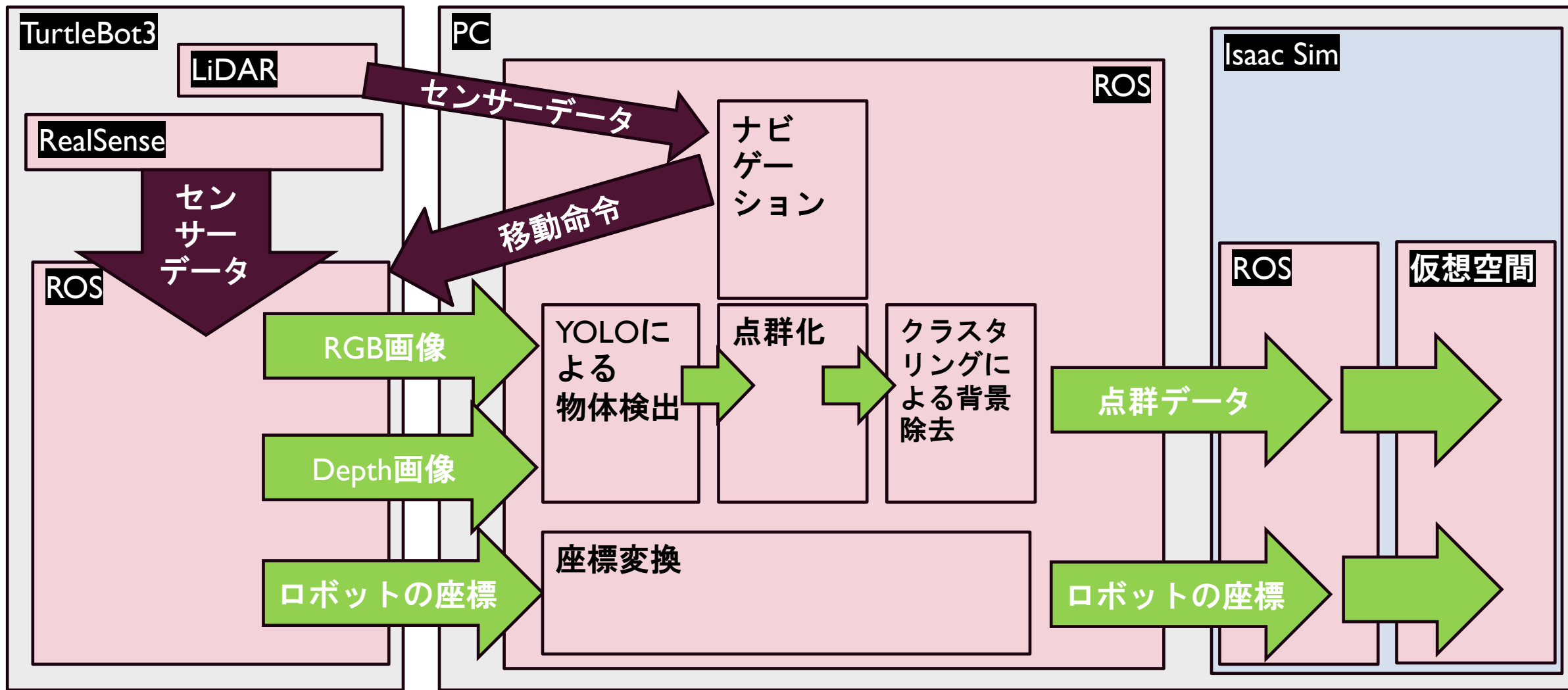
- SBC : Raspberry Pi 4B 8GB
- LiDAR : LSD-01
- Depth Camera : RealSense D435

- PC

- CPU : Intel® Core™ i7-11800H
- RAM : 32GB
- GPU : RTX 3070 RTX Laptop

全体構成とデータの流れ

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化



2.3 ロボットの自律走行

ROS (Robot Operating System)

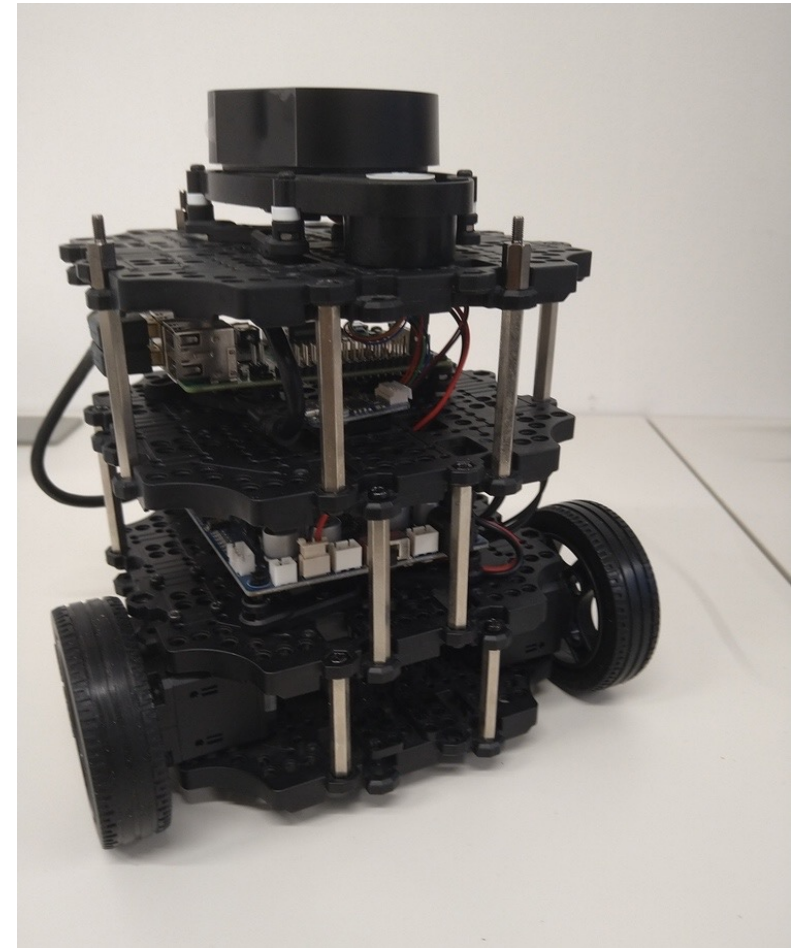
- ロボット開発に用いるオープンソースソフトウェア・ミドルウェア
- 既開発された汎用性の高いパッケージが配布されている



