iMorpheus.ai

Weekly Journal Club

NDT localization outline

Friday 09 Mar 2018, 12:00PM

Journal Club

JOURNAL CLUB介绍与自动驾驶中定位方案相关的论文， 主要关注的方向有： SLAM算法、点云数据的处理和压缩、 特征地图、传感器数据处理和融合、 GNSS信号处理等。我们一直关注领域前沿技术， 选取得到广泛认可的、或者是在我们的实际使用中结果比较好的论文， 与大家分享，共同学习成长。

每周五 北京时间12点

http://imorpheus.ai/journalclub

扫码加入无人驾驶技术群

1. Problems in localization
2. Autoware VS Apollo
3. Options

iMorpheus.ai

1. Problems in localization

How to localize a moving vehicle with sensors && a computer.

- Without memory, GNSS || RTK+IMU is used.
  - Noises || Weak signal || Multi-path Effect
- With memory ⇔ Maps, Sensors(GNSS/IMU, Lidar, Camera…) are used.
  - How to build a HD map.
  - How to localize with online data in a HD map.

2.1 Localization in autoware

Autoware && Manuals in github:

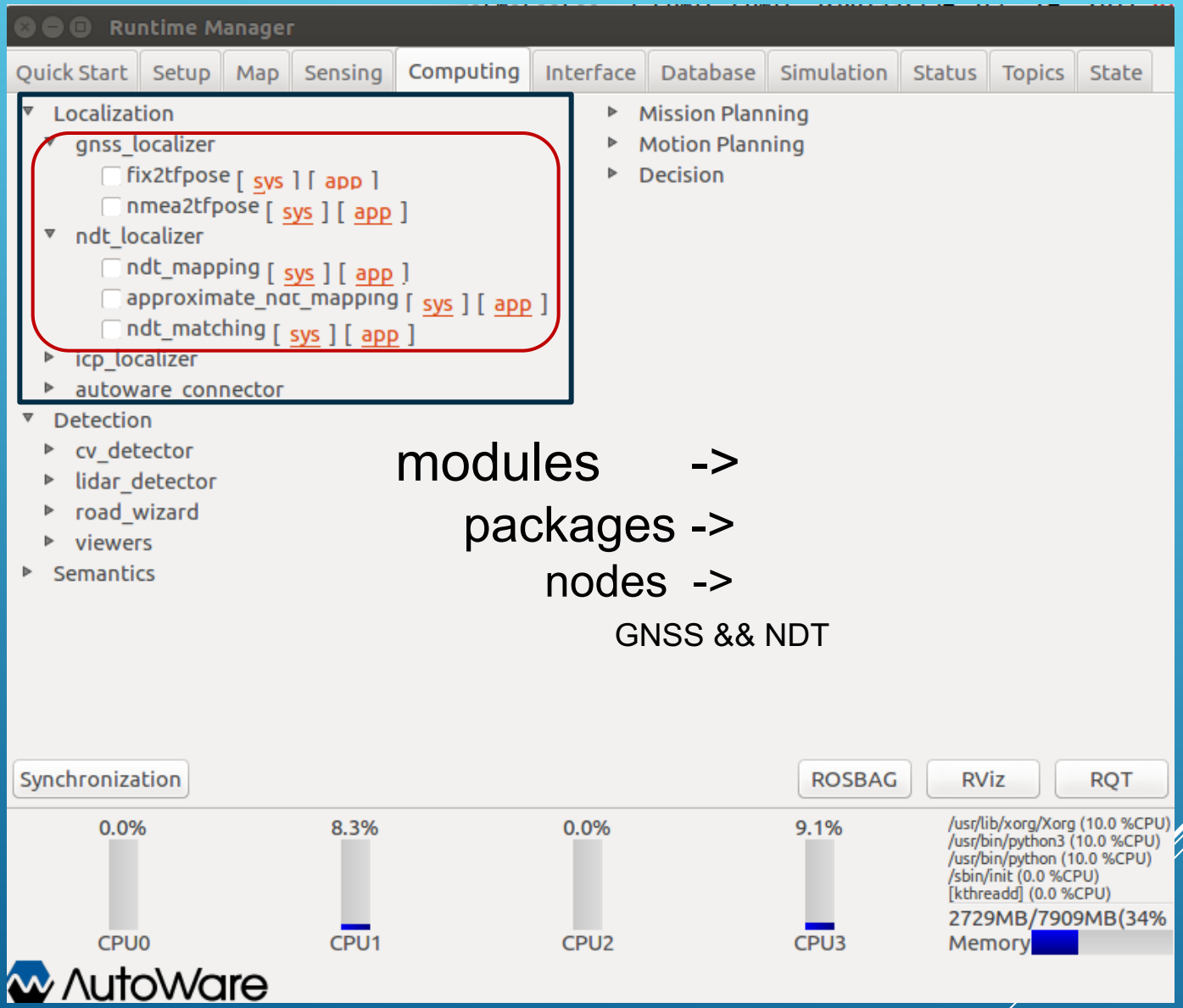
https://github.com/CPFL/Autoware

https://github.com/CPFL/Autoware-Manuals

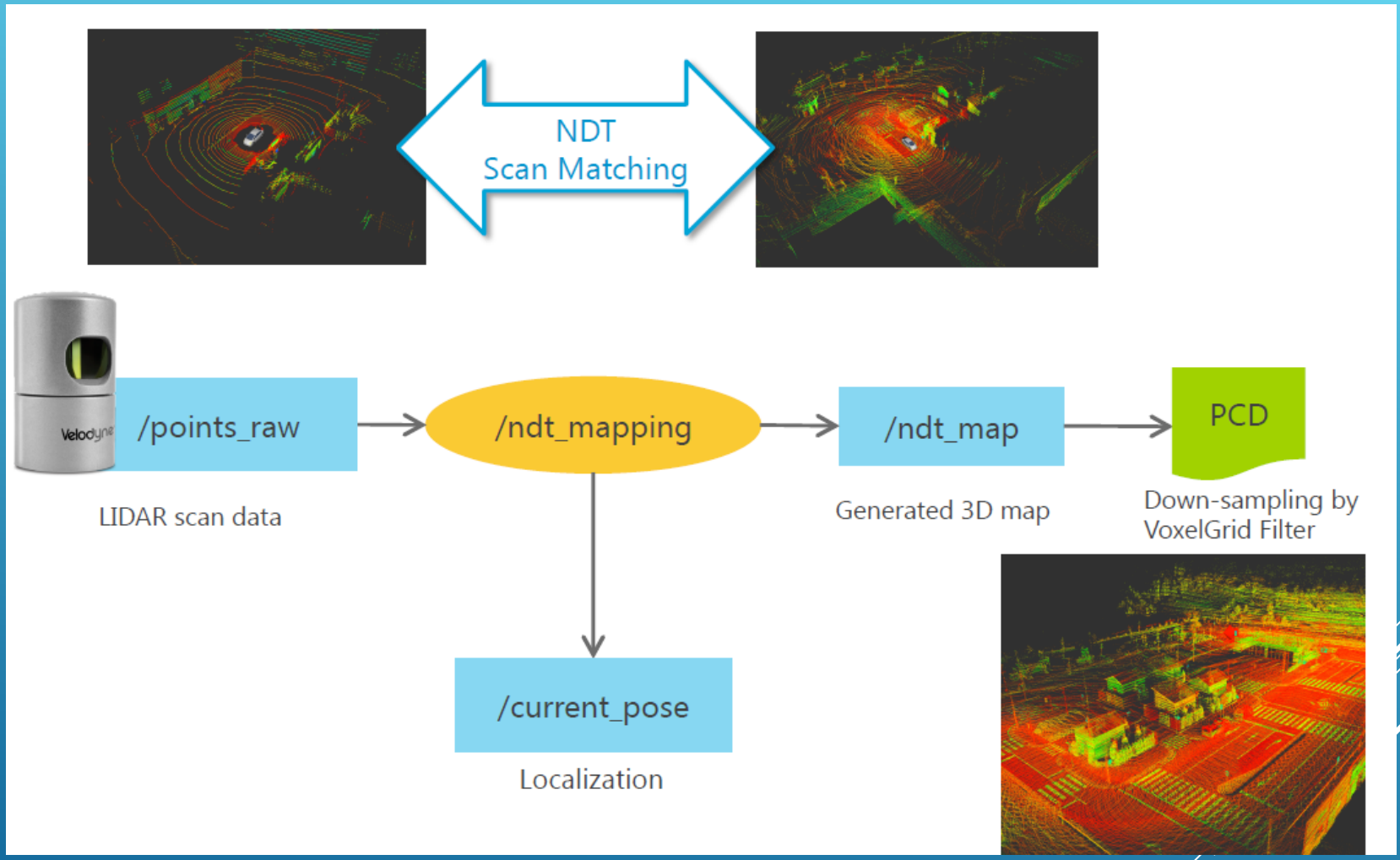
Website of autoware group:

