

抜粋版

はじめてのLiDAR開発キット(Livox版)

～Livox LiDARを使って自動運転, 3D地図, ドローンに活用～

開発編

LIVOX



スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales@spectrum-tech.co.jp

開発キット 目次

- ubuntu運用マニュアル
 - ubuntuについて
 - Linux基本コマンド
 - ubuntu基本操作
 - 日常運用(ウイルススキャン、更新)
- 開発キット 全体像
- 1. ハード概要
- 2. ソフト概要
- 3. Autoware開発
 - Autoware全体像
 - Autoware基本
 - ① lidar動作確認
 - ② Rosbag記録、再生
 - ③ RQT表示
 - ④ 地図作成
 - ⑤ 自己位置推定
 - ⑥ 物体検出
 - ⑦ Autowareデモ1
- 4. 3D地図開発
 - ① Livox_mapping
 - ② Cloudcompare

ページ

[4](#)

[4](#)

[6](#)

[7](#)

[9](#)

[10](#)

[11](#)

[12](#)

[13](#)

[16](#)

[18](#)

[19](#)

[20](#)

[26](#)

[28](#)

[30](#)

[36](#)

抜粋版のためページと一致しません

開発キット 目次

5. その他アプリ開発

ページ

① Livox detection sim	<u>46</u>
② Livox detection	<u>47</u>
③ Livox lane detection	<u>48</u>
④ Livox free space	<u>49</u>
⑤ Livox relocalization	<u>50</u>
⑥ horizon highway slam	<u>51</u>
⑦ loam_livox	<u>52</u>
⑧ Yolov3物体検出	<u>54</u>
⑨ Apollo	<u>55</u>
⑩ Lio_livox	<u>59</u>

6. 参考

① Drone搭載用lidar例	<u>60</u>
------------------	---------------------------

Ubuntu運用マニュアル

1. Ubuntuについて

Linuxの中でも一番シェアの高いOSです。2004年にDebian系から派生。

2. Linux基本コマンド

① システム関係

- 起動: 電源を入れると自動で起動します。
- 再起動: `$ reboot`
又は、左上のメニューの「ゲストを再起動」
- 終了: `$ shutdown`
又は、左上のメニューの「ゲストをシャットダウン」
- ログアウト `$ exit`
ルートからログアウトします
- **日本語／英語の入力切替**: 半角／全角のボタン(ESCボタンの下)

Ubuntu運用マニュアル

2. Linux基本コマンド

② ディレクトリ操作、コピー、移動、削除

masa@ubuntu:~\$ **cd** /home/masa/Documents

ディレクトリの切り替え

masa@ubuntu:~/home/masa/Documents\$ **ls** ファイルとディレクトリの表示(表示したら操作したいファイルを右クリックでコピーして操作します)

masa@ubuntu:~\$ **cp** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへコピー

masa@ubuntu:~\$ **mv** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへ移動

masa@ubuntu:~\$ **rm** ファイル名 ファイルの削除

便利な機能
rm -help
このコマンド共通(マイナスを2個とhelp)

コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべ

③ ユーザ権限、プロセス他

masa@ubuntu:~\$ **su** -

スーパーユーザ(root)に切り替え、パスワードを入力

masa@ubuntu:~\$ **sudo**

ルータ権限で各種コマンドを実施します。

masa@ubuntu:~\$ **ps** a

現状の動いているプロセスを表示

masa@ubuntu:~\$ **kill**

特定のプロセスを強制終了

masa@ubuntu:~\$ **apt-get** install pkg

パッケージのインストールなどに使用

masa@ubuntu:~\$ **date**

日付、時間の設定を行います。

masa@ubuntu:~\$ **leafpad** /etc/network/interfaces

インタフェースに記述している内容を変更します。Viよりも使いやすいです。

④ モジュール、usb、メモリ、HDDなどの表示

masa@ubuntu:~\$ **lsmod**

linuxのモジュールリスト表示

masa@ubuntu:~\$ **lsusb**

usbのデバイス表示

masa@ubuntu:~\$ **free -mt**

メモリ使用状態表示

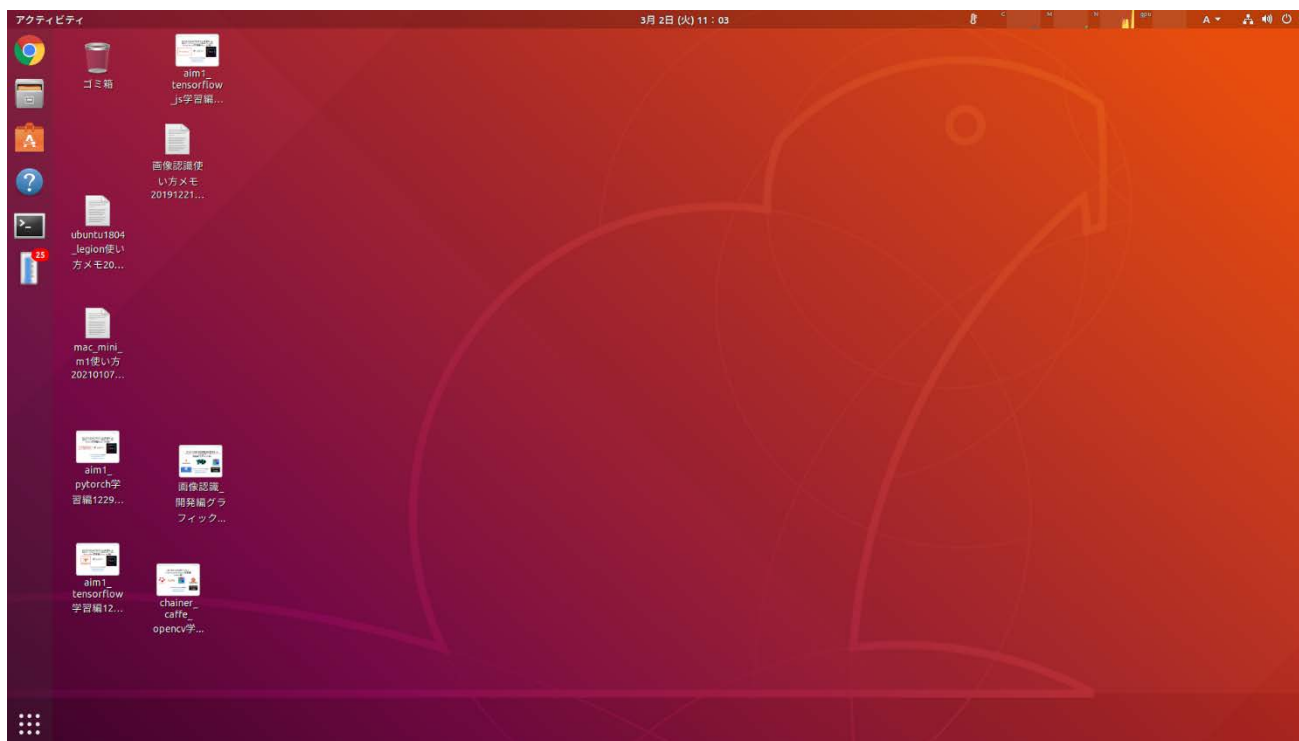
masa@ubuntu:~\$ **df**

HDD(マイクロSD)の使用状態表示

Ubuntu運用マニュアル

3. 基本操作

① 表示画面と内容



主に使用するもの

- ・ブラウザ: Chrome
- ・フォルダ: Documents内に必要なファイルがあります。
- ・コマンド: コマンド画面を立ち上げて、python3のプログラムを動作させます。

Ubuntu運用マニュアル

4. 日常運用

① セキュリティ対策(アンチウイルス更新、スキャン)

- アンチウイルス対策として無料のclamAVをインストールしてます。
- 手動での運用を基本としています。

```
masa@ubuntu: ~  
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)  
masa@ubuntu:~$ sudo freshclam  
Fri Jul 12 09:58:58 2019 -> ClamAV update process started  
2019  
Fri Jul 12 09:58:58 2019 -> ^Your ClamAV installation is O  
Fri Jul 12 09:58:58 2019 -> ^Local version: 0.100.3 Recomm  
Fri Jul 12 09:58:58 2019 -> DON'T PANIC! Read https://www.  
pgrading-clamav  
Fri Jul 12 09:58:58 2019 -> main.cvd is up to date (versio  
f-level: 60, builder: sigmgr)  
Fri Jul 12 09:58:58 2019 -> daily.cld is up to date (versi  
19, f-level: 63, builder: raynman)  
Fri Jul 12 09:58:58 2019 -> bytecode.cvd is up to date (version: 328, stgs: 94,  
f-level: 63, builder: neo)  
masa@ubuntu:~$ sudo clamscan --infected --remove --recursive
```