2.3 ロボットの自律走行

TurtleBot3

- ROS入門のためのオープンソースの移動ロボットのオープンプラットフォーム
- 2D LiDARとシングルボードコンピュータ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

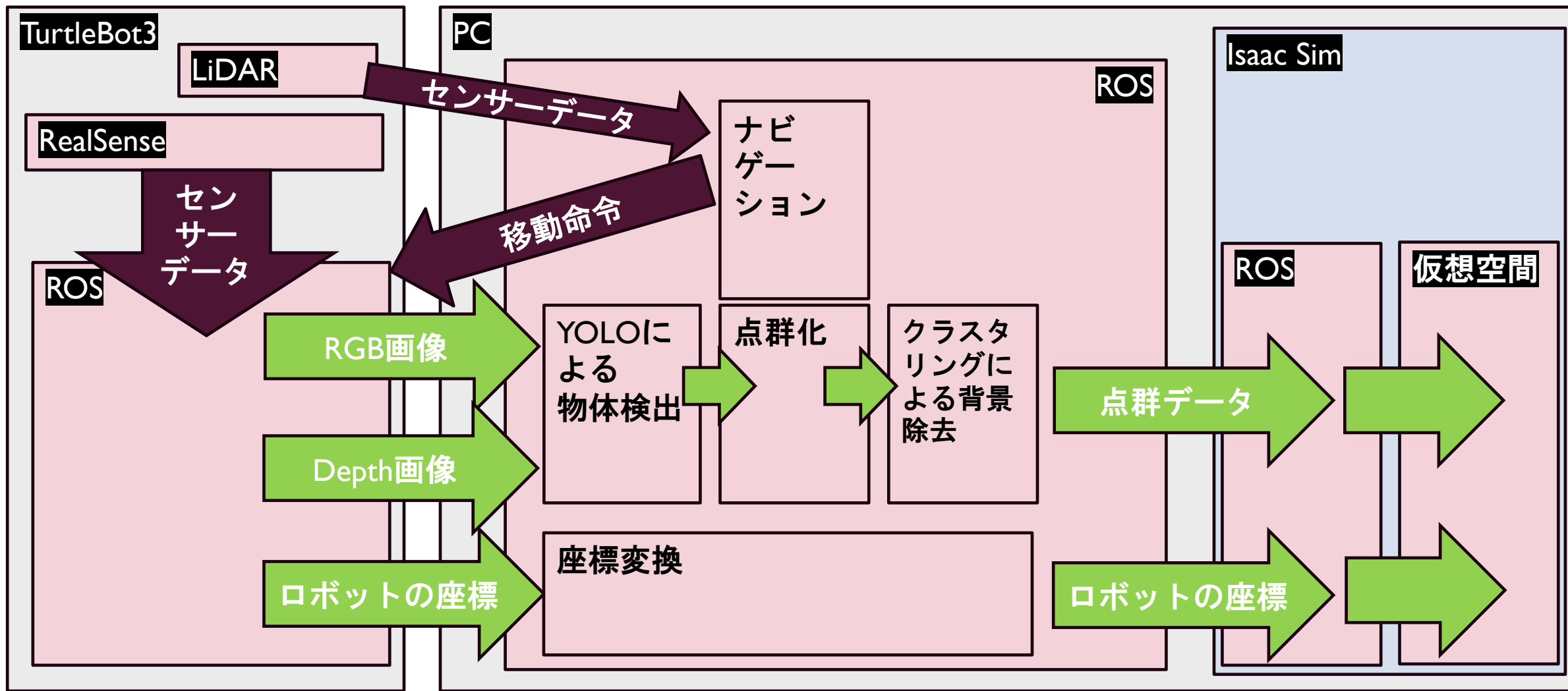


2.3 ロボットの自律走行

- 既存のROS パッケージを用いて，ロボットの自立走行を実行
 - Navigation 2 , SLAM Toolbox

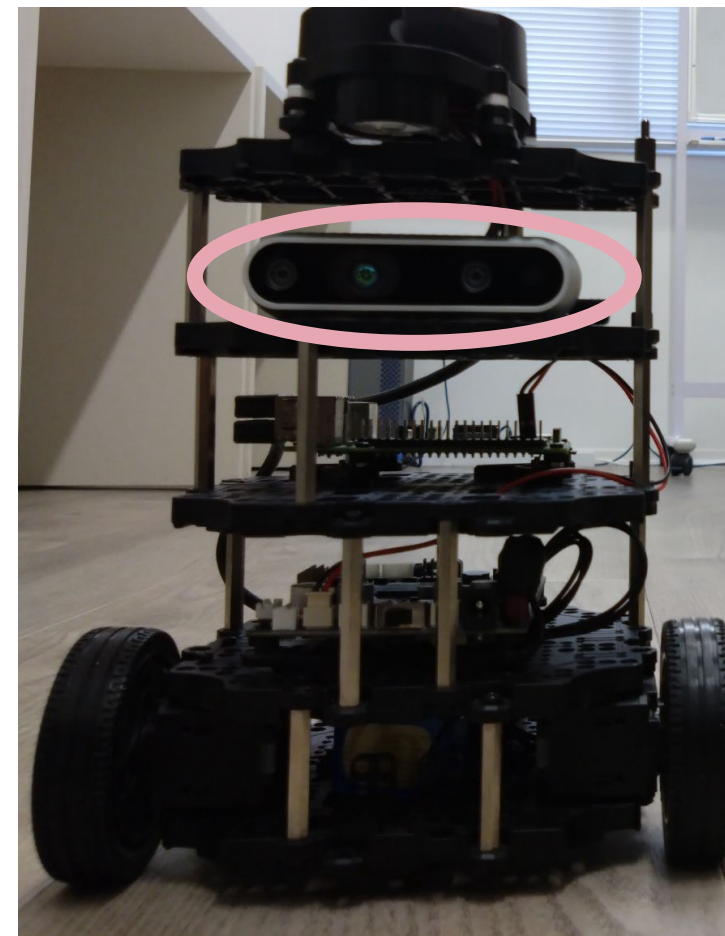
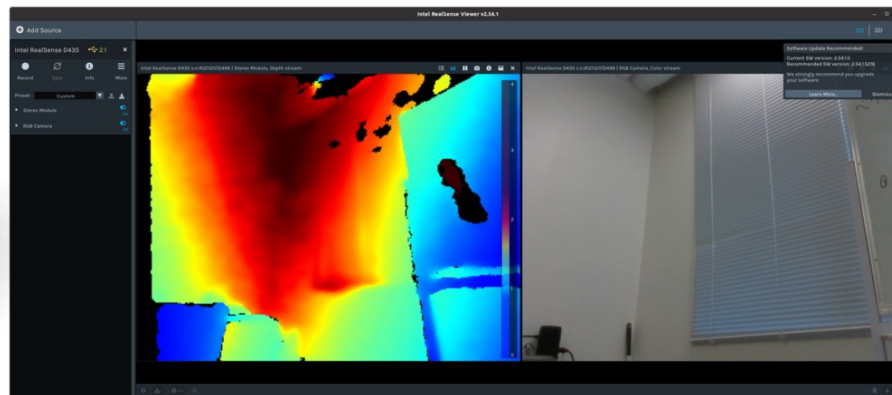
全体構成とデータの流れ

