インストールマニュアル

ロボット DB システム

3 次元点群データのボクセル化

発行日 2021 年 3 月 31 日
公立大学法人会津大学
株式会社東日本計算センター

目　　次

1．はじめに 1
1.1．動作環境 1
1.2．使用機器 2
2．システムの導入 3
2.1. PCL 3
2.2. Octomap 3
2.3. octomap_tutor 4
2.4. octomap-tools 4
2.5. Choreonoid 5

1. はじめに

ロボット DB システム(以下、本システム)は、異種複数ロボットの内部、外部データ等を LICTiA サーバー上に配置された共通のデータベース(以下、DB)に蓄積し、そのデータが利用できる環境を提供します。本書では 3 次元点群データをボクセル化する手順を示します。本システムは会津大学 産学連携ロボット研究開発支援事業の一環として開発したものです。

1.1. 動作環境

動作環境一覧は以下のとおりです。推奨 CPU, GPU, メモリ, ストレージは Choreonoid 公式ドキュメント[1]に記載されています。

表 1-1 動作環境一覧

環境		バージョン
OS	Ubuntu	18.04 LTS
CPU	Intel Core シリーズ、AMD Ryzen シリーズ、コア数 4 以上、ベース周波数 3GHz 程度	-
GPU	NVIDIA GeForce / Quadro、Intel HD Graphics (CPU 内蔵)	-
メモリ	8GB 以上推奨	-
ストレージ	20GB 以上推奨	-
開発言語	Python	2.7.17
	C++	C++11
依存ライブラリ	Choreonoid	開発版
	OpenCV	3 以上
	Boost	1.65.0
	PCL	1.8.1
	Octomap	1.9.0
	octomap_tutor	最新版
	octomap-tools	1.8.1

[1] https://choreonoid.org/ja/manuals/latest/wrs2018/overview.html#wrs2018-overview-simulator

1.2.使用機器

表 1-2 使用機器一覧

No	機器名称	個数	補足
1	PC	1	-

2. システムの導入

2.1. PCL

3 次元点群データ(PCD ファイル)の操作に必要とします。

```
$ cd (ダウンロード先)
$ wget --no-check-certificate https://github.com/PointCloudLibrary/pcl/archive/pcl-1.8.1.tar.gz
$ tar -xvzf pcl-1.8.1.tar.gz
$ cd pcl-pcl-1.8.1
$ mkdir build && cd build
$ cmake -DCMAKE_BUILD_TYPE=Release ..
$ make -j2
$ sudo make install
```

2.2. Octomap[2]

3 次元点群データをボクセル表現するために必要とします。Version 指定は特にありませんが、Ver1.9.0 をインストールする例を以下に示します。make 時のコア数指定は、ご使用のスペックに合わせて指定してください。**なお、お手持ちの環境に ROS かつ octomap の ROS パッケージ[3]も導入されている場合はインストール不要です。**

```
$ cd (ダウンロード先)
$ wget --no-check-certificate https://github.com/OctoMap/octomap/archive/v1.9.0.tar.gz
$ tar -xvzf v1.9.0.tar.gz
$ cd octomap-1.9.0
$ mkdir build && cd build
$ cmake ..
$ make -j2
```

[2] https://github.com/OctoMap/octomap

[3] https://wiki.ros.org/octomap

2.3. octomap_tutor[4]

3 次元点群データを Octomap(オクトツリーなどのデータ構造を扱えるライブラリ)で扱えるデータ形式に変換します。octomap_tutor は、位置座標＋色情報＋透明度情報を持った点群データを bt(Binary OcTree ファイル)形式に変換することができます。Readme に記載されているとおり、octomap_tutor を利用するには、先にインストールした PCL と octomap、そして OpenCV と Boost のインストールが必要です。

```
$ sudo apt update
$ sudo apt install libopencv-dev libboost-all-dev

$ cd (ダウンロード先)
$ git clone https://github.com/gaoxiang12/octomap_tutor.git
$ cd octomap_tutor
$ mkdir build && cd build
$ cmake ..
$ make -j2
```

2.4. octomap-tools

Octomap で扱えるデータの編集、比較、変換等ができるツールです。本書では bt 形式から VRML 形式に変換するツールを利用します。

```
$ sudo apt update
$ sudo apt install octomap-tools
```