パターンファイル更新

```
$ sudo freshclam
```

手動スキャン時にも更新されます

手動でスキャン

```
$ sudo clamscan --infected --remove --recursive
```

自動化可能ですが、バックグラウンドで重くなる可能性大。コマンド入力後時間がかかります。

4. 日常運用

- Linuxの場合は、頻繁に更新が発生します。アップグレードを定期的 to 実施してください。
- 更新前には、バックアップを取ることをお勧めします。特にアップグレードはまれに動作不良、戻せない状態が発生します。自己責任で実施してください。

```

masa@ubuntu: ~
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
^C
masa@ubuntu:~$ sudo apt-get update
ヒット:1 http://jp.archive.ubuntu.com/ubuntu bionic InRelease
取得:2 http://jp.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
無視:3 http://dl.google.com/linux/chrome/deb/ InRelease
取得:4 http://jp.archive.ubuntu.com/ubuntu bionic-security InRelease
無視:5 https://developer.download.nvidia.com/linux-gpu/ InRelease
tu1804/x86_64 InRelease
無視:6 http://developer.download.nvidia.com/linux-gpu/ InRelease
InRelease
ヒット:7 https://developer.download.nvidia.com/linux-gpu/ InRelease
untu1804/x86_64 Release
ヒット:8 http://developer.download.nvidia.com/linux-gpu/ InRelease
64 Release
ヒット:9 http://archive.ubuntu.com/ubuntu bionic InRelease
ヒット:10 http://archive.ubuntu.com/ubuntu bionic-updates InRelease
ヒット:11 http://dl.google.com/linux/chrome/deb/ InRelease
取得:13 http://security.ubuntu.com/ubuntu bionic-security InRelease
取得:16 http://security.ubuntu.com/ubuntu bionic-security InRelease
9 kB]
取得:17 http://security.ubuntu.com/ubuntu bionic-security InRelease
50 kB]
取得:18 http://security.ubuntu.com/ubuntu bionic-security InRelease
56 kB]

masa@ubuntu:~$ sudo apt-get upgrade
パッケージリストを読み込んでいます... 完了
依存関係ツリーを作成しています
状態情報を読み取っています... 完了
アップグレードパッケージを検出しています... 完了
以下のパッケージは保留されます:
  libgl1-mesa-dri libxatracker2 mesa-va-drivers mesa-udpvr-drivers
以下のパッケージはアップグレードされます:
  firefox firefox-locale-en firefox-locale-ja gnome-settings-daemon
  gnome-settings-daemon-schemas libsystemd1 ubuntu-report
アップグレード: 7 個、新規インストール: 0 個、削除: 0 個、保留: 4 個。
54.4 MB のアーカイブを取得する必要があります。
この操作後に追加で 4,019 kB のディスク容量が消費されます。
続行しますか? [Y/n] y
取得:1 http://jp.archive.ubuntu.com/ubuntu bionic-updates/main amd64 gnome-setti
ngs-daemon-schemas all 3.28.1-0ubuntu1.3 [12.9 kB]
取得:2 http://jp.archive.ubuntu.com/ubuntu bionic-updates/main amd64 gnome-setti
ngs-daemon amd64 3.28.1-0ubuntu1.3 [316 kB]
取得:3 http://jp.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsystemd1
cs1 amd64 1.3.2 [1,475 kB]
取得:4 http://security.ubuntu.com/ubuntu bionic-security/main amd64 firefox amd6
4 68.0+build3-0ubuntu0.18.04.1 [49.8 MB]

```

```
更新リスト取得
$ sudo apt-get update
アップグレード実施
$ sudo apt-get upgrade
```


LiDAR開発キット 全体像(Livox版)

PCハードウェア

CPU



cpu,16GB RAM

+

GPU

Nvidia RTX30、20
シリーズ等に対応



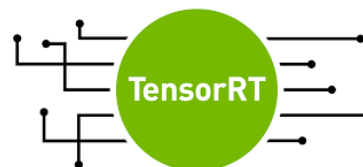
USBメモリ



OS



関連ソフト系



Livoxハードウェア

オプションで提供
します。

Mid-70



Avia



Horizon

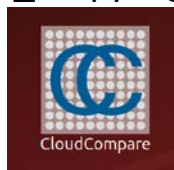


自動運転系



3D地図系

SLAM
Livox_mapping
Lio



プログラム言語

python



1. ハードウェア概要

①必要なハードウェア仕様

ハードウェアの概要です。

区分		プロダクツ	メーカー	備考
USBメモリ (livox版)		128GB USB3.0 livox関連ソフト		
お客様準備品				
PC本体	cpu	Intel i9, i7, amd Ryzen 9,7など cpu		GPUが搭載できるものに限り ます。
	GPU	RTX30,20シリーズ	nvidia	
	メモリ	16GB以上		
	SSD	512GB以上		
LiDAR	Mid-70	Detection Range (@ 100 klx):260 m @ 80% reflectivity FOV:70.4° (Circular) Angular Precision (1σ):< 0.1° Beam Divergence:0.28° (Vertical) × 0.03° (Horizontal) Point Rate:100,000 points/s (first or strongest return)	Livox	弊社からの提供も可能で す。 livox製の他の製品も対応 可能です。検証は、mid-70 で実施しました。
	Avia	Detection Range (@ 0 klx):450 m @ 80% reflectivity FOV:Non-repetitive scanning pattern: 70.4° (Horizontal) × 77.2° (Vertical) Angular Precision (1σ):< 0.05° Beam Divergence:0.28° (Vertical) × 0.03° (Horizontal) Point Rate:240,000 points/s (first or strongest return)		

2. ソフトウェア概要

①ソフトウェア一覧

ソフトウェアの概要です。

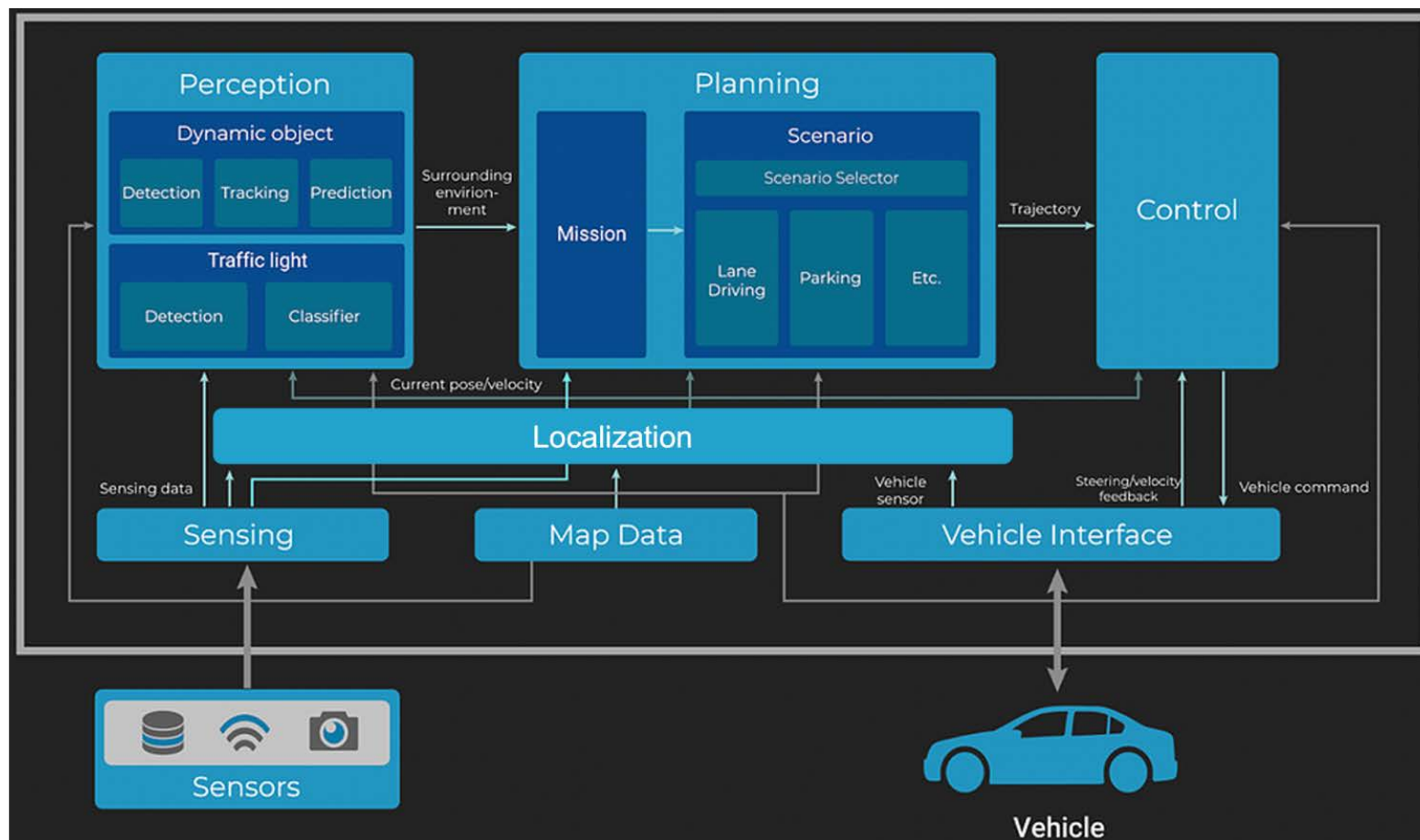
- 弊社からのマニュアルに基づいてインストールします。
- OS,GPUは、お客様で実施してください。