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化

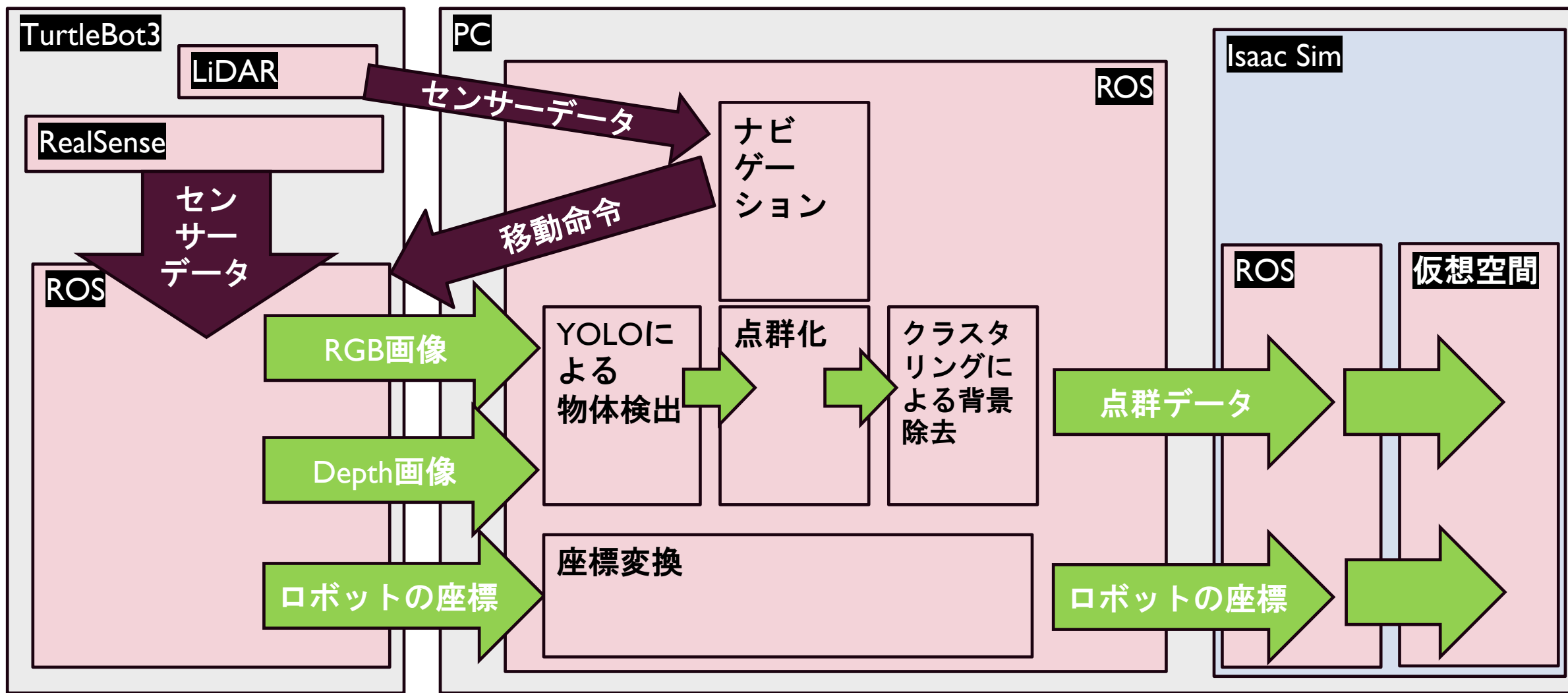


2.4 RealSense D435 (デプスカメラ)

- Intelが販売するデプスカメラ
- 深度画像とRGB画像を取得可能なデバイス
- Intelから、ROSパッケージが提供



自律走行ロボットが取得したデータからの 点群データとしてのリアルタイムな可視化



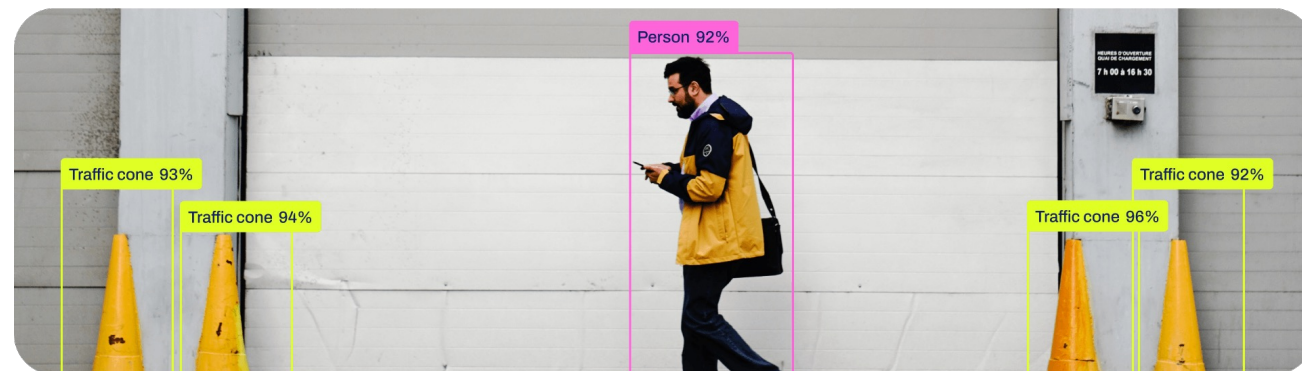
2.5 点群データ処理

- 2.5 受信したデータから，物体を抽出し点群データを作成
 - 2.5.1 YOLOを用いて，RGB画像から物体検出