https://tier4.jp/en/

2.1 Localization in autoware

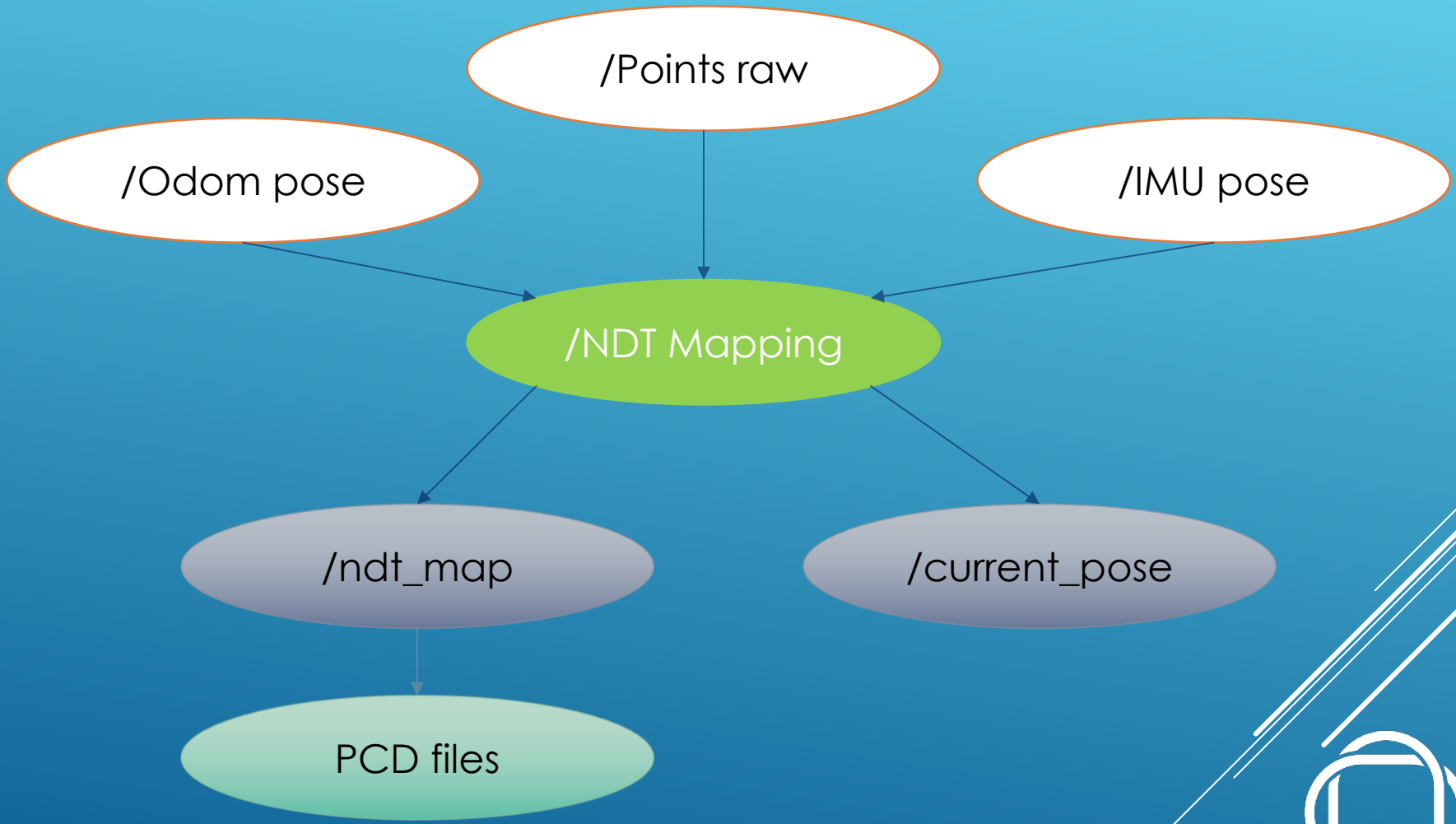


2.1 Localization in autoware-Mapping



Autoware manual of mapping, 2017

2.1 Localization in autoware-Mapping



Analysis of Autoware, 2018

2.1 Localization in autoware-Mapping