区分	ソフト名	バージョン	備考
OS	ubuntu	18.04 LTS	
GPU用	cuDNN	8.4.2+cuda11.4	cuda11ではautowareの一部が動作しません。
プログラム言語	python3	3.6.9	
Livox関係	Livox-sdk	2.3.0	
	Livox-ros-driver	2.6.0	
	Livox-autoware-driver	Versionなし	
自動運転	Autoware	1.14	docker版で使用
	ROS	melodic	
	apollo	5.5	
コンテナ関連	Docker-ce	>19.03.5	
	Nvidia-driver	>465	
AI用プログラム	tensorflow	1.15	仮想化等で使用
各種アプリ	Livox_mapping, detection, cloudcompareなど多数		

3. Autoware開発 全体像(Autoware)

AutowareはLinuxとROSをベースとした自動運転システム用オープンソースソフトウェアです。名古屋大学、長崎大学、産総研による共同成果の一部として、自動運転の研究開発用途に無償で公開しました。レーザレーダ、カメラ、GNSSなどの環境センサを利用して、自車位置や周囲物体を認識しながら、カーナビから与えられたルート上を自律走行できます。

<https://www.autoware.org/>



3. Autoware開発

① Autoware基本: lidar動作確認

https://github.com/Livox-SDK/livox_autoware_driver

Autoware起動: 毎回実施

```
cd /home/masa/Documents/lidar/livox_autoware_ws/livox_autoware_driver
```

```
./run_livox.sh
```

dockerに入り、autoware@ubuntu2:/home/autoware\$が起動

```
roslaunch runtime_manager runtime_manager.launch
```

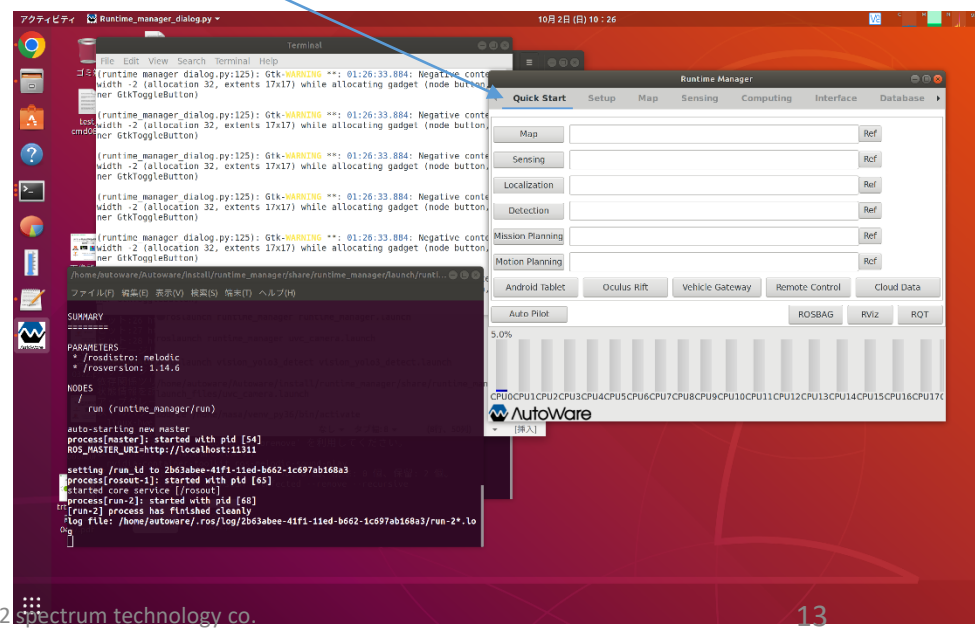
マネージャ起動でok

```
コマンド入力
$ cd
/home/masa/Documents/lidar/livox_autoware_ws/livox_autoware_driver
$ ./run_livox.sh
$ roslaunch runtime_manager runtime_manager.launch
```

Dockerなどの設定については、設定編参照

```
autoware@ubuntu2: /home/autoware
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)

apolloauto/apollo      localization_volume-x86_64-latest
109001137d4a 3 years ago 5.44MB
apolloauto/apollo      local_third_party_volume-x86_64-latest
5df2bf3cc4b9 3 years ago 156MB
apolloauto/apollo      yolo3d_volume-x86_64-latest
6a9cbf71163e 4 years ago 275MB
apolloauto/apollo      map_volume-sunnyvale_loop-latest
36dc0d1c2551 4 years ago 906MB
masa@ubuntu2:~$ cd /home/masa/Documents/lidar/livox_autoware_ws/livox_autoware_driver
masa@ubuntu2:~/Documents/lidar/livox_autoware_ws/livox_autoware_driver$ ./run_livox.sh
Using options:
  ROS distro: melodic
  Image name: autoware/livox
  Tag prefix: latest
  Cuda support: on
  Pre-release version: off
  UID: <1000>
Launching autoware/livox:latest
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
autoware@ubuntu2:/home/autoware$
```



3. Autoware開発

② Autoware基本: Rosbag記録、再生 Rosbag再生

Setup>Livoxを選択、TFを押す

Sensing>Livox lidar **チェックを外すこと**。二重表示になる。

Simulation>Ref: autoware>shared_dir>autoware-20221005xxx.bagを選択

Playを押し、topics:sensor_msgs/pointclouds2が表示されていればokで、その後pause
Rvizタグ

global option>fixed frame:**livox_frame**

sensing>points raw>topic:**/livox/lidar**を選択

Simulation

pause外してplay又は、最初からplayでもok

録画した時間が下部に表示

録画したlidar画像が出ればok

Gpu未搭載だと
動かないかも

The screenshot displays the Autoware development environment. On the left, the 'Runtime Manager' dialog is open, showing the 'Simulation' tab. The 'Ref' field is set to '/home/autoware/shared_dir/autoware-20221005062747.bag'. The 'Rate' is set to 17%. The 'Start Time' is 0. The 'Stop' button is highlighted. Below the dialog, the terminal shows the command 'roslaunch autoware_simulation autoware_simulation.launch' and the output 'Starting ROS master...'. On the right, the 'RViz' window is open, showing the 'Global Options' panel. The 'Fixed Frame' is set to 'livox_frame'. The 'Sensing' panel shows the 'Points Raw' topic selected. The 'Status' is 'Ok'. The 'RViz' window also shows the 'Livox Lidar' topic selected. The bottom status bar shows 'ROS Time: 1664951350.17', 'ROS Elapsed: 6.71', 'Wall Time: 1664951588.36', 'Wall Elapsed: 38.81', and '31 fps'.

