2022.2

**自分のPCでLivox mappingを行うための備忘録**

千葉大学大学院園芸学研究院

生物環境気象学研究室

**0. 使用環境**

* LiDARセンサ: Livox mid-70
* PCスペック(ノートPC):

CPU: Intel core i5 7300U

メモリ: DDR3-1866 4GB\*2 = 8GB

GPU: Intel HD Graphics 620

OS: Windows 10 Pro

* PCスペック(デスクトップPC)

CPU:

メモリ:

* マッピングを長時間行う場合、メモリは8GB以上必要。

**1. Ubuntu 18.04のインストール (デュアルブート)**

Windows10には気軽にLinuxを使えるWSLという機能があるが、Livox mappingはうまく動作しないようなのでWindowsとは別にUbuntuをインストールする必要がある。

なお、NVIDIAのグラフィックボードを搭載している場合、起動画面が正しく表示されずインストールできない可能性がある(対処方法は5.3参照)。

1.1 Ubuntu 18.04 LTSの64bit版isoファイルを入手

* 有志による日本語版もあるが[本家](https://releases.ubuntu.com/18.04/?_ga=2.181431527.1761933874.1623307409-1453293.1623307409)からダウンロードするのが無難？
* "Ubuntu-18.04.5-desktop-amd64.iso"をクリックしてダウンロードする。

1.2 インストールメディアの作成

* ダウンロードしたisoファイルはDVDかUSBメモリに記録する必要がある。
* DVDに記録する場合は空のDVDをセットし、isoファイルを右クリック→[ディスクイメージの書き込み]から作成する。
* USBメモリに記録する場合は[Rufus](https://rufus.ie/ja/)を使用すると良い。

1.3 デュアルブートの構築　[参考](https://www.pc-koubou.jp/magazine/35542)

* あらかじめ，インストール予定のドライブに空き容量があるか確認しておく。データを外部ストレージに保存するつもりなら推奨環境の25GB程度あれば足りると思われる。このときパーティションの分割を行う必要はない。
* DVDもしくはUSBメモリをPCに挿入し、電源を入れる。メーカーのロゴが表示されたらBIOS起動キー(だいたいF2かDeleteあたり)を押してBIOSに入る。
* 普通にWindowsが立ち上がってしまう場合は[電源とスリープ]→[電源の追加設定]→[電源ボタンの動作の選択]から高速スタートアップを無効化する。
* BIOSを開くとだいたい[Boot]の項目があり、そこでDVD/USBメモリの起動順位を一番上にする、もしくはブートデバイスとして選択することでUbuntuを起動する。
* うまく起動しない場合は[Fast Boot], [Secure Boot]の設定を無効化する。
* 起動後、[インストールする]をクリックして設定を進める。インストールの種類では必ず[Ubuntuをwindowsとは別にインストール]とすること。[ディスクを削除してUbuntuをインストール]を選択するとpcの全データが消去されるので注意する。
* Ubuntuに割り当てるディスク容量は最低25GB程度あれば大丈夫そうだが、余裕があれば多めにしておいたほうが良い。
* 設定が終了すると再起動(何故か自動で電源が切れない)してインストールが完了する。

**2. プラットフォーム・ソフトウェアのインストール**

端末を開き(ショートカットキー: ctrl+Alt+T)以下のコマンドのコピペ、実行(Enter)を繰り返す。

有線LANを使ったほうがスムーズに進行することが多い。

2.1 ROS Melodic　[参考](https://demura.net/robot/16518.html)

sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb\_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

sudo apt install -y curl

curl -sSL 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xC1CF6E31E6BADE8868B172B4F42ED6FBAB17C654' | sudo apt-key add -

sudo apt update

sudo apt upgrade

sudo apt install -y ros-melodic-desktop-full

sudo apt install python-rosdep

sudo rosdep init

rosdep update

cd

echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc

source ~/.bashrc

sudo apt install -y python-rosinstall python-rosinstall-generator python-wstool build-essential python-catkin-tools

2.2 Point Cloud Library　[参考](https://qiita.com/tewi_r/items/941acb2af690f8f184a1)

sudo apt install libpcl-dev

sudo apt install cmake

2.3 Eigen　[参考](http://eigen.tuxfamily.org/index.php?title=Main_Page)

cd

git clone <https://gitlab.com/libeigen/eigen.git>

2.4 OpenCV　[参考](https://cocoinit23.com/ubuntu-opencv-install/)

sudo apt-get install libopencv-dev python3-opencv

2.5 Livox SDK　[参考](https://github.com/Livox-SDK/Livox-SDK)

cd

sudo apt install cmake

git clone https://github.com/Livox-SDK/Livox-SDK.git

cd Livox-SDK

cd build && cmake ..