 - 2.5.2 物体領域に対応する深度画像を点群データに変換
 - 2.5.3 クラスタリングを用いて，物体以外の点群を削除

2.5 点群データ処理

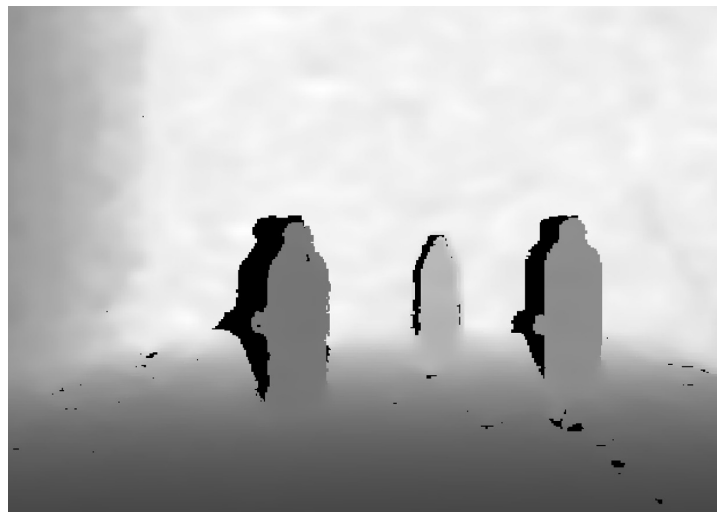
YOLO

- YOLOはYou Only Look Onceをコンセプトに研究されている物体検出技術



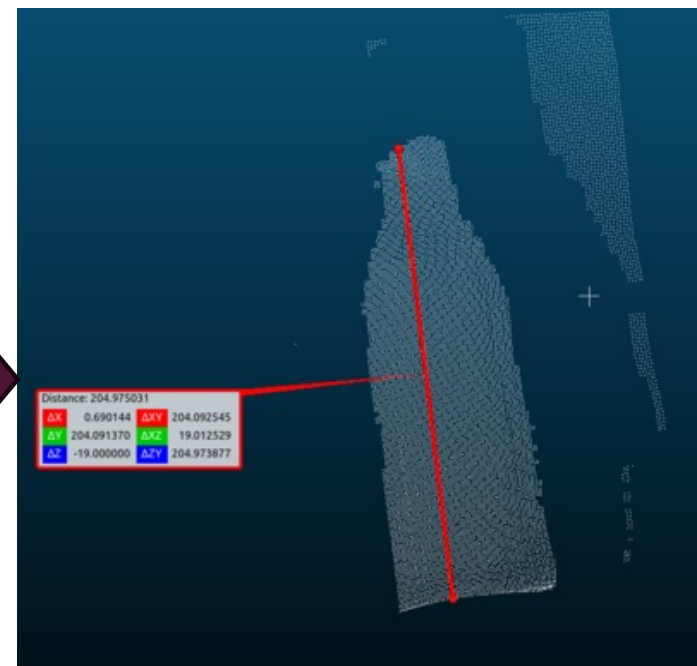
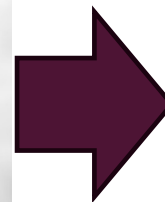
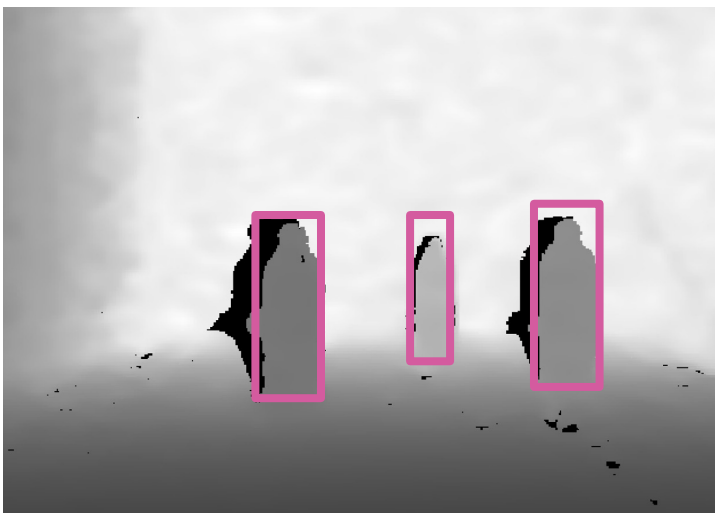
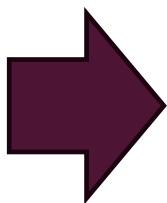
2.5.1 YOLOの実行

