

TIER IV ACADEMY

自動運転システム構築塾

Day4 Autoware演習②

Autoware 演習 6 : 物体検出とトラッキング



この演習について

Autoware 演習 1 : データの記録・再生

Autoware 演習 2 : センサーキャリブレーション

Autoware 演習 3 : 3次元地図の作成

Autoware 演習 4 : 自己位置推定

Autoware 演習 5 : パラメータ調整

✓ Autoware 演習 6 : 物体検出とトラッキング

Autoware 演習 7 : 信号検出

Autoware 演習 8 : 経路生成と経路計画

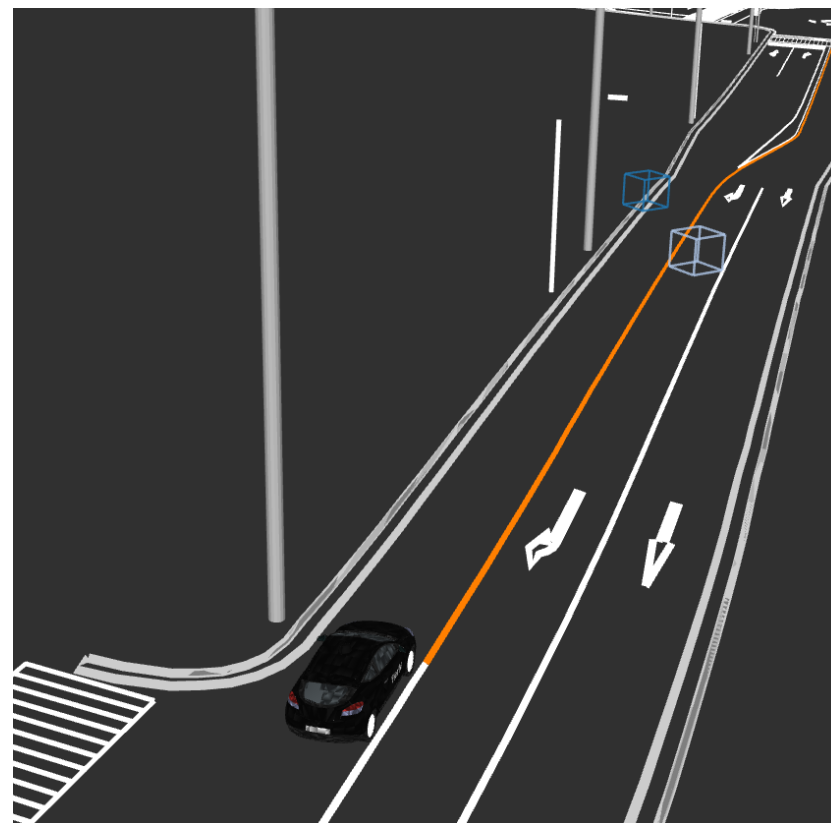
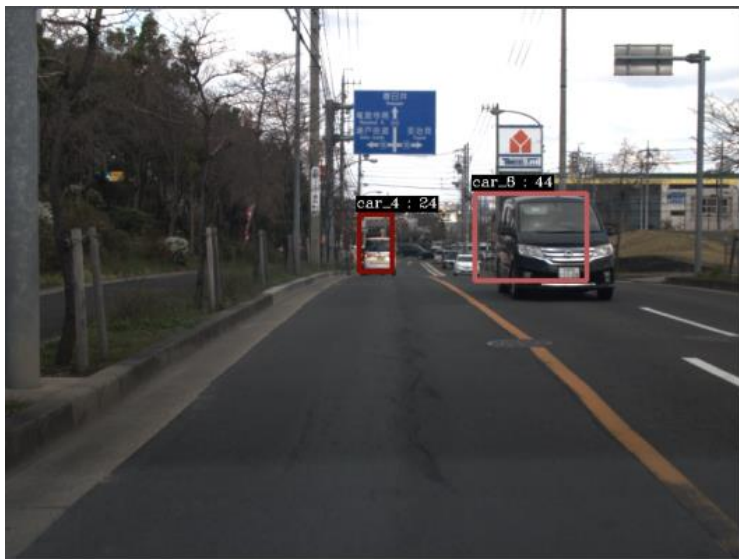
Autoware 演習 9 : 経路追従と車両制御