make

sudo make install

2.6 livox\_ros\_driver　[参考](https://github.com/Livox-SDK/livox_ros_driver)

cd

git clone https://github.com/Livox-SDK/livox\_ros\_driver.git ws\_livox/src

cd ws\_livox

catkin\_make

source ./devel/setup.sh

2.7 livox\_mapping　[参考](https://github.com/Livox-SDK/livox_mapping)

cd

mkdir -p catkin\_ws/src

cp -r ~/ws\_livox/src/livox\_ros\_driver ~/catkin\_ws/src/

cd ~/catkin\_ws/src

git clone https://github.com/Livox-SDK/livox\_mapping.git

cd ..

catkin\_make

source ~/catkin\_ws/devel/setup.bash

**3. その他の詳細設定**

3.1 Launchfileの設定(pcd/bagファイルの保存場所指定)

* 任意の場所に点群データの保存場所を作る(例: /home/<ユーザー名>/testfiles)。
* ~/catkin\_ws/src/livox\_mapping/launch/mapping\_mid.launchを開く。
* 9行目を以下のように変更する:

<param name="map\_file\_path" type="string" value="/home/<ユーザー名>/testfiles" />

* ~/catkin\_ws/src/livox\_ros\_driver/launch/livox\_lidar.launchを開く
* 11行目を以下のように変更する:

<arg name="rosbag\_enable" default="true"/>

* 39行目を以下のように変更する:

<node pkg="rosbag" type="record" name="record" output="screen" args="-a -O /home/<ユーザー名>/testfiles/test.bag"/>

3.2 有線LAN設定

* 以下のように有線LANを設定する

[IPV4]→ IPV4メゾッド: 手動

アドレス: 192.168.1.50 ネットマスク: 255.255.255.0 ゲートウェイ: 192.168.1.1

3.3 撮影モード設定

* catkin\_ws/src/livox\_ros\_driver/configにあるlivox\_licar\_config.jsonを開く。
* “return\_mode”の数字を変更

0: single first return

1: strongest single return

2: dual return

3.4 反射強度出力をRGBから元の値に変更する

* 出力される点には0-255(256段階)の反射強度(Reflectivity)情報が付与されている。
* デフォルトでは反射強度はRGB(カラー)に変換されて出力される。視覚的にはわかりやすいが分析を行うためには再び元の反射強度に戻す必要があり面倒なため、初めから元の値を出力するようプログラムを変更する。
* catkin\_ws/src/livox\_mapping/srcにあるlaserMapping.cppを開く。
* 230行あたりからpcdのRGB出力に関する記述があり、300行目付近を次のように書き換える。

po->r = reflection\_map;

po->g = reflection\_map;

po->b = reflection\_map;

* 次にこの変更を反映させるためコンパイルする。
* cd ~/catkin\_ws
* catkin\_make
* ちなみにコンパイル先はcatkin\_ws/build/livox\_mapping/CMakefiles/loam\_laserMapping.dir/src/laserMapping.cpp.o

**4. 撮影とマッピング**

4.1 現地で撮影しながらマッピング

* mid-70、Livoxコンバータ、電源、PCをそれぞれ接続し、mid-70が起動していることを確認する。
* 端末を使い、以下のコマンドを実行する。

cd ~/catkin\_ws/src

source ~/catkin\_ws/devel/setup.bash

roslaunch livox\_mapping mapping\_mid.launch

* rvizが立ち上がる。端末にctrl+Zを入力しプロセスを一時停止し、以下のコマンドを実行する。

roslaunch livox\_ros\_driver livox\_lidar.launch

* マッピングが開始し、rvizにその様子が描画される。
* マッピングを終了する場合は端末を閉じる。その後、3.1で作成した保存場所に.pcdファイルと.bagファイルが保存される。
* ファイルはこのフォルダ内に上書きされるため、続けて撮影する場合は別のフォルダへの移動や名前の変更を忘れないようにする。