- I.YOLOを用いて，RGB画像から物体検出



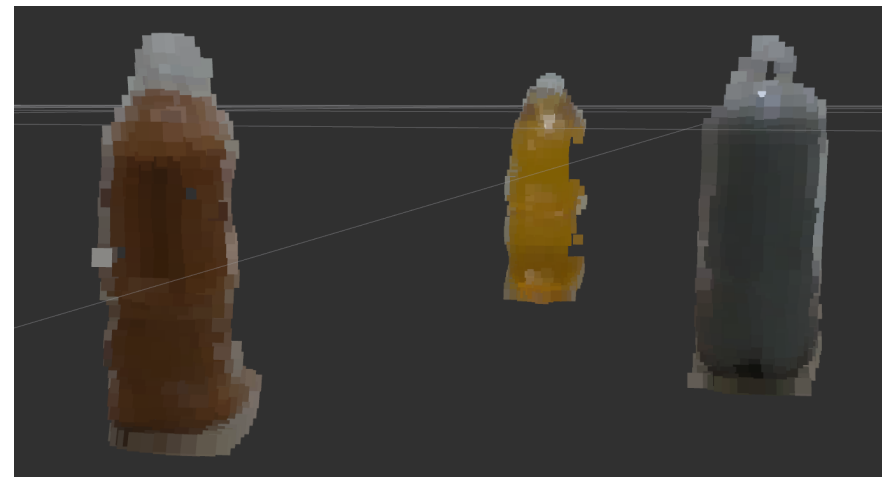
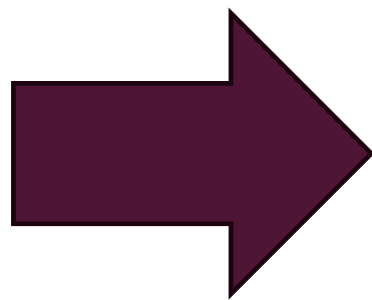
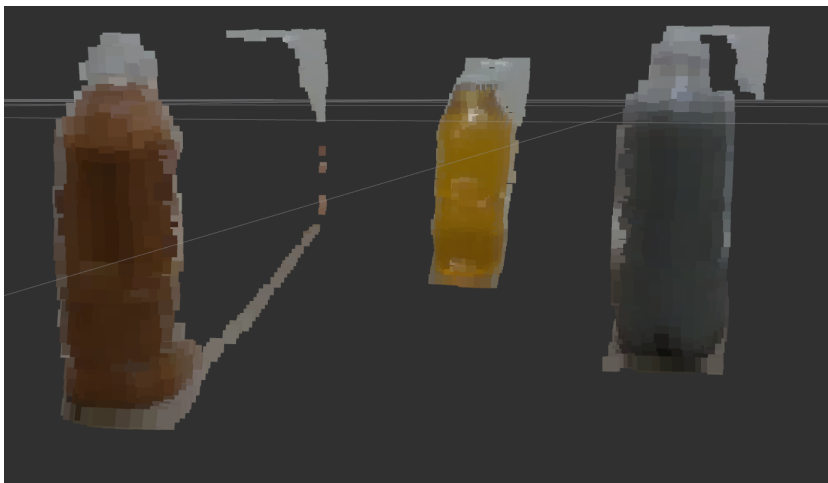
2.5.1 点群化