1. PCD file is in a local coordinate, you should transform the local map into a global map.

2. /map -> /world

   Autoware uses tf_mapping to do these job.

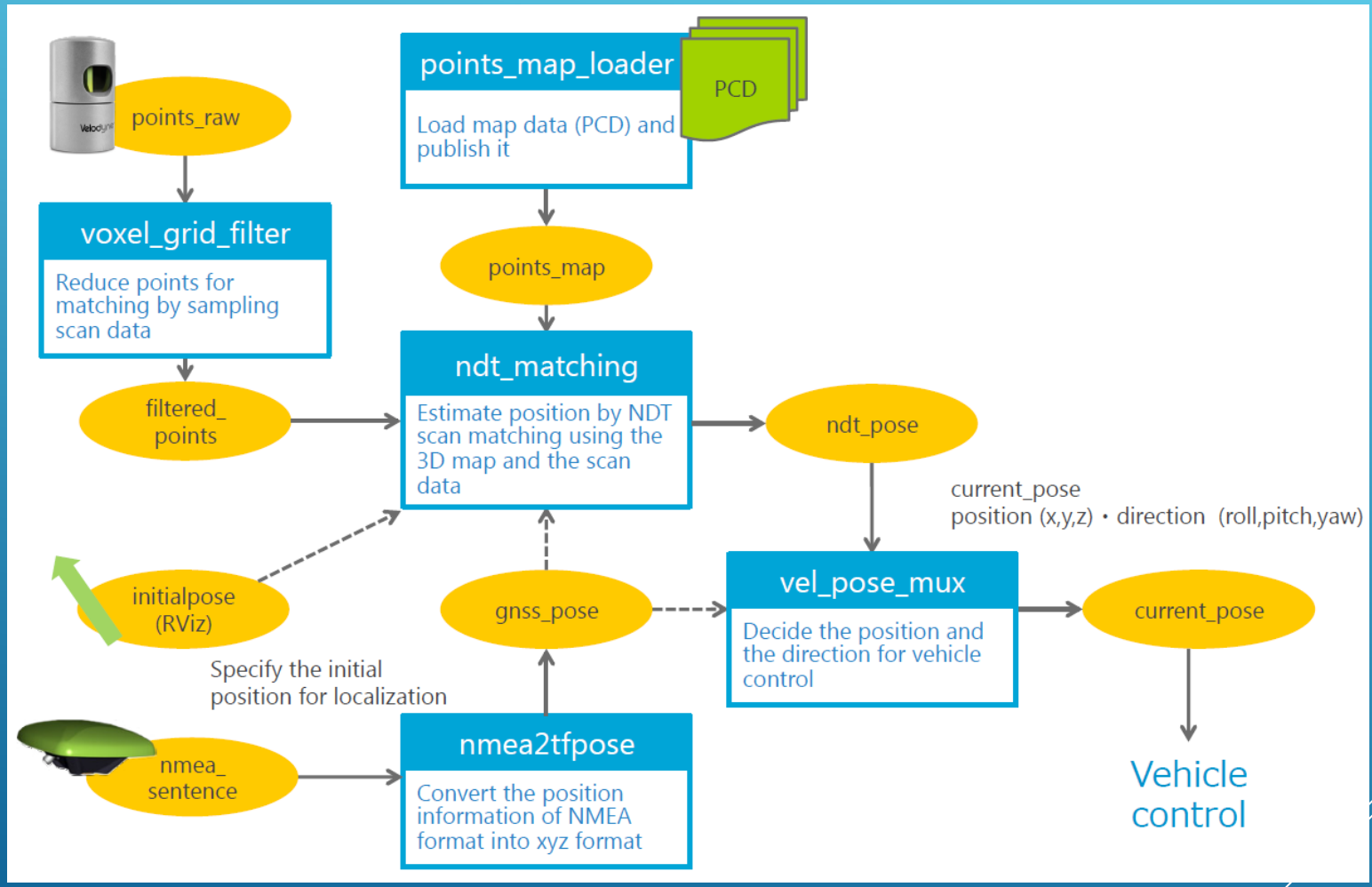
   Given a initial 6-DOF value to construct rotation and translation matrix.