3. Autoware開発

④ Autoware基本:地図作成

3D地図作成:ndt_mapping

Setup>Livoxを選択、TFを押す

Sensing>Livox lidarチェック

Rosbagでデータを記録、保存

computing>ndt_mapping

Simulation>roslaunchのデータをplay, pause

Rvizタグ

本来は、点群データをダウンロードする機能がndt_mappingのappにはありますが、動作しません。他のツール等で実施が必要

File>open config:

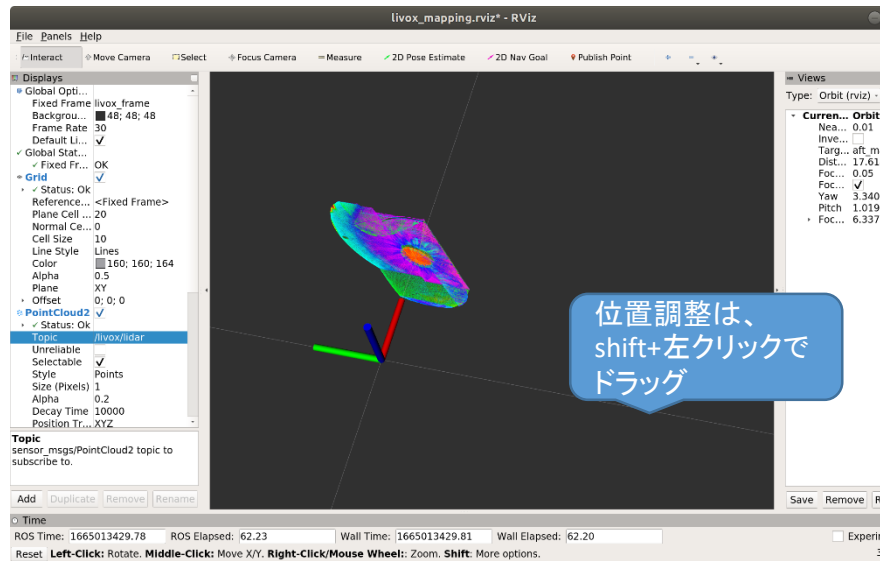
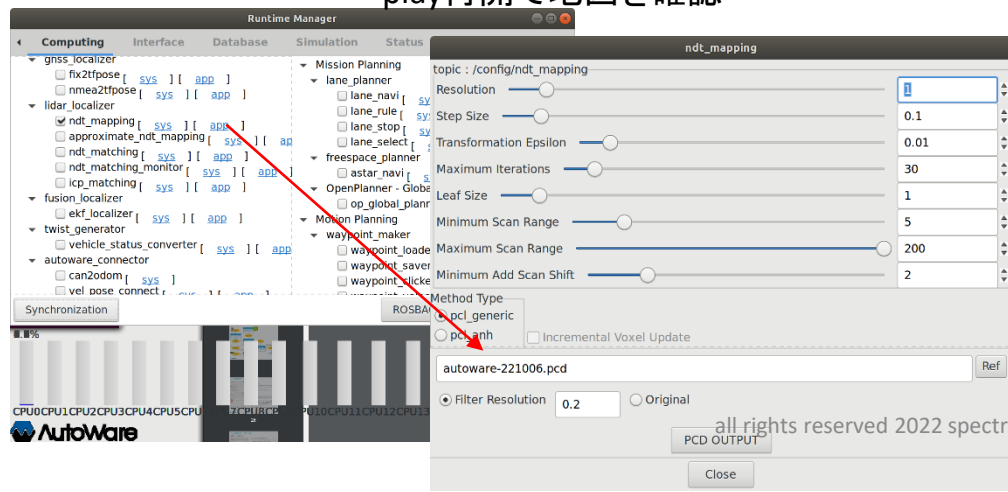
/home/autoware/Autoware/src/drivers/awf_drivers/livox_ros_driver/livox_ros_driver/c
onfig/ livox_mapping.rviz を選択

global option>fixed frame:livox_frame

pointcloud2>:/livox/lidarを選択

play再開で地図を確認

同じフォルダ内の
ndt_mapping.rvizではうまく表示しない。特にglobal
option:mapの場合、不明



3. Autoware開発

⑤ Autoware基本: 自己位置推定

3D地図作成: ndt_matching

Setup>Livoxを選択、TFを押す

map>refでpcdデータを選択し(100個等全て選択)、point cloudを押し、読み込む、TFも押す

Sensing>Livox lidarチェック、point downsample:random_filterチェック

Computing>ndt_matchingをチェック、app:initial posをチェック

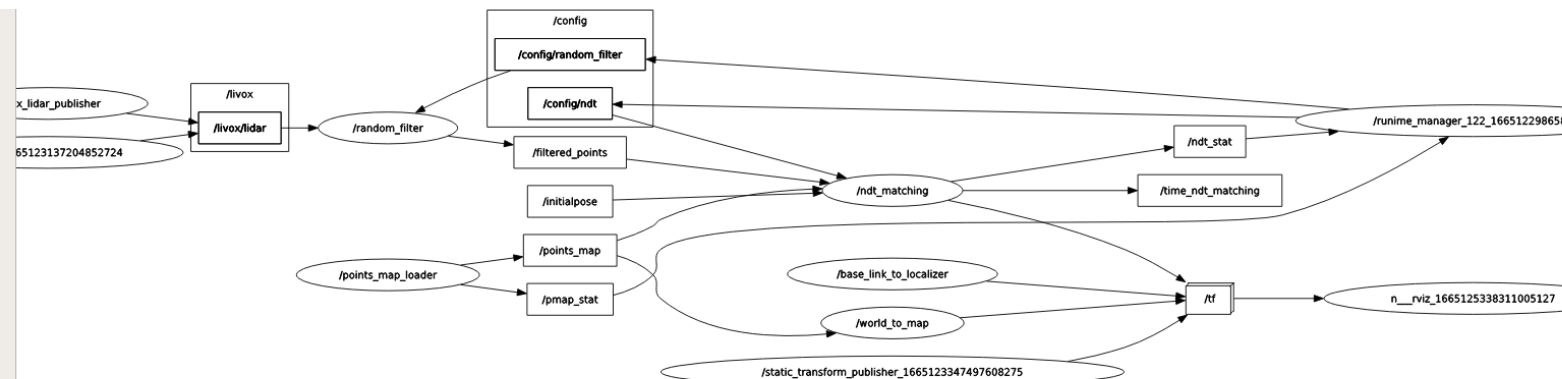
Simulation>rosbagのデータをplay, pause

Rvizタグ

fixed frame>map, map>points_map,sensing>filtered points: livox_frameとmapのリンクがないのでエラー

\$ rosrun tf2_ros static_transform_publisher 0 0 0 0 0 0 map livox_frame 別端末でリンク付けする
play再開で地図とlidarのsensingの双方を確認