[4] https://github.com/gaoxiang12/octomap_tutor

2.5. Choreonoid[5]

(1) ホームディレクトリに移動します。

```
$ cd ~
```

(2) GitHub から Choreonoid をクローンします。

```
$ git clone https://github.com/choreonoid/choreonoid.git
```

(3) Choreonoid ディレクトリに移動します。

```
$ cd choreonoid
```

(4) 依存パッケージのインストールを行います。

```
$ misc/script/install-requisites-ubuntu-18.04.sh
```

(5) ビルドディレクトリを作成＆移動します。

```
$ mkdir build && cd build
```

(6) Cmake によるビルド設定を行います。-D オプションで設定可能ですが、本書では GUI で設定します。

```
$ ccmake ..
```

[5] https://choreonoid.org/ja/manuals/latest/wrs2018/setup.html#choreonoid

(7) トップ画面が表示されたら、c キーを押下します。

図 2 - 1 ccmake Top 画面(configure 未実施)

(8) 様々なオプションが表示されるので、変更したい項目を上下カーソルキーで選択し、
Enter キーで ON に切り替えます。変更したい項目は以下です。
ENABLE_CORBA、BUILD_CORBA_PLUGIN

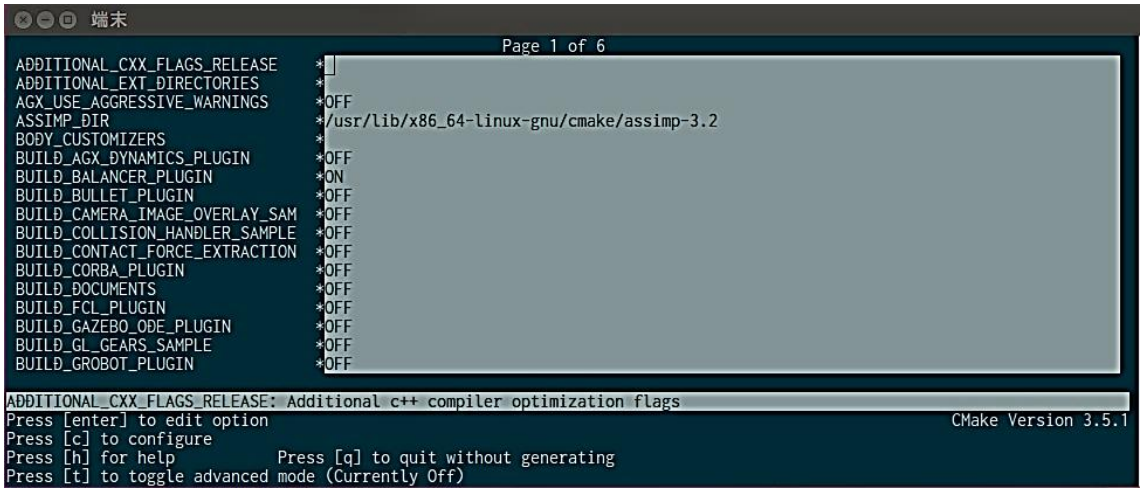

図 2 - 2 ccmake Top 画面(設定項目構成後)

(9) 指定完了したら、c キーを押下します。その後 g キーを押下することで設定内容反映完了後にターミナルに戻ります。

(10) ビルドをします。マシンがマルチコア CPU であれば、-j オプションでビルドを並列化できます。マシンに搭載されている CPU の論理コア数を指定します。以下はコア数が 4 の場合です。

```
$ make -j4
```

(11) インストールします。

```
$ sudo make install
```

(12) Choreonoid を起動します。特にエラーが無ければインストール完了です。

```
$ choreonoid
```

※Choreonoid は開発継続中ですので、必要に応じて最新版に更新してください。

```
$ cd ~/choreonoid
$ git pull
$ cd build
$ cmake ..
$ make -j$(nproc)
$ sudo make install
```

著作権

本文書の著作権は公立大学法人 会津大学に帰属します。
この文書のライセンスは以下のとおりです。
クリエイティブ・コモンズ 表示 2．1日本
http://creativecommons.org/licenses/by/2.1/jp/