Autoware 演習 10 : パラメータ調整

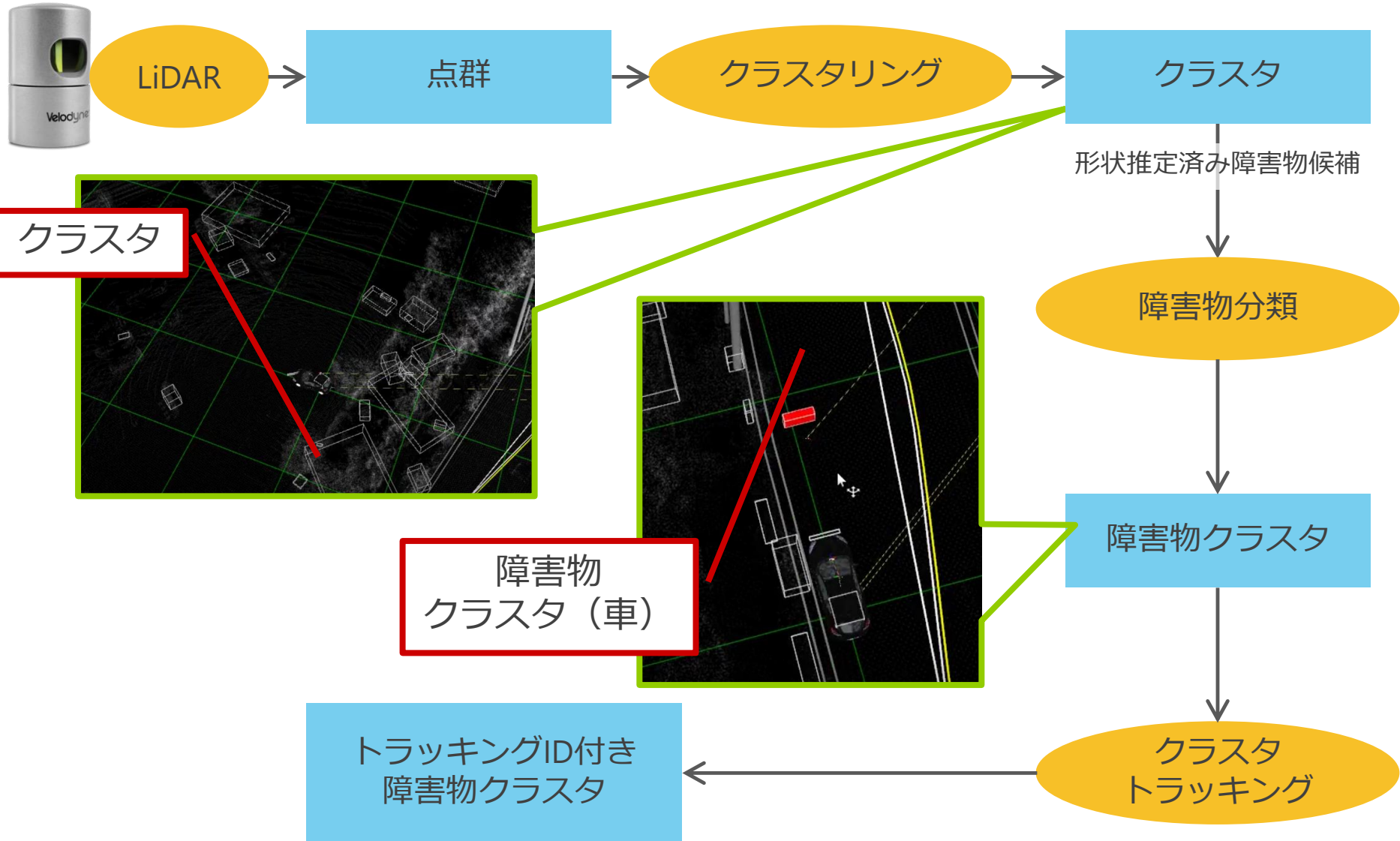
物体検出 - 概要

車両や歩行者を検出

- 3次元座標位置
- 形状
- トラッキングID
- 障害物の種類（車両・歩行者）



物体検出 - 構成・概略図



クラスタリング – 概要・構成



- 点群データから地面の排除
- クラスタの生成
 - ✓ 形状推定
 - ✓ 姿勢推定
 - ✓ 重心推定

CloudClusterArray.msg

```
std_msgs/Header header  
CloudCluster[] clusters
```

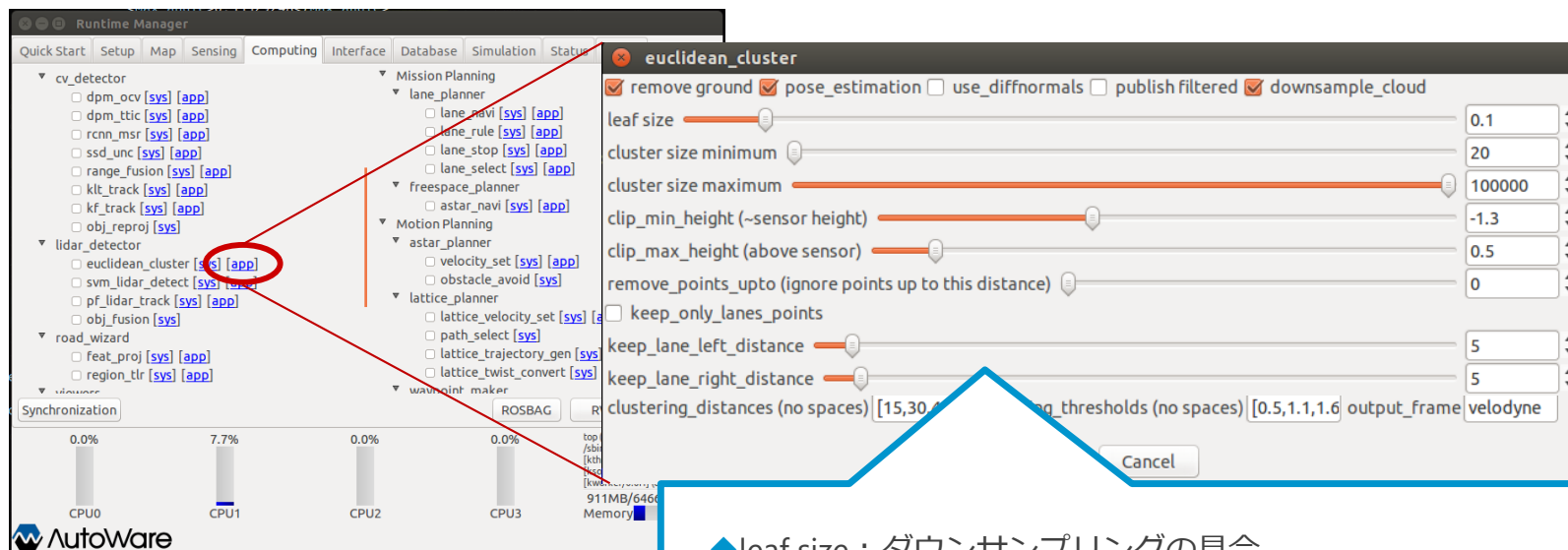
CloudCluster.msg

```
std_msgs/Header header  
uint32 id //トラッキングID  
string label //オブジェクト名  
sensor_msgs/PointCloud2 cloud //クラスタ  
geometry_msgs/PointStamped min_point //最小  
geometry_msgs/PointStamped max_point //最大  
geometry_msgs/PointStamped avg_point //平均  
geometry_msgs/PointStamped centroid_point //重心  
float64 estimated_angle //クラスタの方向  
geometry_msgs/Vector3 dimensions //立方体の大きさ  
geometry_msgs/Vector3 eigen_values //固有値  
geometry_msgs/Vector3[] eigen_vectors //固有ベクトル  
jsk_recognition_msgs/BoundingBox bounding_box
```

Autaware/ros/src/computing/perception/detection/packages/lidar_tracker/nodes/euclidean_cluster

クラスタリング – 手順 (1/2)