3. NDT Mapping -> same problems in SLAM.

4. Without RTK, even in fixed scenes.

5. No deal with dynamic objects-noises.

2.1 Localization in autoware-Matching



2.1 Localization in autoware-Matching

1. **Input data**
   - online lidar pcd -> local coordinate
   - offline pcd map -> global coordinate
   - initial pose(6-DOF) || gnss_pose (gnss+ yaw, pitch, roll)

     gnss_pose is needed in two cases.

2. **Model in NDT matching**

   linear || quadratic || none

3. **Initial value is critical important.**

   We have tested that: with 5-10meters errors, the software crashes.

4. **Single localization scheme with simple algorithm.**

2.2 MSF in apollo

Apollo in github:

https://github.com/ApolloAuto/apollo

For understanding MSF, three papers are needed:

Stanford

2007, Map-Based Precision Vehicle Localization in Urban Environments
2010, Robust Vehicle Localization in Urban Environments Using Probabilistic Maps

BaiDu

2017, Robust and Precise Vehicle Localization based on Multi-sensor Fusion in Diverse City Scenes

2.2 Localization in apollo

Apollo 1.0

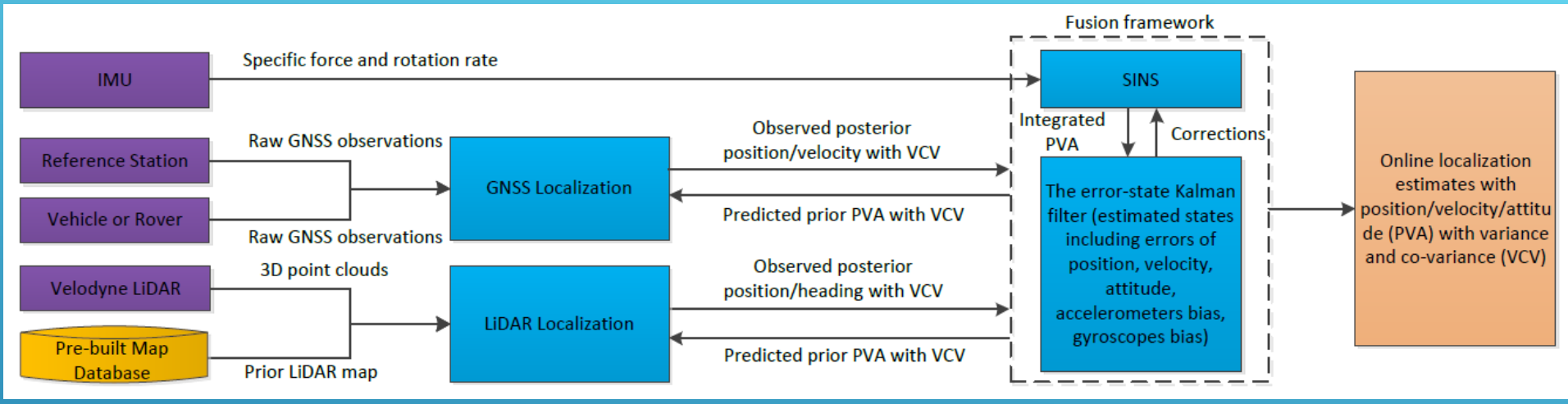
RTK method -> without memory(map)

Apollo 2.0

MSF method -> with memory(map)

Map building && localization

2.2 MSF in apollo

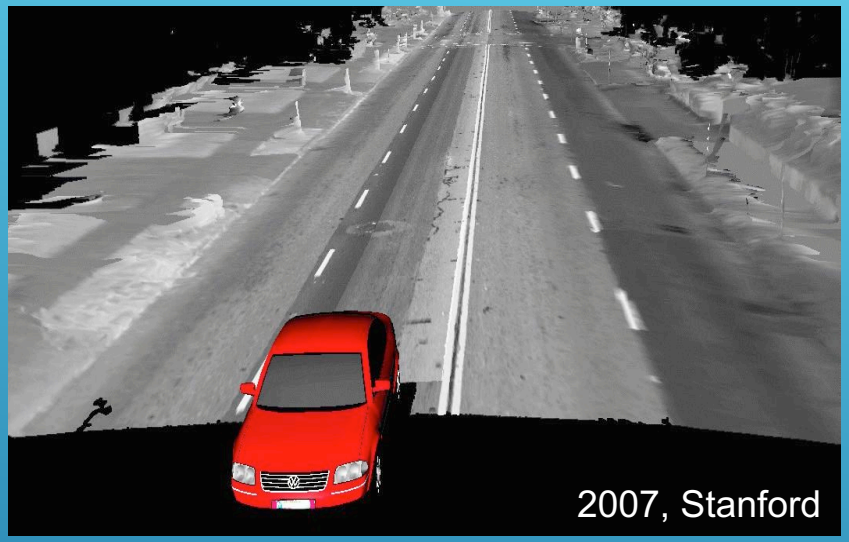


1. Lidar based localization
   - Intensity Map && Altitude Map
   - Heading Estimation <- Lucas Kanade Algorithm
   - Horizontal Localization <- Histogtam filter

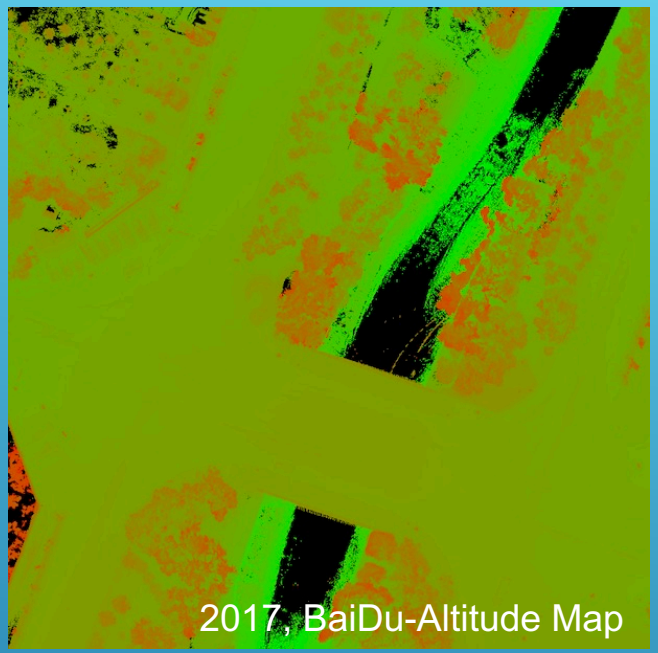
2. GNSS based localization
   - Raw data received && RTK algorithm mentioned
   - Tightly coupled with INS