RQT図



4. 3D地図開発

① Livox_mapping

地図変換: SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト

https://github.com/Livox-SDK/livox_mapping

起動

```
source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash
```

端末毎に実施のこと

RosbagからPCDに変換

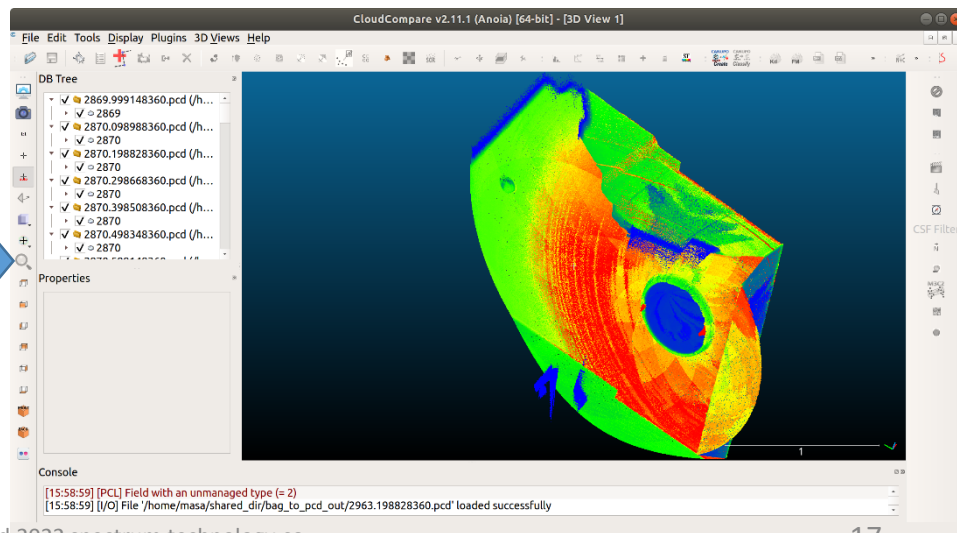
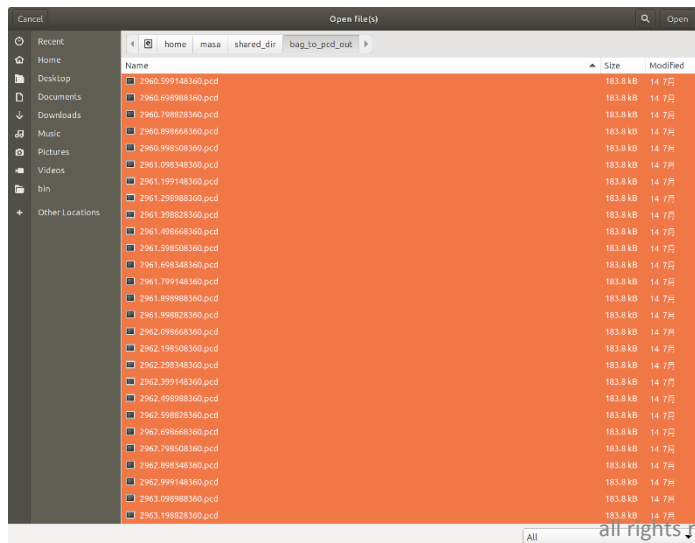
```
roscore          coreを起動
```

```
roslaunch pcl_ros bag_to_pcd test.bag /livox/lidar bag_to_pcd_out
```

pcdの確認その1

Cloudcompareをアプリから起動

File>open: /home/masa/shared_dir/bag_to_pcd_out:全ファイル選択



4. 3D地図開発

① Livox mapping

https://github.com/Livox-SDK/livox_mapping

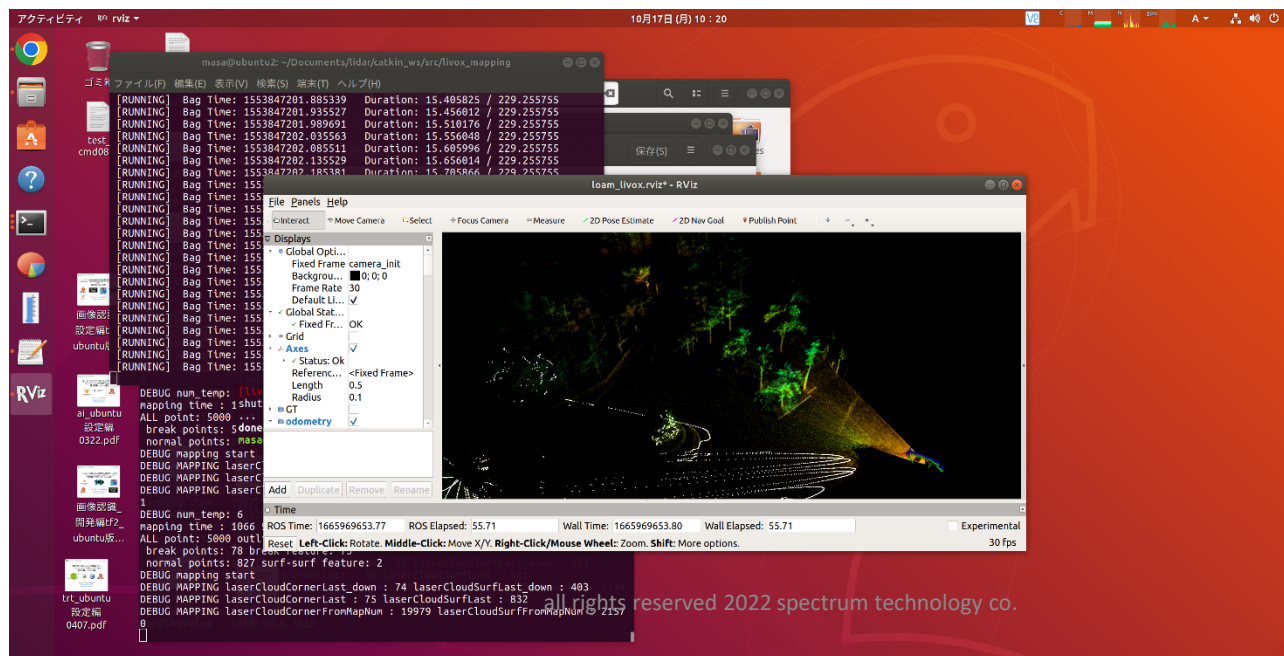
SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト
起動

source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash 端末毎に実施のこと
デモ1

roslaunch livox_mapping mapping_mid.launch

cd /home/masa/Documents/lidar/catkin_ws/src/livox_mapping

rosbag play CYT_02.bag



4. 3D地図開発

① Livox mapping

https://github.com/Livox-SDK/livox_mapping

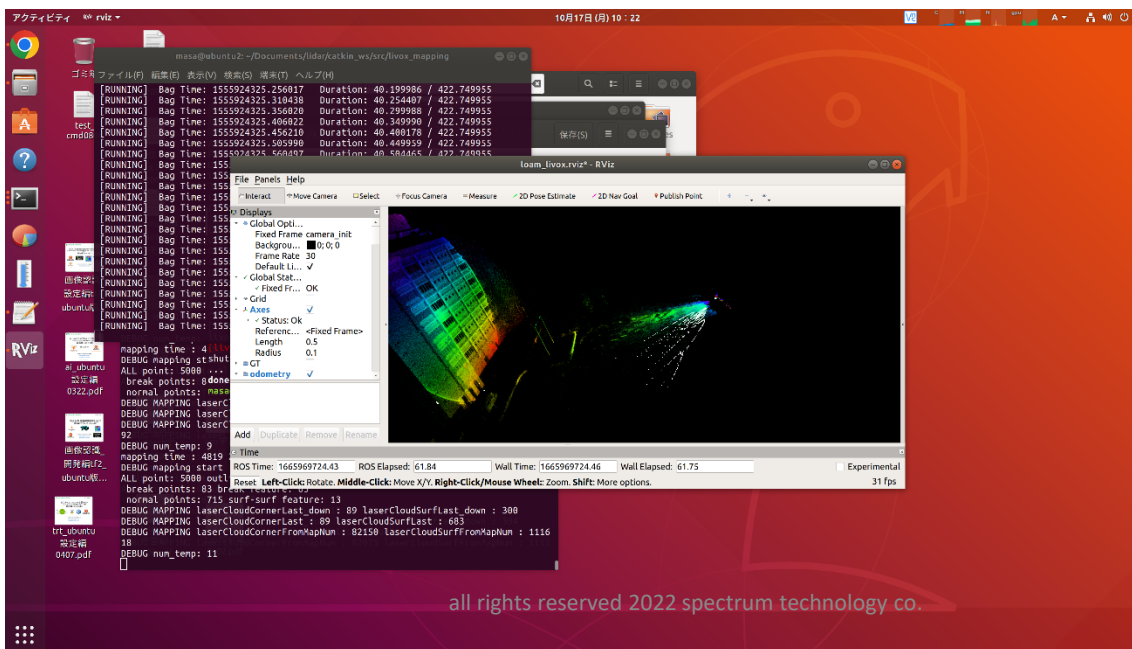
SLAM (Simultaneous Localization And Mapping)に使用する地図作成ソフト
起動

source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash 端末毎に実施のこと
デモ2

roslaunch livox_mapping mapping_mid.launch

cd /home/masa/Documents/lidar/catkin_ws/src/livox_mapping

rosbag play HKUST_01.bag



4. 3D地図開発

② cloudcompare

Pcdの点群データの加工、比較などを行うオープンソース

<https://www.danielgm.net/cc/>

<https://hdtopography.github.io/learning/cloudcompare/CloudCompare.html> 東大の人が解説
起動