1. 「Computing」 タブの 「euclidean_cluster」 のappでパラメータの確認



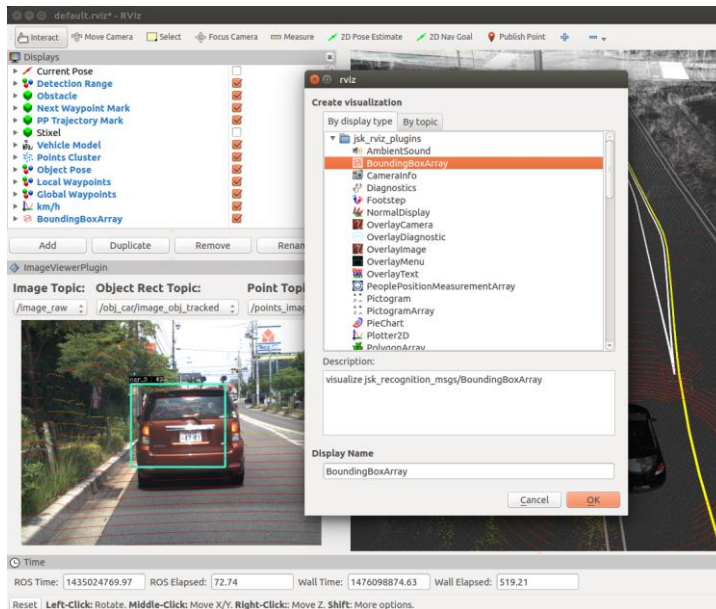
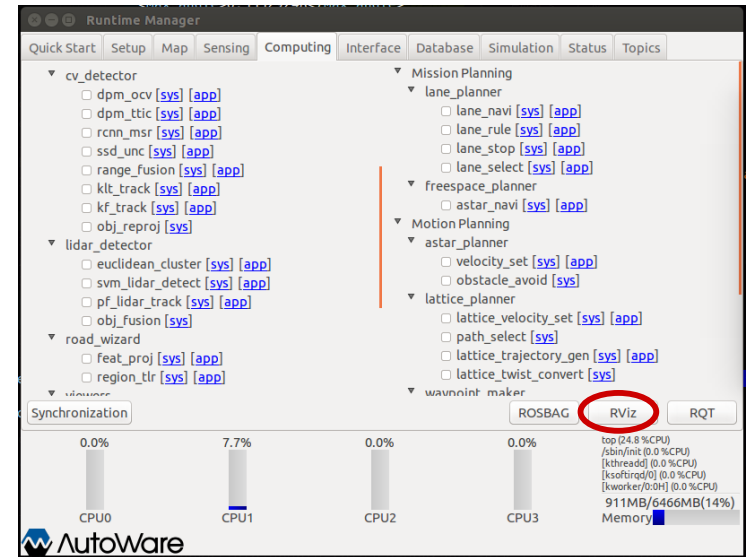
- ◆ leaf_size : ダウンサンプリングの具合
- ◆ cluster_size_minimum : clusterの点数の下限
- ◆ cluster_size_maximum : clusterの点数の上限
- ◆ clip_min_height : クラスタリング範囲 (センサからの最小の高さ)
- ◆ clip_max_height : クラスタリング範囲 (センサからの最大の高さ)
- ◆ remove_points_upto : 点群データを無視する範囲 (センサからの距離)
- ◆ keep_only_lane_points : 左右のクラスタリングする範囲
- ◆ clustering_distances : しきい値変更距離 (5段階)
- ◆ clustering_thresholds : 距離に応じた5段階のしきい値設定

2. 「euclidean_cluster」 を して起動

クラスタリング – 手順 (2/2)

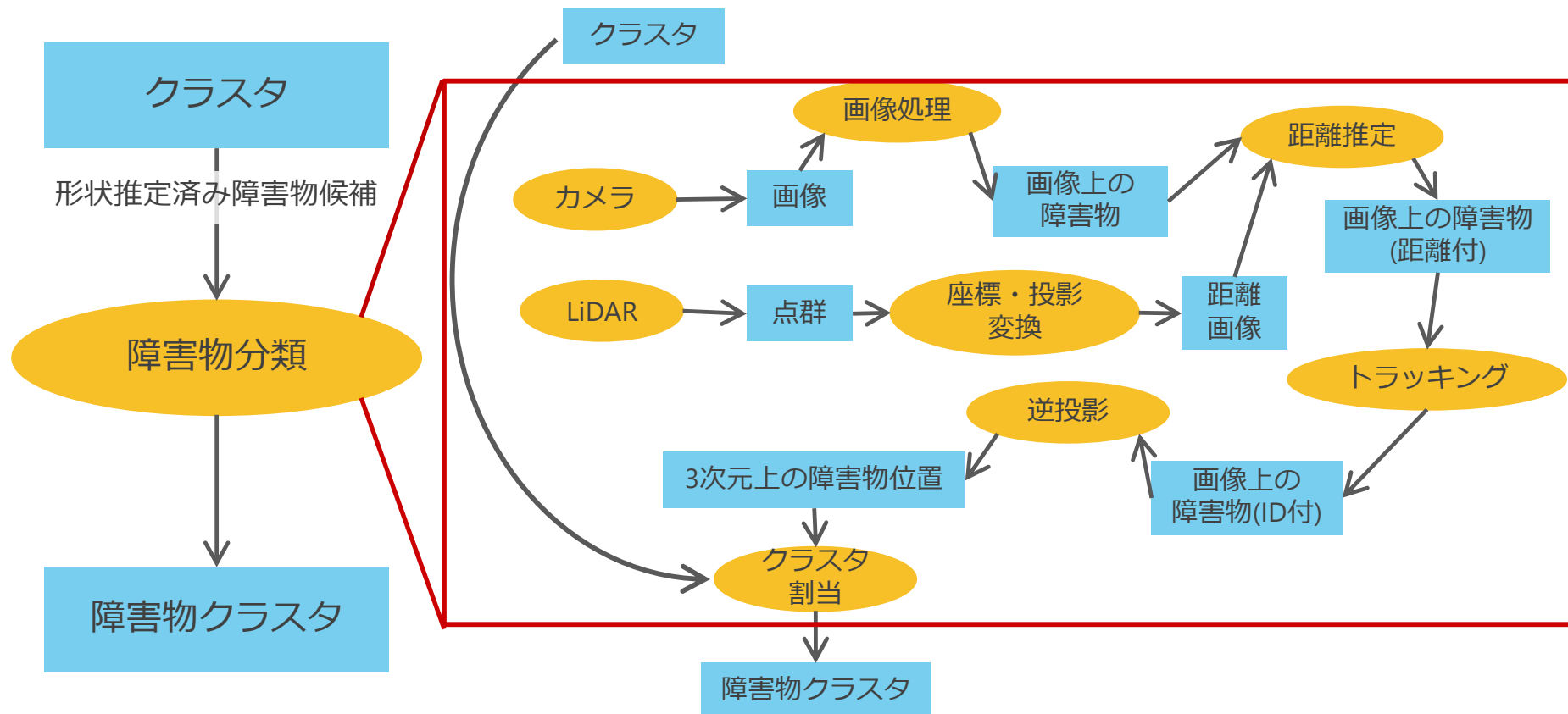
3. RVizでクラスタを確認

- A) 「RViz」をクリック
- B) 「Add」ボタンをクリック
- C) 「BoundingBoxArray」を選択
- D) Rviz左のトピックリストに「BoundingBoxArray」が追加された後、「Topic」で「/bounding_box」を選択