3. Sensors fusion localization
   - An error-state KF is applied.

2.2 MSF in Apollo-Lidar based localization



2.2 MSF in Apollo-Lidar based localization



1. Altitude Map, 3D info is added.

2. Map Building Method

- With good GNSS signal reception

  The GNSS/INS solution based on post-processing algorithms, such as the NovAtel Inertial Explorer, is able to produce enough accurate vehicle motion trajectories.

- In weak GNSS signal scenarios

  BD treat it as a classic map reconstruction problem combining several techniques including NovAtel IE post-processing, LiDAR SLAM, loop closure, and the global pose-graph optimization.

2.2 MSF in Apollo-Lidar based localization

Algorithm 1 LiDAR-based localization

Input: Prior map m, online point cloud z, rough transformation $T_0 = (x_0, y_0, a_0, \phi_0, \theta_0, h_0)$ and search space X, Y.

Output: Best registration $(\hat{x}, \hat{y}, \hat{a}, \hat{h})$, and covariance matrix $\mathcal{C}_{xy}$.

1: $\hat{h} \leftarrow$ heading angle estimation ▷ IV-A
2: $\hat{a_0} \leftarrow m(x_0, y_0)$ ▷ Get altitude from map
3: Transform z with the transformation $(x_0, y_0, \hat{a_0}, \phi_0, \theta_0, \hat{h})$
4: **for** $x_i, y_i \in \{x_0 + X, y_0 + Y\}$ **do**
5: $P_r \leftarrow SSD_r(x_i, y_i, z, m)$ ▷ Equ. 6 8
6: $P_a \leftarrow SSD_a(x_i, y_i, z, m)$ ▷ Equ. 7 8
7: $P(z|x_i, y_i, m) \leftarrow (P_r)^{\gamma} \cdot (P_a)^{1-\gamma}$ ▷ Equ. 5 9 10
8: $P(x_i, y_i) \leftarrow P(z|x_i, y_i, m) \cdot (\bar{P}(x_i, y_i))^{1/\kappa}$ ▷ Equ. 4
9: **end for**
10: $(\hat{x}, \hat{y}) \leftarrow \{P(x_i, y_i)\}$ ▷ Equ. 11
11: $\mathcal{C}_{xy} \leftarrow \{P(x_i, y_i)\}$ ▷ Equ. 12
12: $\hat{a} \leftarrow m(\hat{x}, \hat{y})$ ▷ Get altitude from map
13: **return** $(\hat{x}, \hat{y}, \hat{a}, \hat{h}, \mathcal{C}_{xy})$