- 2. 物体領域に対応する深度画像を抽出
- 3. 抽出した深度画像を点群データに変換



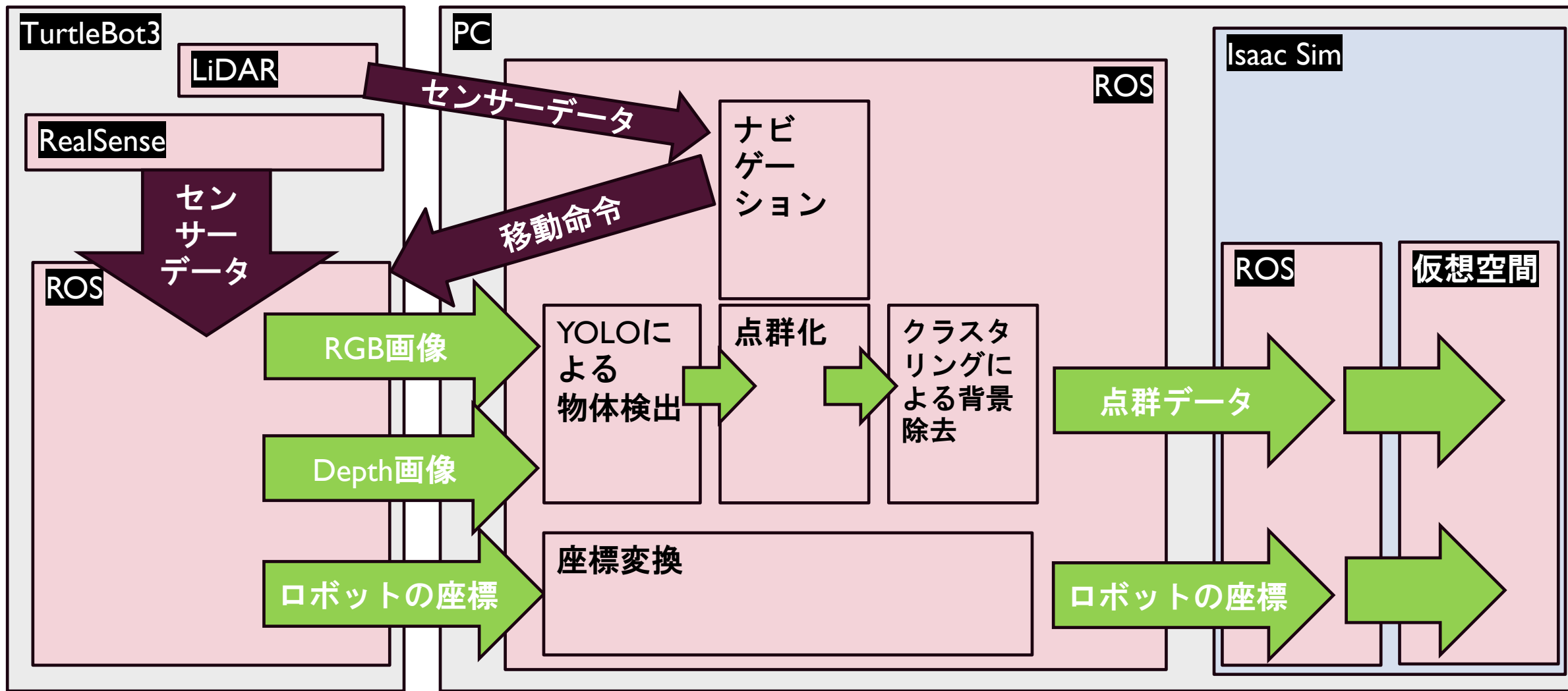
2.5.3 背景除去

- 4. クラスタリングを用いて，物体以外の点群を削除



全体構成とデータの流れ

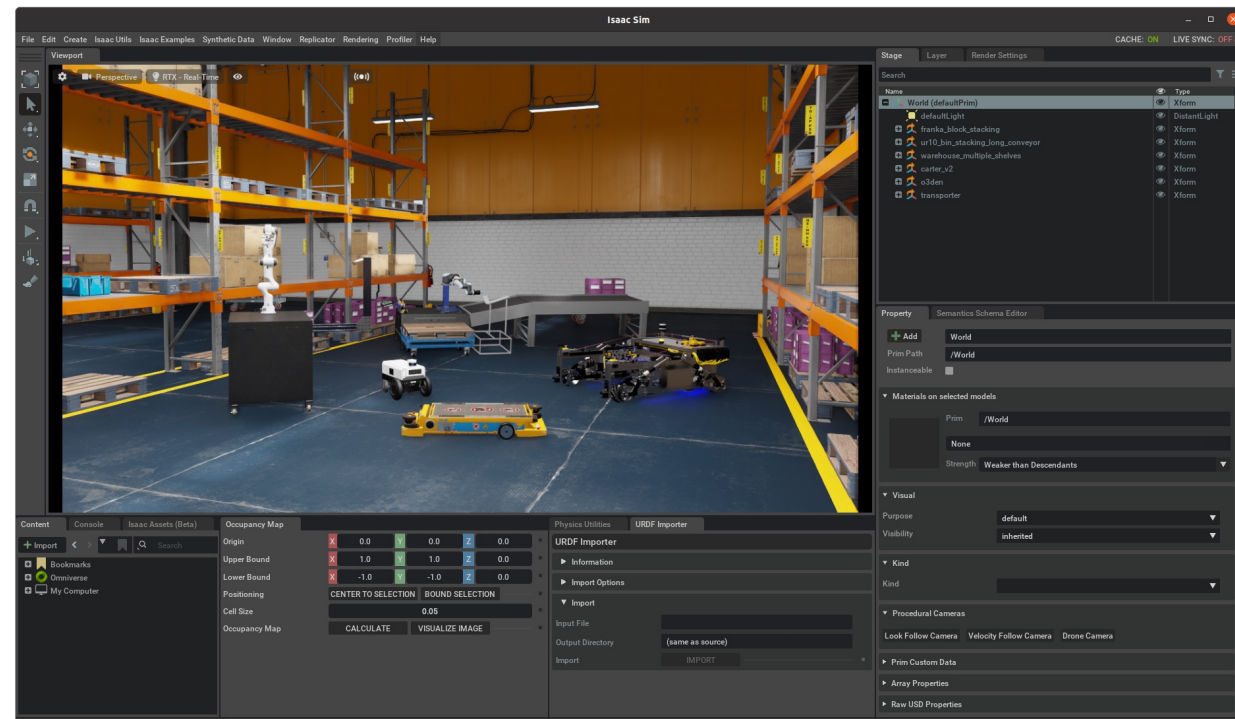
自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化