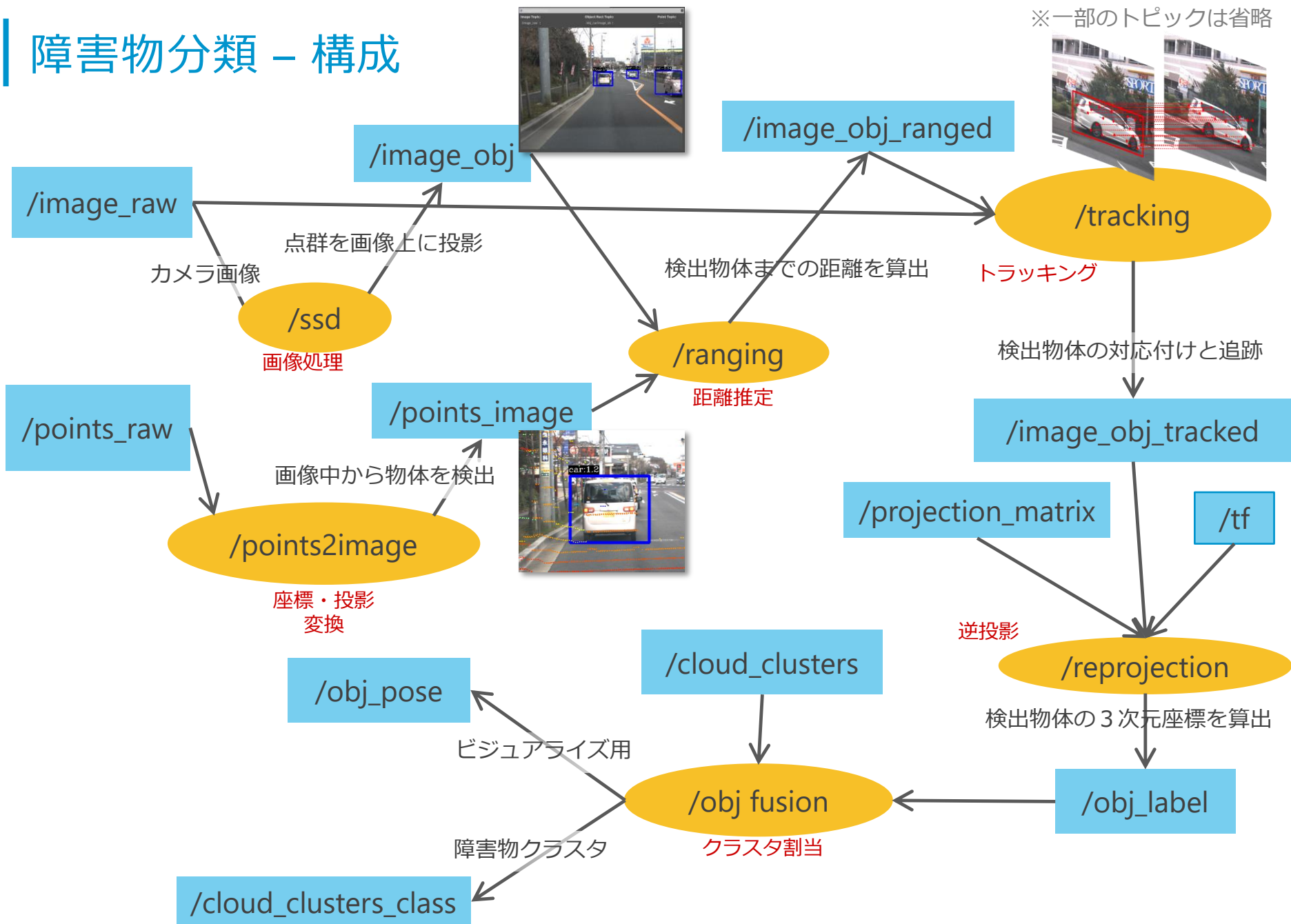
障害物分類 – 概要

- 手法1 : カメラとLiDARによるセンサフュージョン
- 手法2 (今後) : 点群クラスタのSVM (機械学習) による判断
- カメラとLiDARによるセンサフュージョン