2.2 MSF in Apollo-GNSS based localization

1. A RTK solution is proposed.

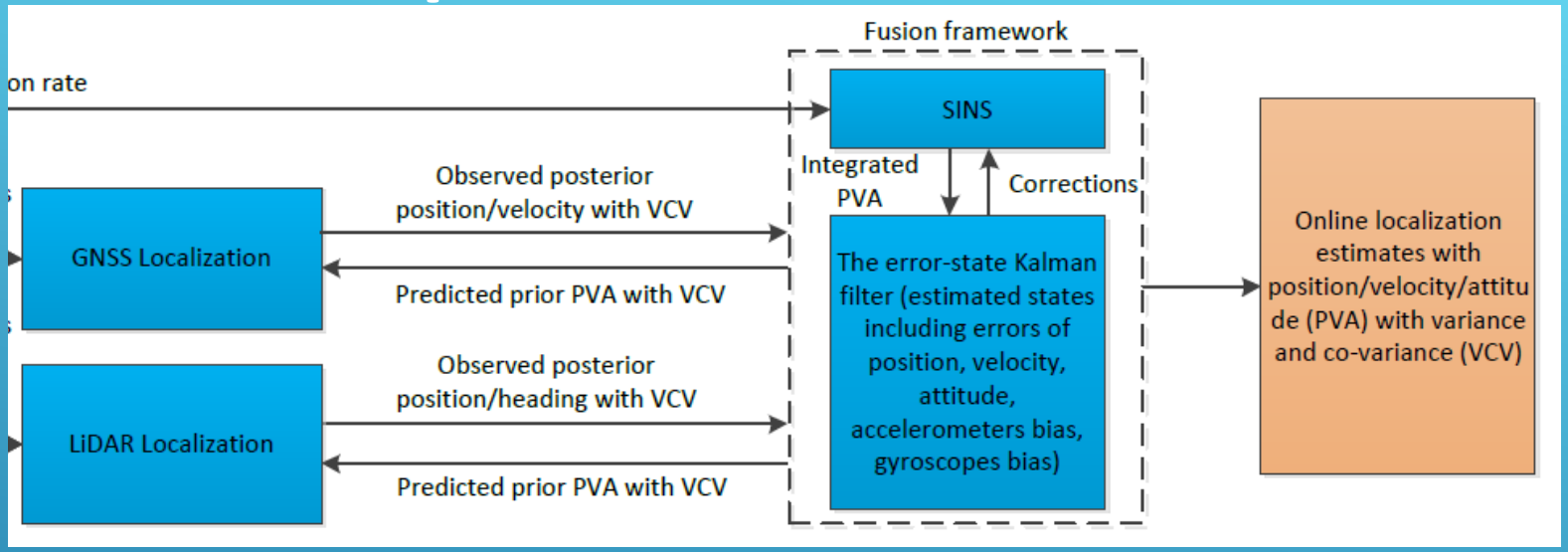
   RTK-LIB is referenced in this part.

2. A tightly coupled solution is presented.

   Currently, Baidu only loosely couples the sensor observations, which means GNSS results are come from hardware.

   Option 1: NovAtel SPAN-IGM-A1
   Option 2: NovAtel SPAN® ProPak6™ and NovAtel IMU-IGM-A1

2.2 MSF in Apollo-Fusion-Error-state KF



1. SINS Kinematics Equation And Error Equation
2. Filter State Equation
3. Measurement Update Equation
   - 3.1 LiDAR measurement update equation
   - 3.2 GNSS measurement update equation

3. Options

High Precision Localization absolutely needs memory ⇔ Map.

During building map, we should concern:

1. Which kind of map is needed.

   Raw PCD || Feature Map || Intensity Map || Altitude Map…

2. How to control map precision.

   RTK Solution && Valid Satellites Collection && MSF Algorithm…

3. How to deal with dynamic objects.

   Ground Constrained && Statistics Features…

iMorpheus.ai

3. Options

High Precision Localization absolutely needs sensors.

During online localization, we should concern:

1. Robustness && Precision && Efficiency

   GNSS/IMU && LiDAR && Cameras && MSF Algorithm

Journal Club

iMorpheus.ai Weekly Journal Club

Next Friday, 16/03/2018 12:00PM GMT+8

Hand-eye Calibration

关键词：手眼标定，Machine Vision（机器视觉），Sensor Fusion（传感器融合）

Website : **http://imorpheus.ai**
Email Address : **live@imorpheus.ai**

iMorpheus.ai