2.6 現実空間に対応する仮想空間の実現

Isaac Sim

- NVIDIAの提供するロボティクスシミュレーター
- 本演習では，仮想空間の実現に使用



画像引用:<https://docs.omniverse.nvidia.com/isaacsim/latest/overview.html>

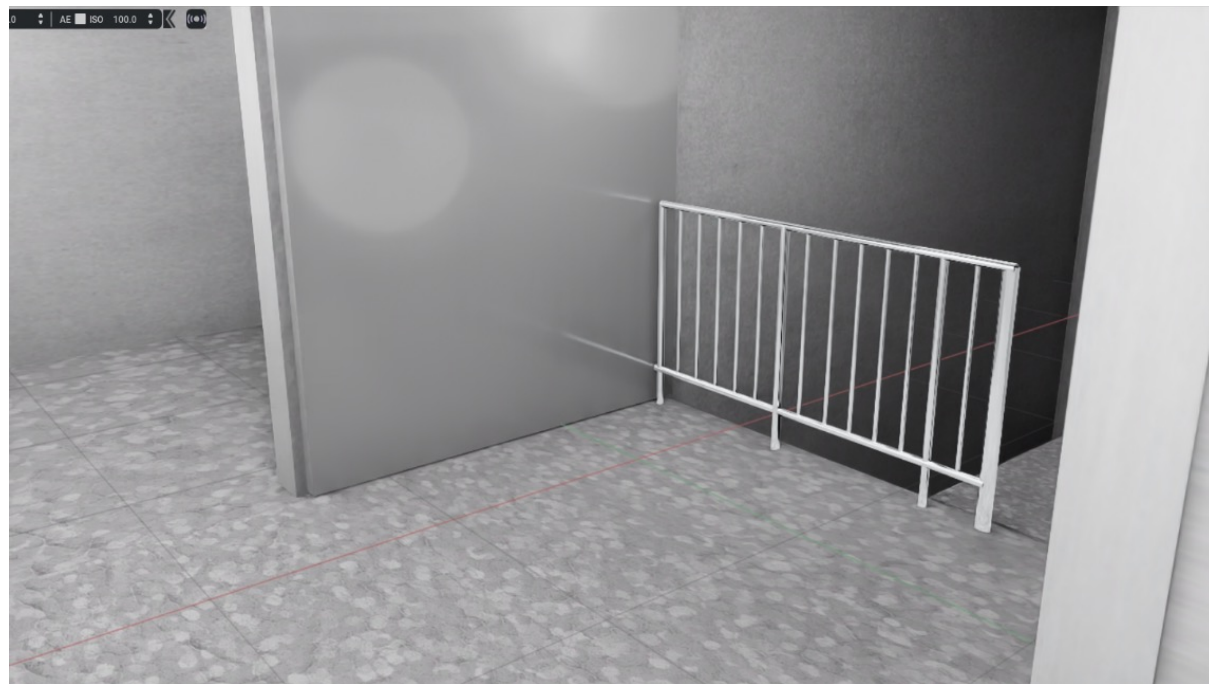
2.6 現実空間に対応する仮想空間の実現

- 現実空間の3Dモデルの用意

現実空間

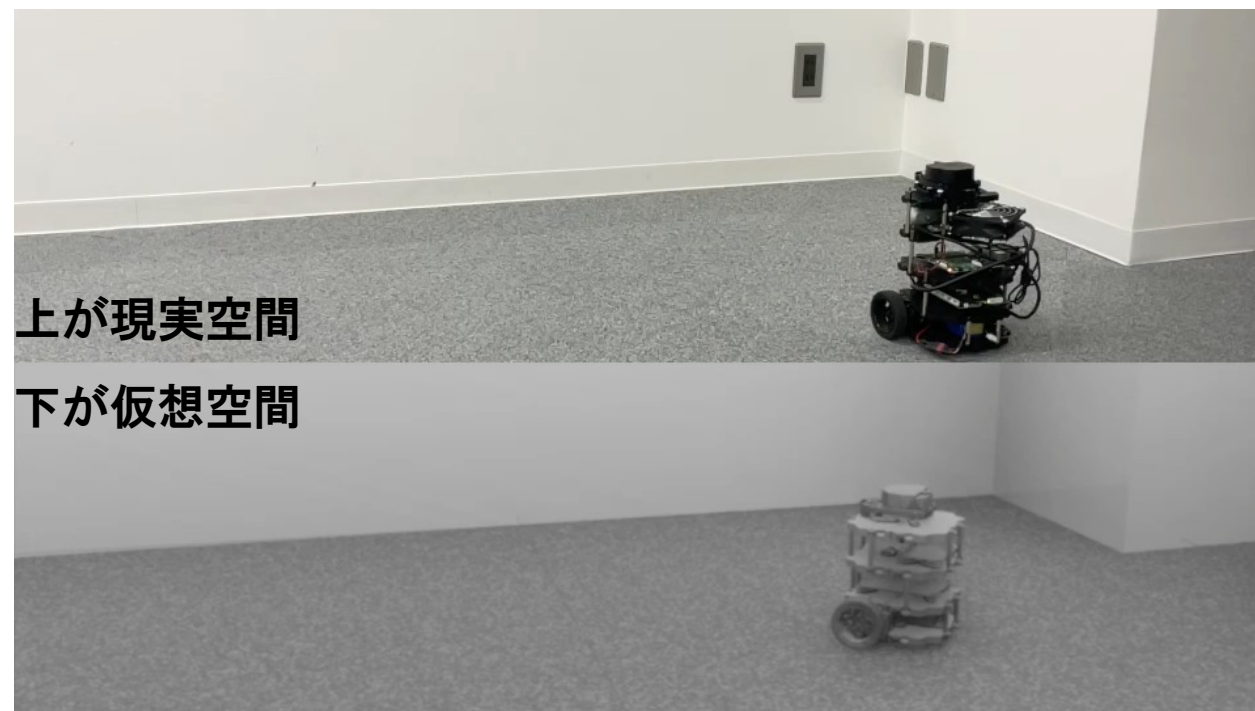


用意した3Dモデル



2.6.1 ロボットの同期

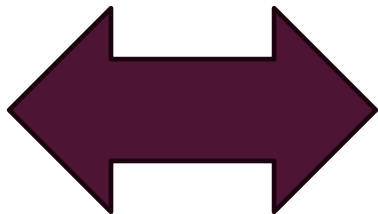
- ロボットの3Dモデル作成
- 現実空間のロボットの座標に対応させて,
仮想空間にロボットを配置