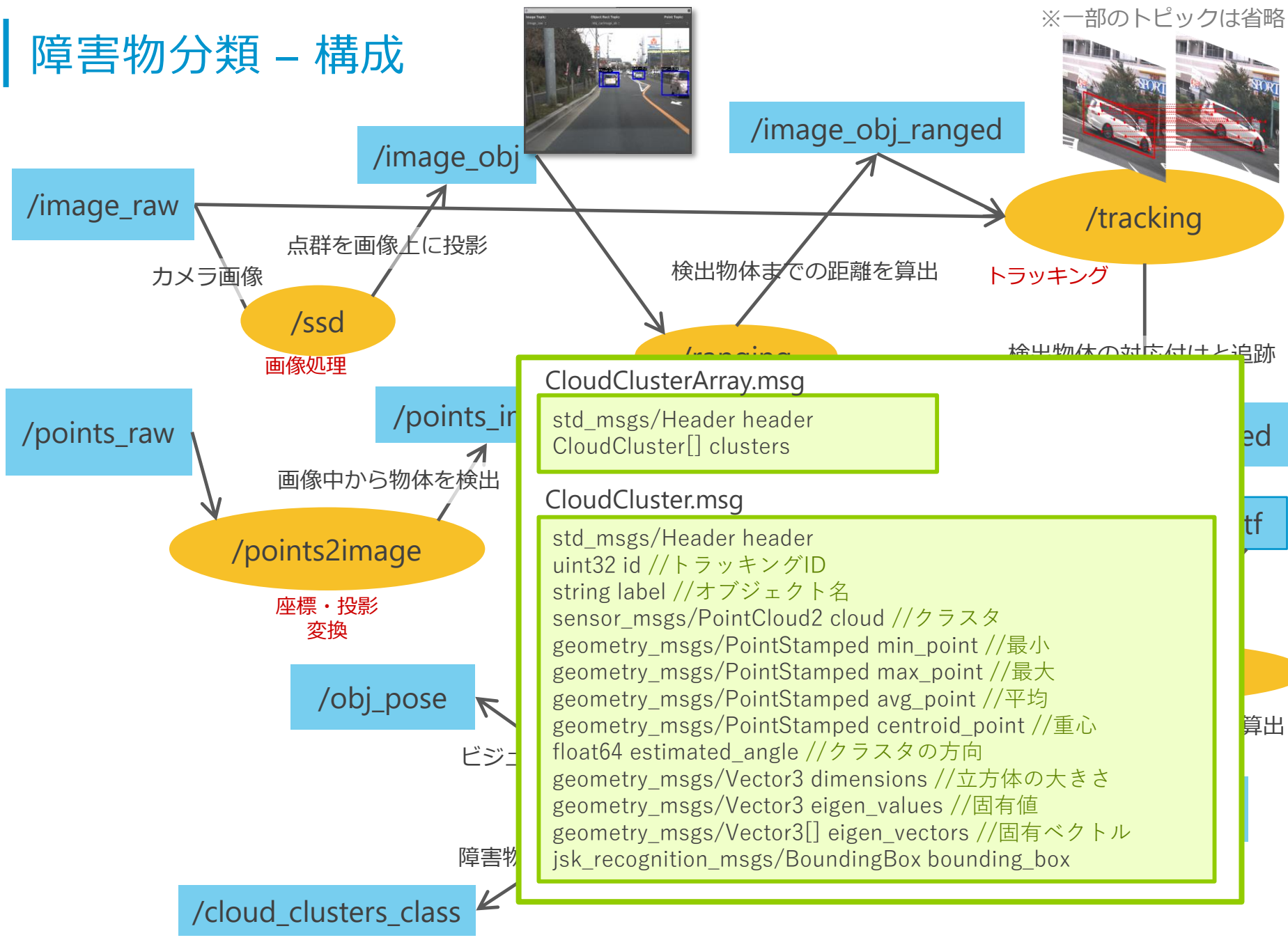
障害物分類 - 構成

※一部のトピックは省略



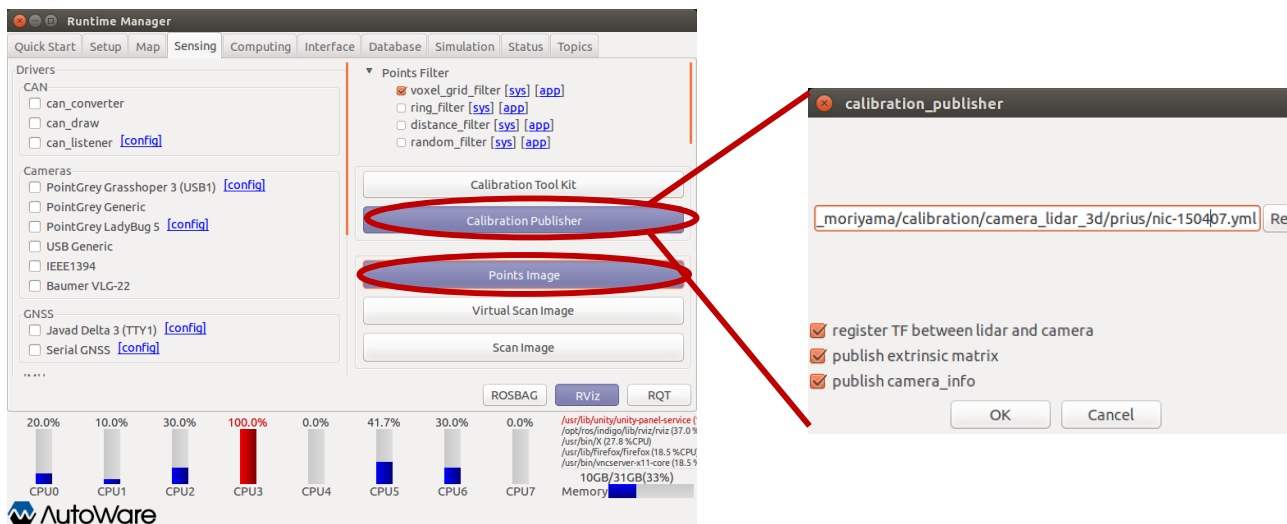
障害物分類 - 構成

※一部のトピックは省略



障害物分類 – 手順 (1/5)

- 座標・投影変換：points2imageノードの起動



1. Calibrationファイルの読み込み

- A) 「Sensing」タブの「Calibration Publisher」ボタンをクリック
- B) ファイルの選択ウィンドウが表示にて、Calibration ファイルを選択

2. Points Image を起動

- A) 「Points Image」ボタンをクリック

障害物分類 – 手順 (2/5)

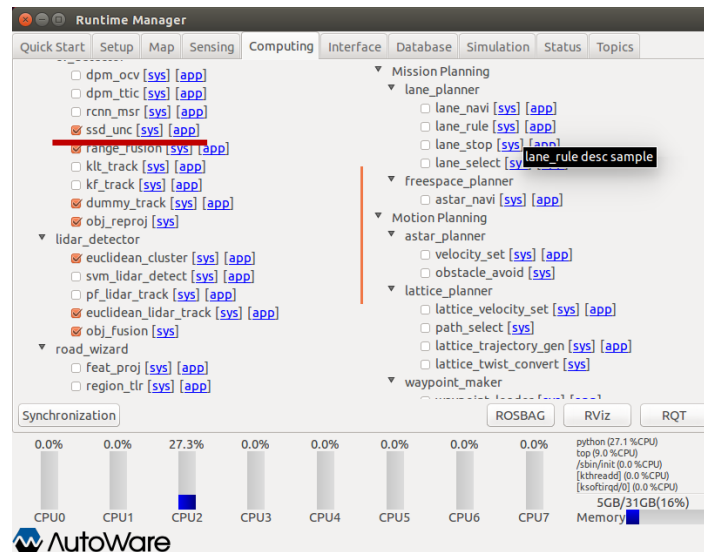
- 画像処理：ssdノードの起動
- 距離推定：rangingノードの起動
- トラッキング：trackingノードの起動