4.2 bagファイルからマッピング

* 以前に生成した.bagファイルを用いてマッピングを行う。
* 端末を使い、以下のコマンドを実行する。

cd ~/catkin\_ws/src

source ~/catkin\_ws/devel/setup.bash

roslaunch livox\_mapping mapping\_mid.launch

* rvizが立ち上がる。端末にctrl+Zを入力しプロセスを一時停止し、以下のコマンドを実行する。

rosbag play <bagファイルのパス・ファイル名>.bag

* マッピングが開始し、rvizにその様子が描画される。
* マッピングを終了する場合は端末を閉じる。その後、3.1で作成した保存場所に.pcdファイルが保存される。
* ファイルはこのフォルダ内に上書きされるため、続けて撮影する場合は別のフォルダへの移動や名前の変更を忘れないようにする。

**5. 補足**

5.1 pcdファイルの処理

* pcdファイルを扱えるフリーソフトとして[CloudCompare](https://www.danielgm.net/cc/)が有名。

5.2 Livox viewerで取得したデータを用いる

* Livox viewerはlivox LiDARセンサが取得した点群データを簡単に閲覧・保存できるソフトウェアであり、[こちら](https://www.livoxtech.com/jp/downloads)からダウンロードできる。
* Ubuntuの他にwindows版も配布されているため、ubuntu搭載パソコンを使わずに点群データ(.lvx形式)を取得できる(保存例: /home/<ユーザー名>/lvx/test2.lvx)
* マッピングを行うにはまず.lvxファイルを.bagファイルに変換する必要があるため、以下のコマンドを実行する。

cd ws\_livox

source ./devel/setup.sh

roslaunch livox\_ros\_driver lvx\_to\_rosbag.launch lvx\_file\_path:="/home/<ユーザー名>/lvx/test2.lvx /"

* .bagファイルが生成されるので、4.2の通りにマッピングを行う。

5.3 Nvidiaグラフィックボード搭載PCへのインストール

* Ubuntuのインストールメディアを起動しようとすると黒い画面に線や壊れた文字のようなものが表示されたきり動かなくなってしまうことがある。これはグラフィックボードのドライバが正常に動作していないためである。
* BIOSからUbuntuのインストールメディアを起動する際に立ち上げる方法を選択する画面が現れるので、[Install Ubuntu]を選択した状態で”E”キーを押し、起動コマンド編集画面に入る。
* “quiet splash”と書かれている部分を”nomodeset”に書き換え、F10を押して起動する。
* 一応画面が表示されるようになるので、インストールを進める。
* 解像度が最適化されていないため設定画面が見切れてしまうことがある。このときWindowsキーを押しながらウィンドウをドラックすると端まで表示させることができる。
* インストールが完了し再起動した後も同様に”E”キーを押して起動コマンド編集画面に入り、“quiet splash”と書かれている部分を”nomodeset”に書き換え、F10を押して起動する。
* 以下のコマンドを実行してドライバを更新する。

sudo apt update

sudo apt upgrade

sudo ubuntu-drivers autoinstall

* 起動コマンドを変更せず再起動し、正常に表示されていることを確認する。

5.4 SLAMと実行環境について

* ~~内蔵グラフィックの場合はマッピング中にrvizのフレームレートがすぐ低下する。ただし強力なグラボを積んだPCでマッピングすれば成功率が上がるというわけではないため、SLAMの演算自体にGPU性能はあまり関与していないと思われる。単にrviz上で大量の点群を描画するためにGPU性能が必要なだけか。~~
* マッピングで生成される点群の量(密度)はPCの性能が高いほど多くなる。同じBAGファイルを様々なPCでマッピングしたところ判明。

例) 2021/6/22撮影のBAGファイルでマッピング

1. ノートPC(Core i5 7300U)で実行→　PCDファイルサイズ150MB
2. デスクトップPC(Core i7 4770, 内蔵GPU)で実行→　PCDファイルサイズ290MB
3. デスクトップPC(Core i9 9900k, RTX2080)で実行→　PCDファイルサイズ350MB

* rosbagの再生速度を遅くするとマッピングで生成される点群の量(密度)は多くなる。再生速度の変更は実行時末尾に -r <実行速度> を追加するだけ。

rosbag play <bagファイルのパス・ファイル名>.bag -r 0.5

この場合、再生速度は0.5倍になる。

例) 2021/7/27撮影のBAGファイルでマッピング

1. 通常速度で実行→　PCDファイルサイズ463MB
2. 0.5倍速で実行→　PCDファイルサイズ758MB
3. 2.0倍速で実行→　PCDファイルサイズ341MB