Cloudcompareをアプリから起動

File>open: /home/masa/shared_dir/bag_to_pcd_out:全ファイル選択:例

基本操作

セグメンテーション(切り出し、断面)

合成

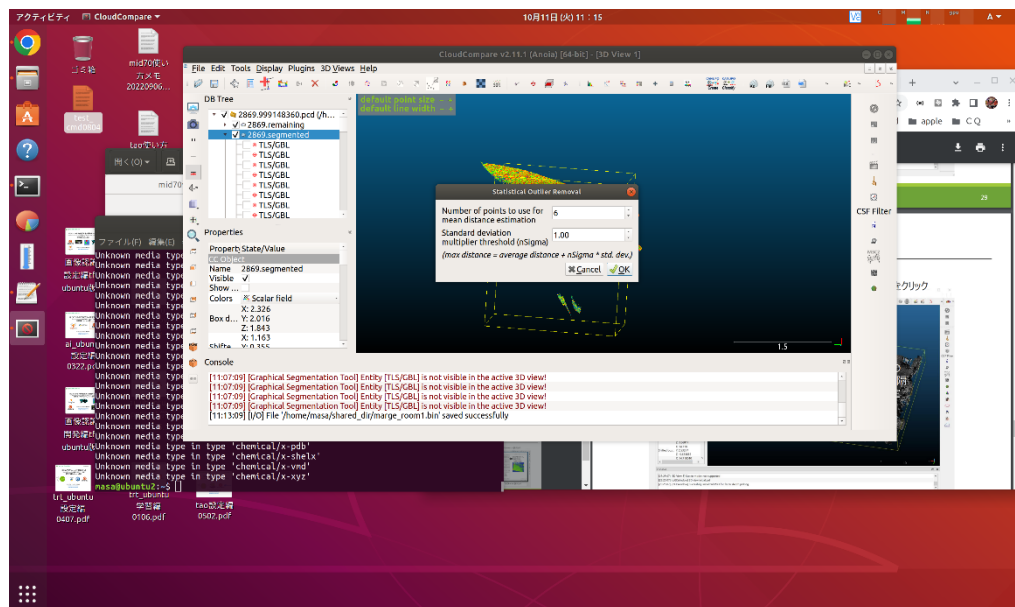
ノイズ除去

リサンプリング

メッシュ化

クラウド間距離

Pcdから他の形式に変換:plyへ



4. 3D地図開発

② cloudcompare

Pcdの点群データの加工、比較などを行うオープンソース

<https://www.danielgm.net/cc/>

<https://hdttopography.github.io/learning/cloudcompare/CloudCompare.html> 東大の人が解説
起動

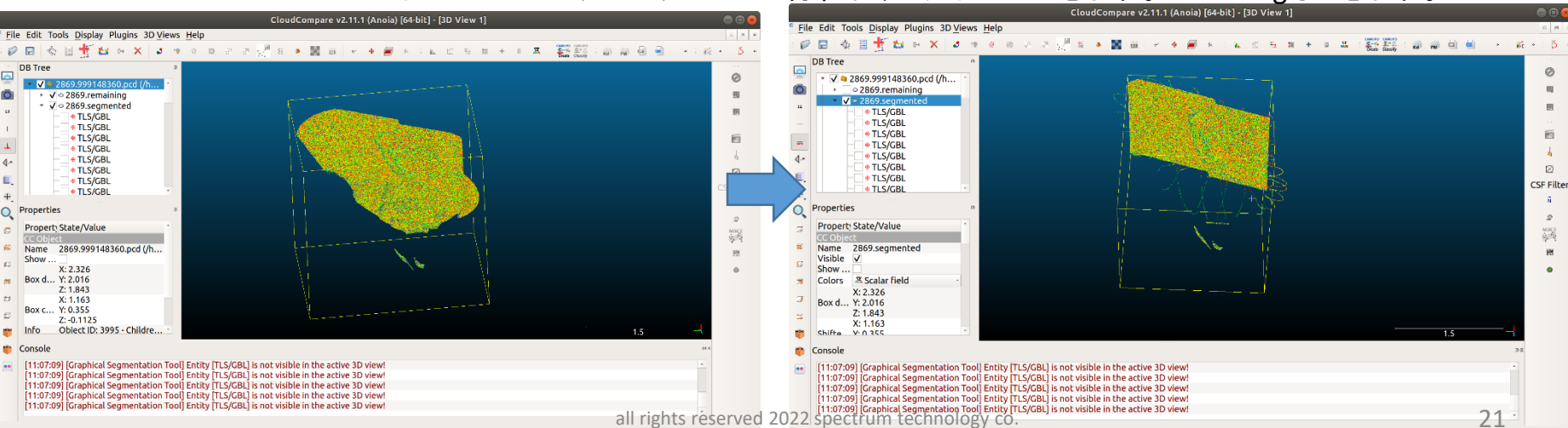
Cloudcompareをアプリから起動

File>open: /home/masa/shared_dir/bag_to_pcd_out:全ファイル選択:例

基本操作

セグメンテーション(切り出し、断面)

- セグメンテーションする前に読み込んだ、pcdの複数ファイルをマージします。
- 切り出し: 上のはさみのアイコンで、エリアを左クリック、で選択して切り出し。終了は右クリック
- アイコンの✓マークでセグメントが行われ、ファイルができます。remainingもできます。



4. 3D地図開発

② cloudcompare

Pcdの点群データの加工、比較などを行うオープンソース

<https://www.danielgm.net/cc/>

<https://hdtopography.github.io/learning/cloudcompare/CloudCompare.html> 東大の人が解説
起動

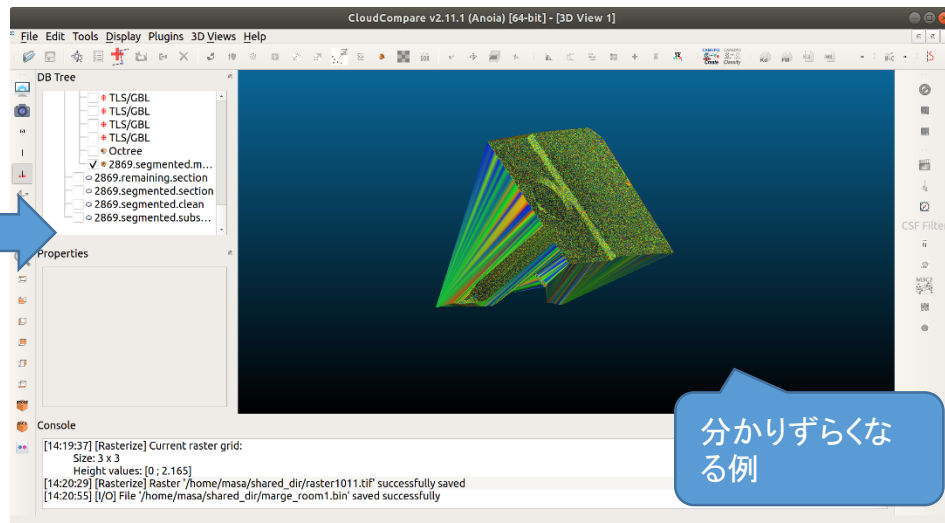
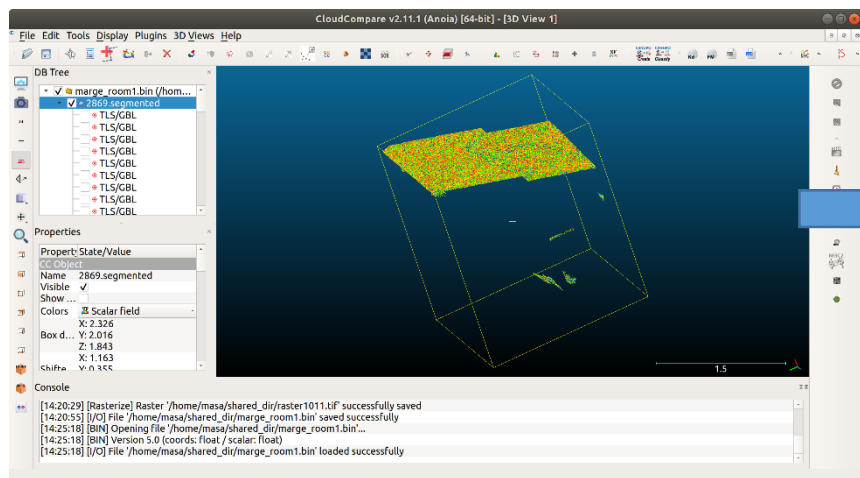
Cloudcompareをアプリから起動