3. ssd_uncを起動

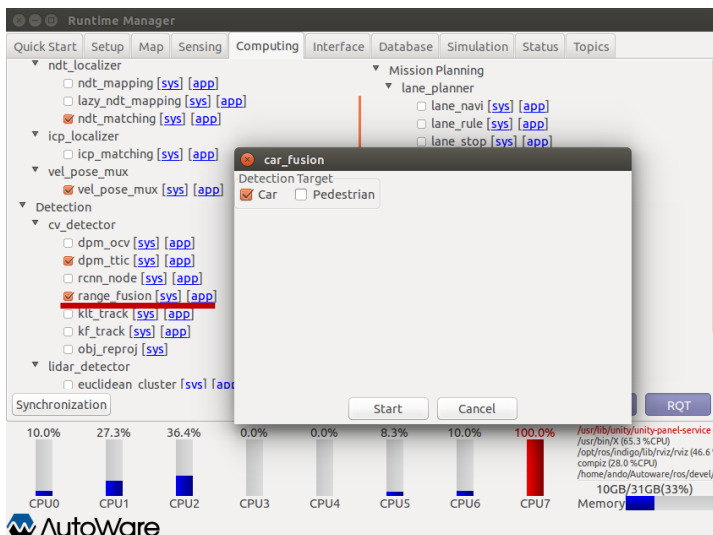
4. range_fusionを起動

A) 「range_fusion」をチェックし、表示されるウィンドウにて「Start」ボタンをクリック

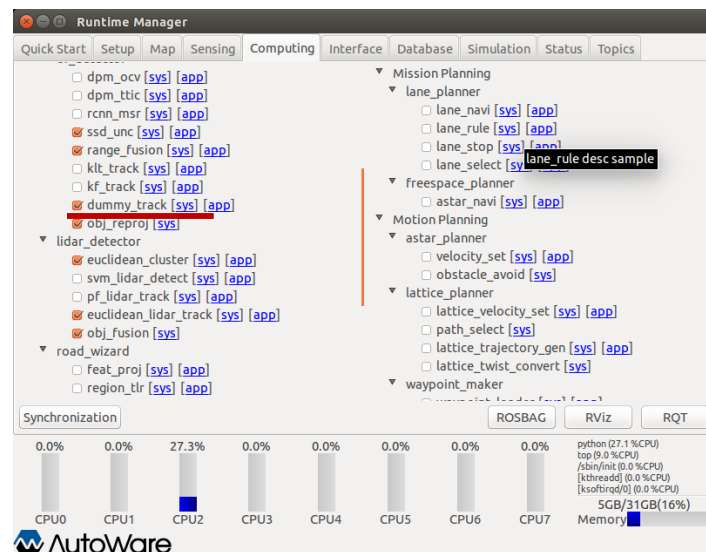
5. dummy(klt, kf)_trackを起動



3. ssd_uncの起動



4. range_fusionの起動

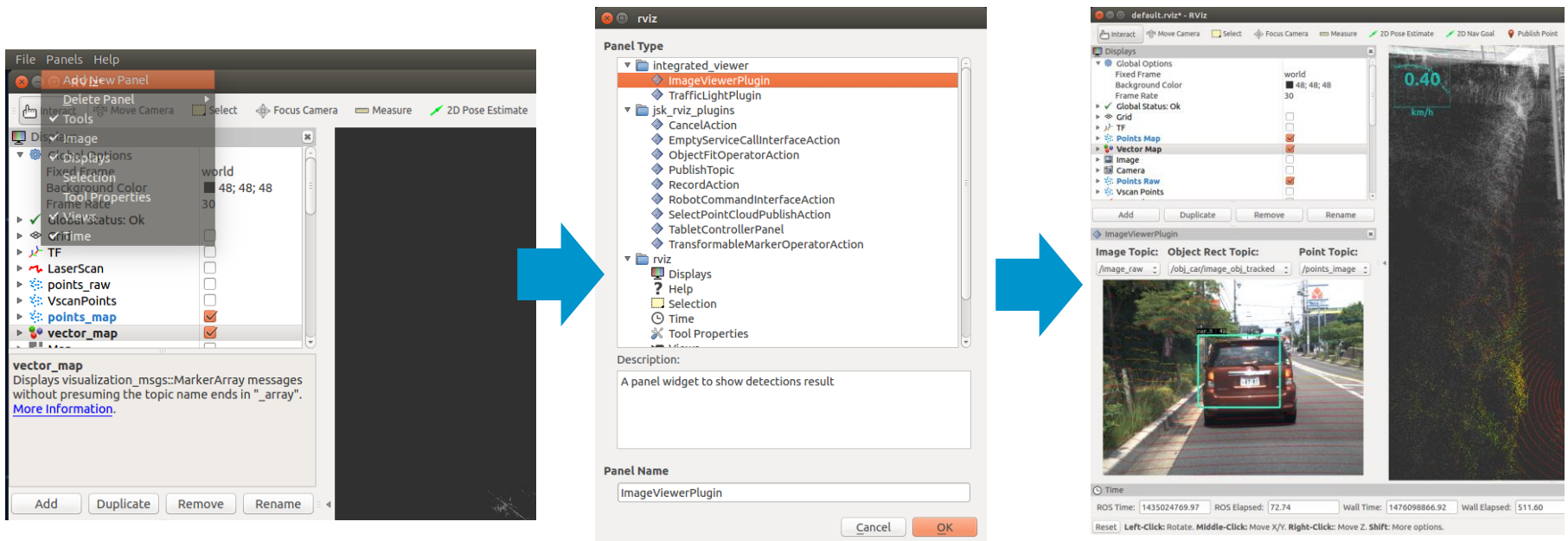


5. dummy_trackの起動

障害物分類 – 手順 (3/5)

6. 検出結果表示用のパネルを追加

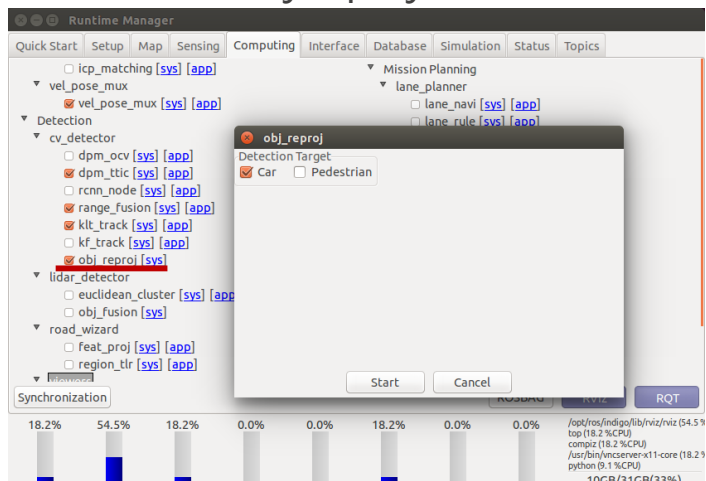
- A) RVizに移り、「Panels」→「Add new panel」を選択
- B) 「Image Viewer Plugin」を選択
- C) 「ImageViewerPlugin」の「Image Topic:」を下図のように選択



6. 検出結果表示用パネルの追加

障害物分類 – 手順 (4/5)