2.6.2 オブジェクトの同期

- 得られた点群データを，仮想空間上に配置

現実空間

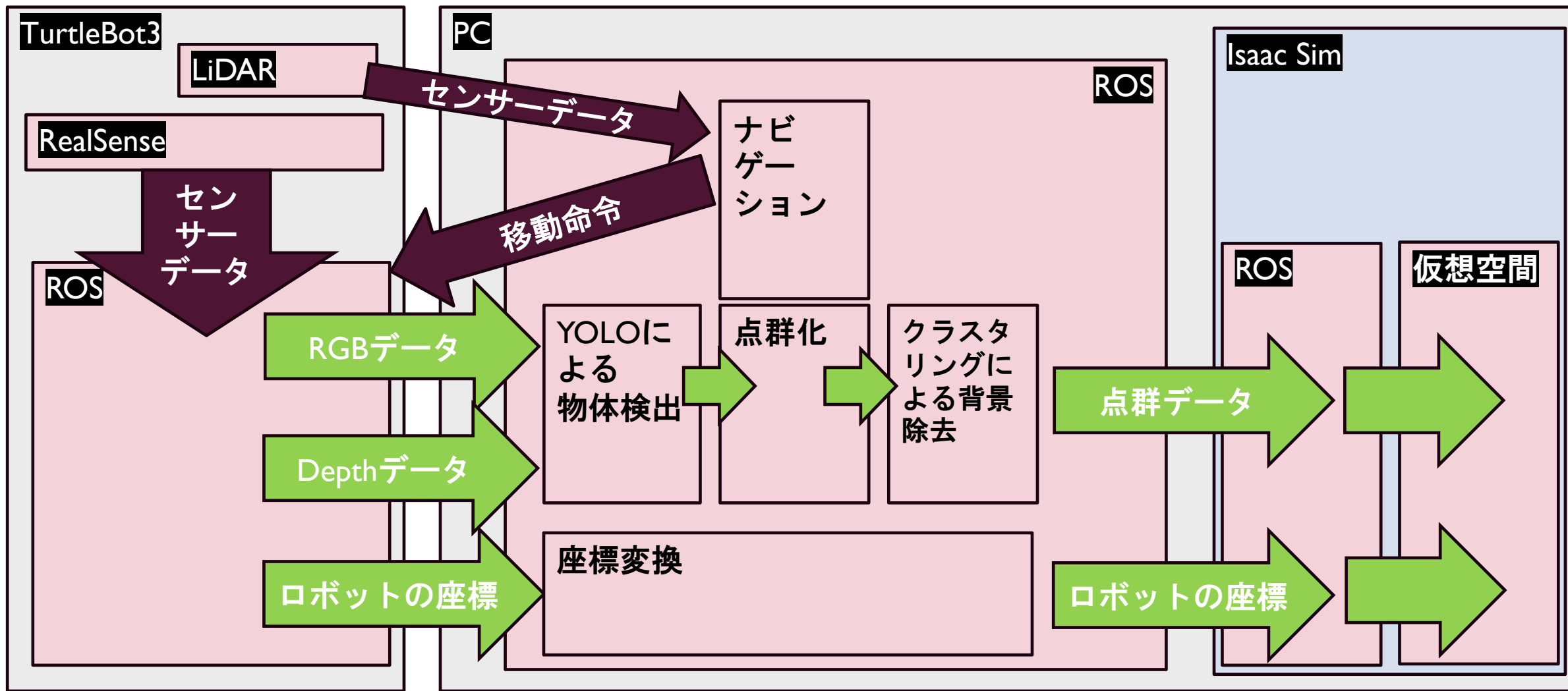


仮想空間



全体構成とデータの流れ

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化

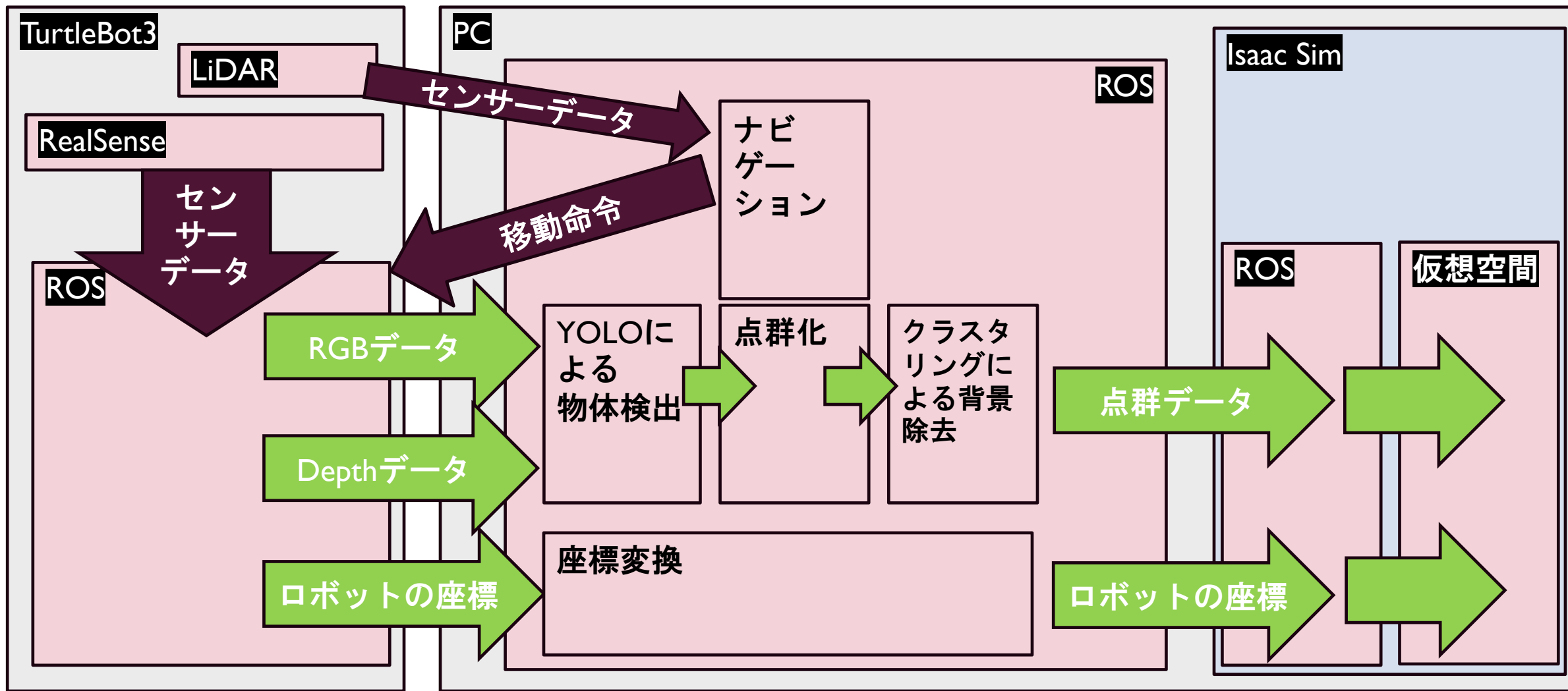


2.7 デモ



全体構成とデータの流れ

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化



目次

- 1 はじめに

- 1.1 背景
- 1.2 動機・目的
- 1.3 先行事例
- 1.4 演習概要

- 2 演習内容

- 2.1 演習内容
- 2.2 演習環境
- 2.3 ロボットの自律走行
- 2.4 RealSense
- 2.5 点群データ処理
- 2.6 可視化

- 3 おわりに

4 おわりに

- 今回の演習は，株式会社Tengun-labelとの共同研究
 - システムアイデアはTengun-label
 - 詳細設計は川口とTengun-labelとの共同
 - 実装は川口が単独担当
(1518行)



Tengun-label

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

4 おわりに

今後の課題

- ロボットの位置推定が正確ではなく，現実の動きと異なる挙動
 - ロボットの位置推定に使うアルゴリズムの変更
- 現在は，ロボットの視界内に「今」ある物しか可視化できていない
 - 一度認識した物体の点群を，可視化し続ける機能の実装
- ロボットのコンピューターの性能不足による画像の破損
 - 搭載するコンピューターをより高性能の物に変更

全体構成とデータの流れ

自律走行ロボットが取得したデータからの
点群データとしてのリアルタイムな可視化