File>open: /home/masa/shared_dir/bag_to_pcd_out:全ファイル選択:例

基本操作

メッシュ化:点群を3角形で結んで面の集合体で分かりやすくします。(点群のデータによります)

- Edit>mesh: Delaunay 2.5D(best fit)



4. 3D地図開発

② cloudcompare

Pcdの点群データの加工、比較などを行うオープンソース

<https://www.danielgm.net/cc/>

<https://hdttopography.github.io/learning/cloudcompare/CloudCompare.html> 東大の人が解説
起動

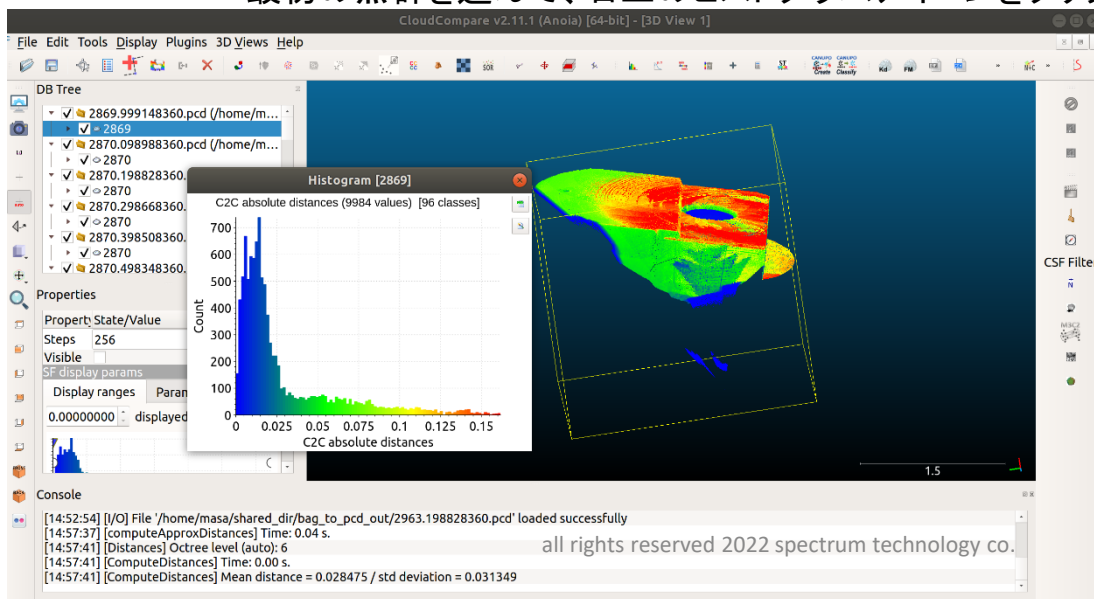
Cloudcompareをアプリから起動

File>open: /home/masa/shared_dir/bag_to_pcd_out:全ファイル選択:例

基本操作

クラウド間距離:ふたつの点群を選んで、距離を測定,二つ目はCTL+左クリックで選択

- Tools>distance>cloud to cloud
- 最初の点群を選んで、右上のヒストグラムアイコンをクリック



X軸: distanceの単位
はmm?
誤差の分布が分かります

5. その他アプリ開発

① Livox detection sim

https://github.com/Livox-SDK/livox_detection_simu

車を検知するデモ

起動

```
source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash
```

端末毎に実施のこと

デモ

```
roscore
```

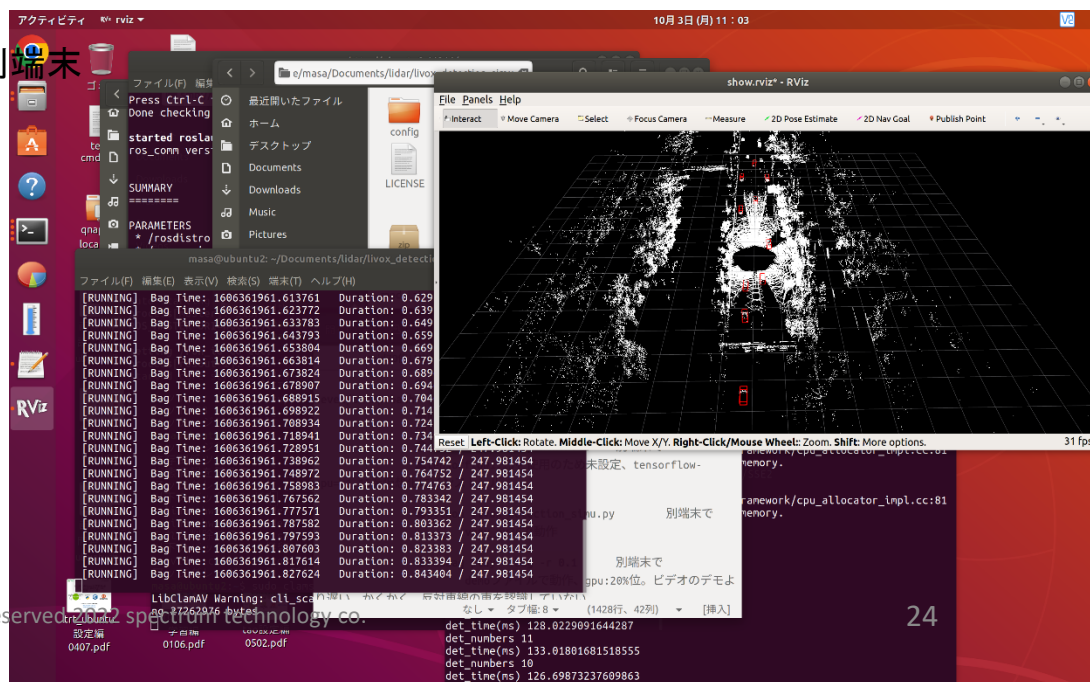
```
cd /home/masa/Documents/lidar/livox_detection_simu
```

```
rviz -d ./config/show.rviz 別端末
```

```
python livox_detection_simu.py 別端末
```

```
roslaunch *.bag -r 0.1 別端末
```

Tensorflow-gpuで設定したが
遅い?? 不明



5. その他アプリ開発

④ Livox free space

https://github.com/Livox-SDK/livox_free_space

駐車場の空を検知

起動

`source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash`

デモ

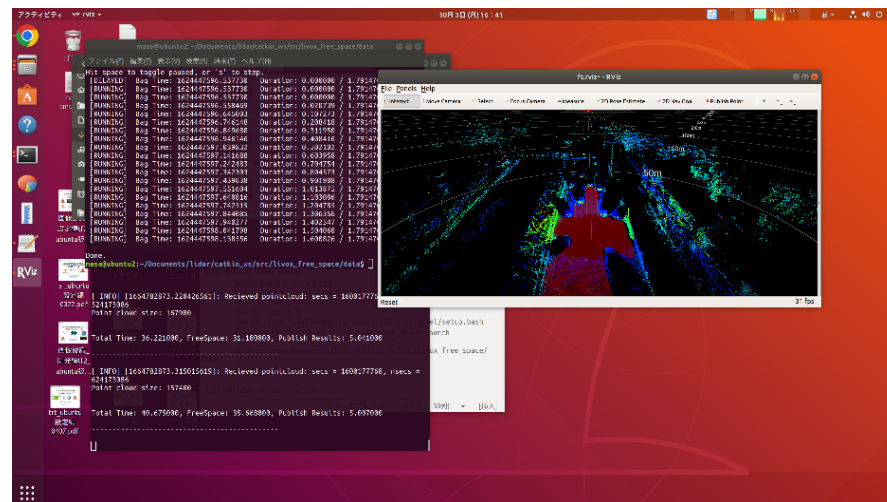
`roslaunch livox_free_space livox_free_space.launch`