● 逆投影： obj_reprojノードの起動



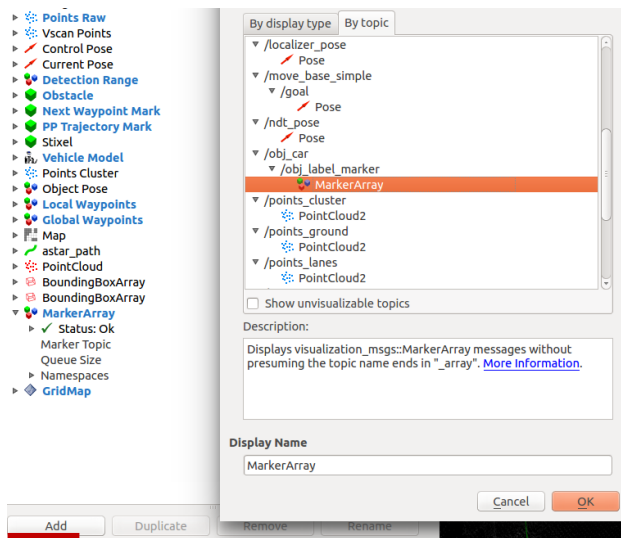
7. obj_reprojの起動

7. obj_reprojを起動

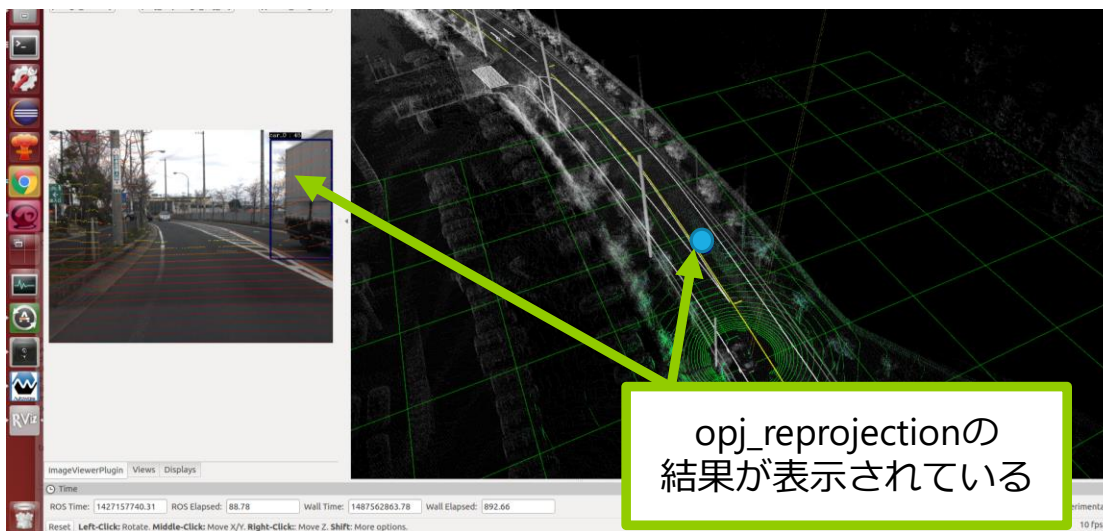
- A) 「obj_reproj」をチェックし、表示されるウィンドウにて「Start」ボタンをクリック

8. obj_reprojの結果を表示

- A) 「Add」ボタンを押す
- B) By topicにて「obj_label_marker」を選択



8. (A)(B)obj_reprojectionのmarkerを追加



8. obj_reprojectionの結果を表示

障害物分類 – 手順 (5/5)

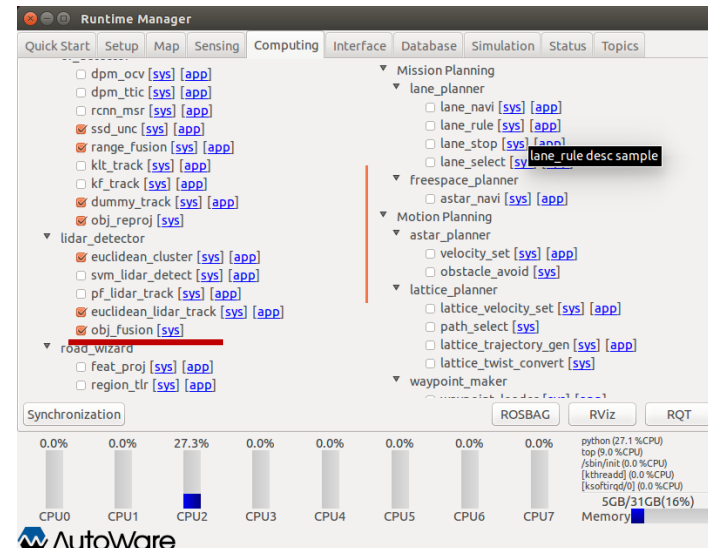
- クラスタ割当： obj fusionノードの起動

9. obj_fusionを起動

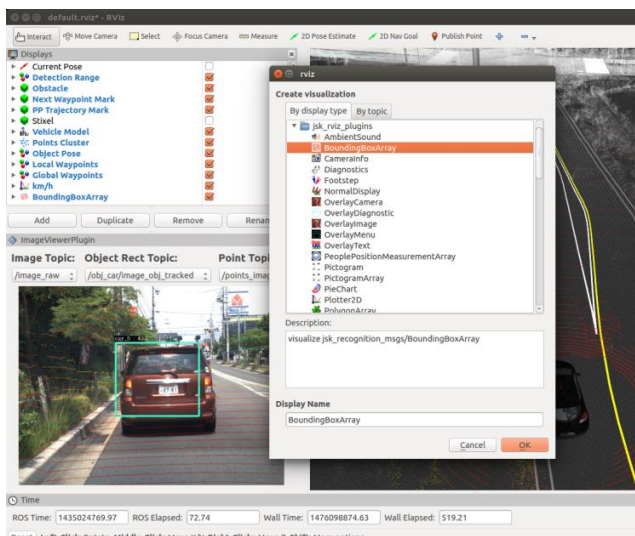
- A) 「obj_reproj」をチェックし、表示されるウィンドウにて「Start」ボタンをクリック

10.obj_fusionの結果を表示

- A) 「Add」ボタンをクリック
- B) 「BoundingBoxArray」を選択
- C) RVizの左のトピックリストに「BoundingBoxArray」が追加されるので、「Topic」で「/obj_car/obj_pose」を選択



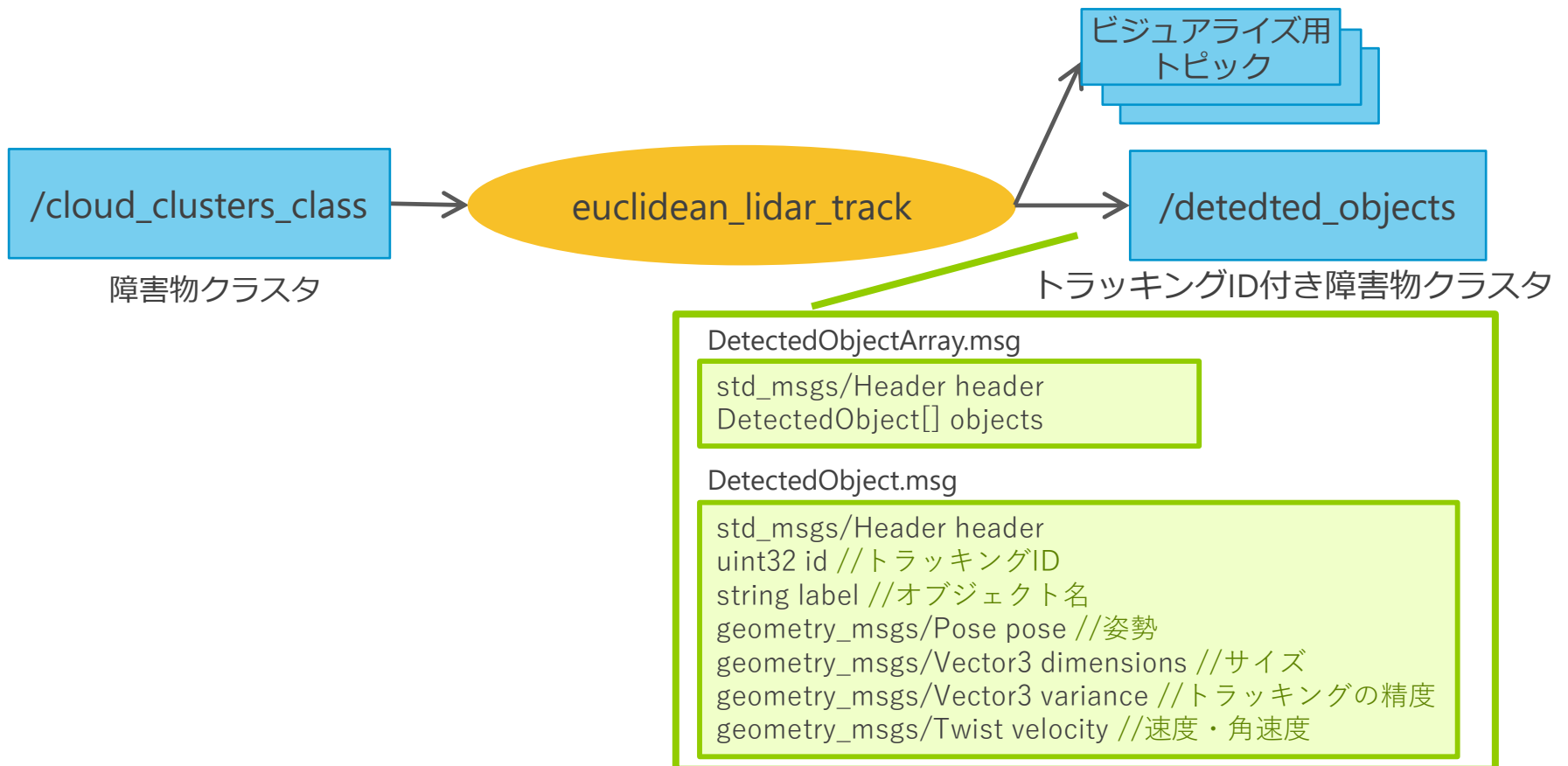
9. obj fusionの起動



10. obj fusionの表示

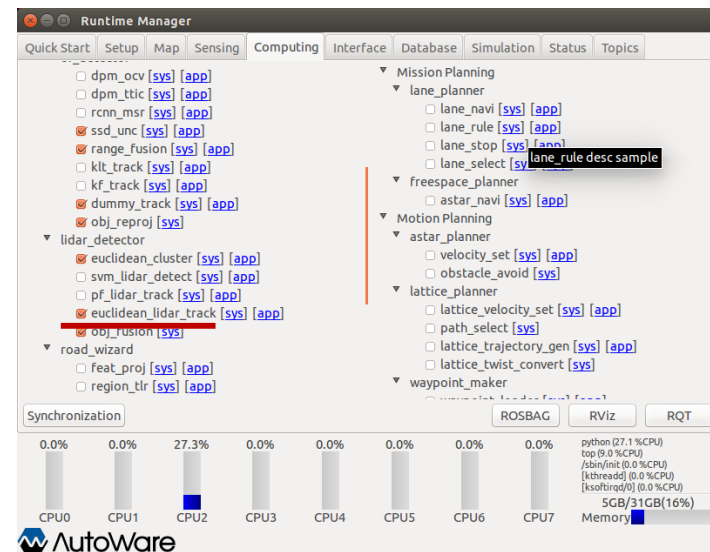
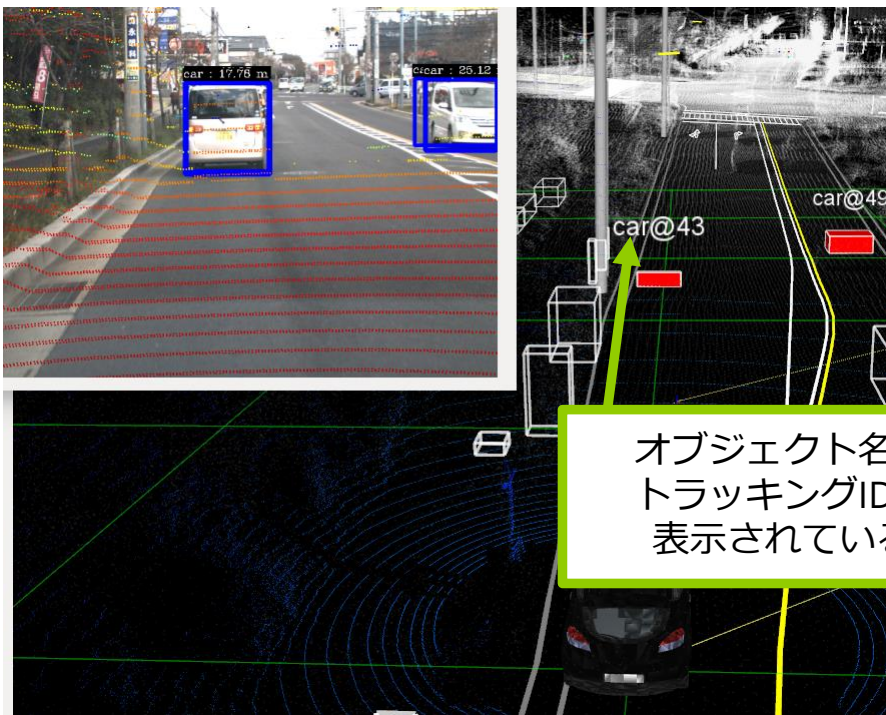
クラスタトラッキング – 概要・構成

- 手法1 : 前回のクラスタとの差分 (距離) によるトラッキング
- 手法2 (今後) : カルマンフィルタによるトラッキング
- 手法3 (今後) : パーティクルフィルタによるトラッキング

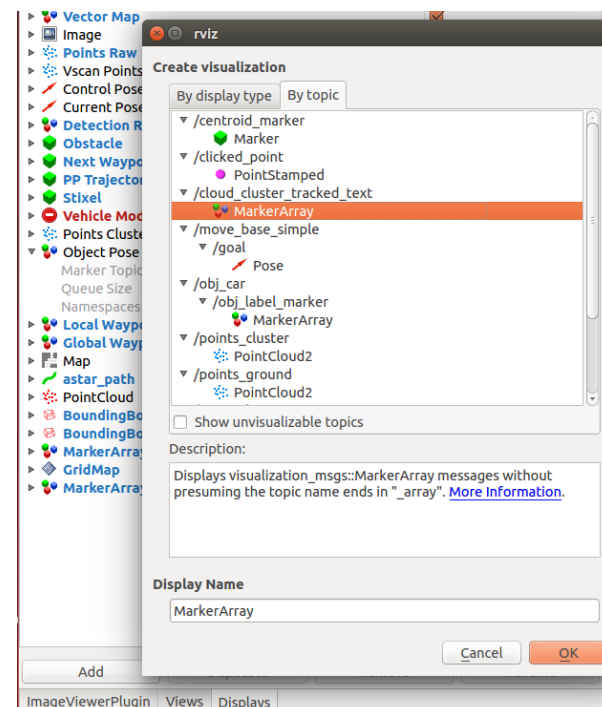


クラスタトラッキング – 手順

1. 「Computing」タブの「euclidean_lidar_track」を
☑して起動
2. euclidean_lidar_trackの結果を表示
 - A) 「Add」ボタンを押す
 - B) By topicにて「cloud_cluster_tracked_text」を選択



1. euclidean_lidar_trackの起動



2. euclidean_lidar_trackの結果表示 17



Intelligent Vehicle

www.tier4.jp