`cd /home/masa/Documents/lidar/catkin_ws/src/livox_free_space/data`

`rosbag play demo.bag` 別端末で

Lidarを動作させて表示させる場合

`roslaunch livox_ros_driver livox_lidar.launch`



5. その他アプリ開発

⑤ Livox relocalization

https://github.com/Livox-SDK/livox_relocalization

作成した地図で自動運転し、車線追尾するデモ

地図ファイル設定変更

```
cd /home/masa/Documents/lidar/catkin_ws/src/livox_relocalization/launch/livox_relocalization.launch
```

```
map_file_pathのvalue変更 /home/masa/Documents/lidar/catkin_ws/src/livox_relocalization/map
```

起動

```
source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash
```

デモ

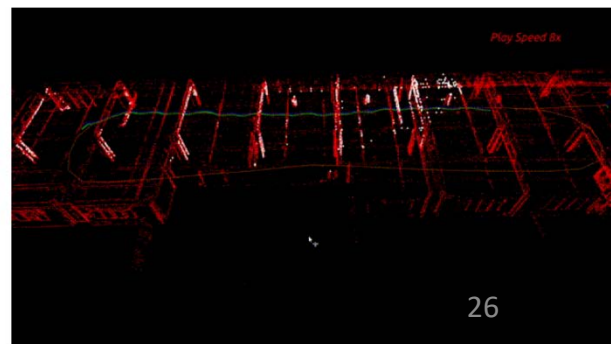
```
roslaunch livox_relocalization livox_relocalization.launch
```

```
cd /home/masa/Documents/lidar/catkin_ws/src/livox_relocalization
```

```
rosbag play mid40_re_example.bag
```

別端末で

• Line track example:



5. その他アプリ開発

⑦ loam_livox

https://github.com/Livox-SDK/loam_livox

LiDAR Odometry and Mapping (LOAM)でリアルタイムに地図作成し、自分の位置を特定する。
起動

```
source /home/masa/Documents/lidar/catkin_ws/devel/setup.bash
```

デモ

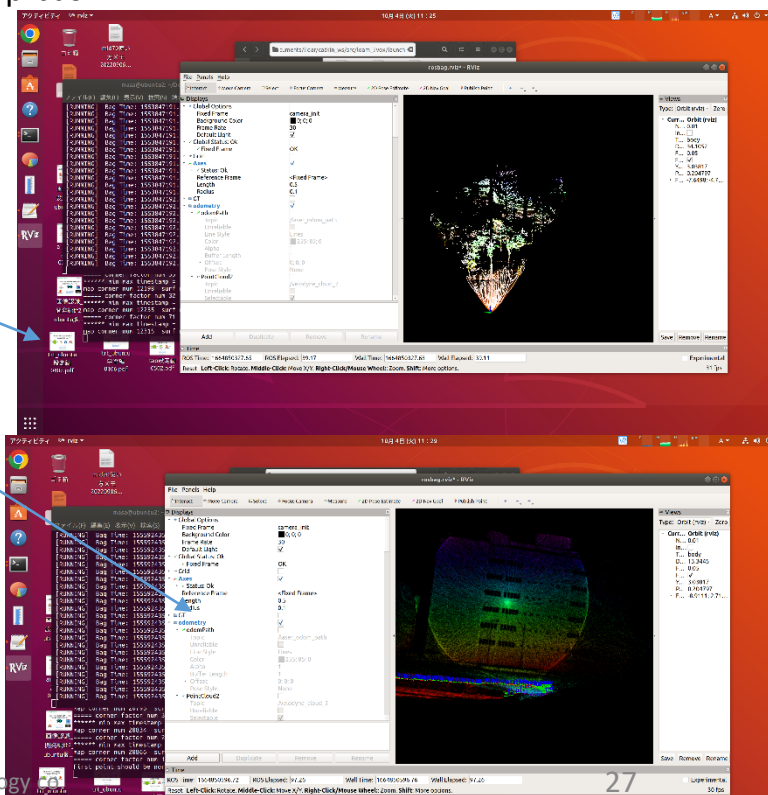
```
roslaunch loam_livox rosbag.launch
```

```
cd /home/masa/Documents/lidar/catkin_ws/src/loam_livox
```

```
rosbag play CYT_02.bag 別端末で
```

```
roslaunch loam_livox rosbag_largescale.launch
```

```
rosbag play HKUST_01.bag 別端末で
```



5. その他アプリ開発

⑨ apollo

baiduなどが推奨している中国版の自動運転

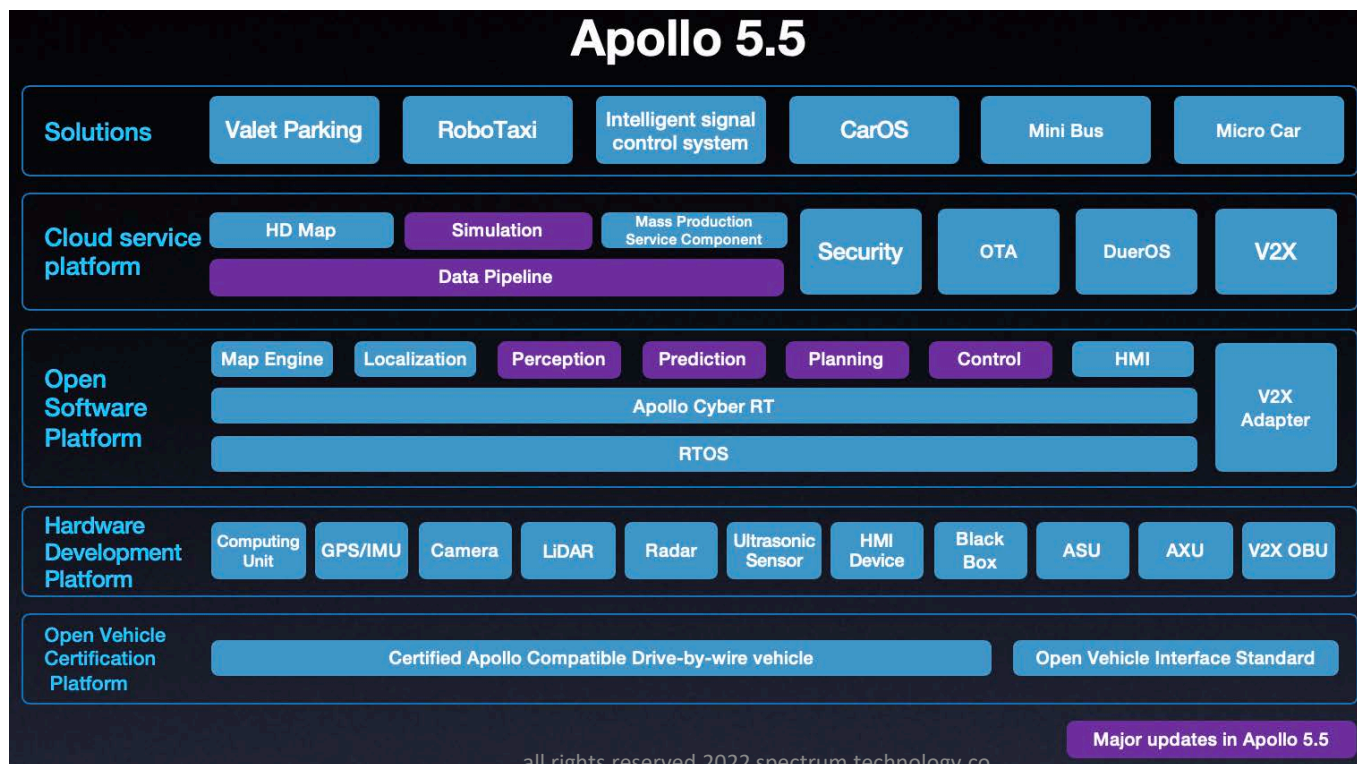
<https://github.com/Livox-SDK/apollo>

<https://www.apollo.auto/>

<https://github.com/ApolloAuto/apollo>

全体像

中国国内にしか公開されていない情報もあり。
最新のv7.0ではlivoxが動作しません。V5.5を推奨



5. その他アプリ開発

⑩ Lio_livox

LiDAR-Inertial Odometry: 慣性オドメトリを用いて移動ロボットの軌跡推定と地図作成をリアルタイムで実現するフレームワーク

<https://github.com/Livox-SDK/LIO-Livox>

livox horizon、Hap lidarのみ対応、mid-70はlivox-mappingを使用のこと。

未検証

