

物体位置姿勢推定コンポーネント
CPU 版マニュアル

名城大学メカトロニクス工学科
ロボットシステムデザイン研究室

2016 年 11 月 24 日

内容

| | | |
|-------|-------------------------|----|
| 1. | はじめに | 3 |
| 1.1 | コンポーネント概要 | 3 |
| 1.2 | 関連文書 | 3 |
| 1.3 | 開発環境 | 3 |
| 2 | RTC 仕様 | 4 |
| 2.1 | インターフェース仕様 | 4 |
| 2.2 | サービスポートのインタフェース情報 | 5 |
| 2.2.1 | MoswlInfoService | 5 |
| 2.3 | 座標系 | 6 |
| 2.4 | 参照画像 | 7 |
| 2.4.1 | 参照画像作成方法 | 7 |
| 2.4.2 | 参照画像設定記述方法 | 7 |
| 3 | RTC の導入・動作確認 | 10 |
| 3.1 | OpenCV3.1 の導入 | 10 |
| 3.2 | 導入 | 10 |

1. はじめに

1.1 コンポーネント概要

本コンポーネントは予め取得した画像をモデルとして，アビアランスベースの任意物体の検出及び位置・姿勢の推定を行う．

本コンポーネントは，第 12 回システムインテグレーション部門講演会(SI2011)で，岩根らが投稿した「音声認識による物体認識システムの開発」に用いられているコンポーネントの一つである，ObjectSensor の CPU 改良版である．

[注意]

このコンポーネントで用いられている，SIFT アルゴリズムは特許化されています．詳しくは下記の URL をご覧ください．

[Demo Software: SIFT Keypoint Detector](#)

1.2 関連文書

本書に関連する文章を以下に示す．

| No. | 文書名 |
|-----|---------------------------------|
| 1 | 音声認識による物体認識システムの開発 SI2011 1K3_4 |

1.3 開発環境

本 RTC の開発環境を以下に示す．

| | |
|-----------|--------------------|
| OS | Windows7 |
| Compiler | Visual Studio 2013 |
| RT ミドルウェア | OpenRTM-aist-1.1.0 |
| OpenCV | OpenCV3.1 |

2 RTC 仕様

2.1 インターフェース仕様

| RTC の名称 | | | |
|-------------------|--------------------|---|--|
| ObjectSensor | |  | |
| 入力ポート | | | |
| 名称 | データ型 | 説明 | |
| SingleImahe | TimedCameraImage | カメラでキャプチャした RGB 画像を入力する. | |
| 出力ポート | | | |
| 名称 | データ型 | 説明 | |
| ObjectPose | TimedDoubleSeq | カメラ座標系における物体の位置・姿勢を同次変換行列により出力する. | |
| ResultImg | TimedCameraImage | 検出及び位置姿勢推定結果を描画した RGB 画像を出力する. | |
| サービスポート | | | |
| 名称 | インターフェース型 | 説明 | |
| Model | AcceptModelService | 物体モデル ID を指定する | |
| 主なコンフィグレーションパラメータ | | | |
| 名称 | データ型 | デフォルト値 | 説明 |
| CfgName | string | /cfg/demo.cfg | 参照画像設定ファイル名及びパスを指定する |
| SIFT_Ethreshold | double | 10 | カメラ画像に対する SIFT 特徴量の設定項目 |
| Display | string | On | 物体の検出及び位置推定結果を描画した画像の表示・非表示を設定する “on”:表示 “off”:非表示 |
| ObjName | string | None | 物体名を記述することで, 使用する参照画像を指定する |

2.2 サービスポートのインタフェース情報

2.2.1 MoswllInfoService

ModelInfoService インタフェースでは, 物体モデルの ID を指定することで, 使用参照モデルを変更することが出来る. これにより, 外部の他コンポーネントが所望物体の位置・姿勢を取得することが出来るようになる.

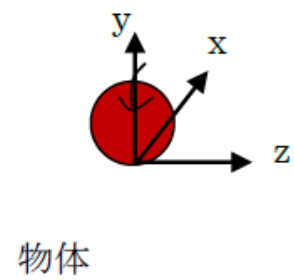
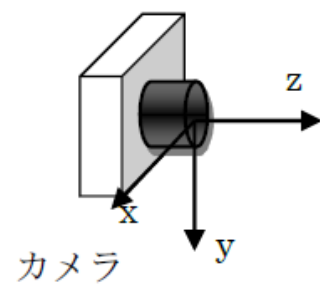
本インタフェースが定義されている IDL ファイルは以下のとおりである.

```
// interface for model provider
interface AcceptModelService{
    typedef sequence<octet> OctetSeq;

    // set model
    // @param model: a model of something
    void setModel(in OctetSeq model);
};
```

2.3 座標系

物体の位置・姿勢を表現するカメラ座標系と物体座標系は下図のようにになっている。



2.4 参照画像

本コンポーネントは物体表面を平面と仮定した参照画像を用いて、検出及び位置・姿勢推定を行う。参照画像の物体領域やカメラと物体との関係といった、参照画像に関する設定を参照画像設定ファイルへ記述する。

2.4.1 参照画像作成方法

カメラ座標系と物体座標系の位置関係を知る為に、メジャーでレンズからの距離を計測しておき、カメラで物体を撮影する。この時、出来るだけ物体がレンズ中心付近にある方が距離を計測するのが容易になる。ここで計測した値を参照画像設定ファイルへ記述する。

計測した距離パラメータは、コンポーネント実行後に結果と照らし合わせながら微調整すると良い。

2.4.2 参照画像設定記述方法

参照画像の ID、物体領域の指定、カメラ座標系と物体座標系の対応関係を参照画像設定ファイルへ記述することで、参照画像を登録することが出来る。

いかに記述コマンドを示す。

[物体 ID]

パラメータ 1=[ファイルパス]

パラメータ 2=[参照画像内の対象物の領域]

パラメータ 3=[外部パラメータ 1 行目]

パラメータ 4=[外部パラメータ 2 行目]

パラメータ 5=[外部パラメータ 3 行目]

| コマンド種類 | コマンド | 備考 |
|---------|------------------|--|
| パラメータ 1 | RefImgFile | 参照画像ファイルへのパス指定 |
| パラメータ 2 | RefImgRegion | 参照画像における物体領域の指定, パラメータは以下のように記述 始点 x 座標 : 始点 y 座標 : x 座標のサイズ : y 座標のサイズ |
| パラメータ 3 | CameraExtrinsic0 | カメラ座標系から物体座標系への同次座標変換行列の一行目 |
| パラメータ 4 | CameraExtrinsic1 | カメラ座標系から物体座標系への同次座標変換行列の二行目 |
| パラメータ 5 | CameraExtrinsic2 | カメラ座標系から物体座標系への同次座標変換行列の三行目 |

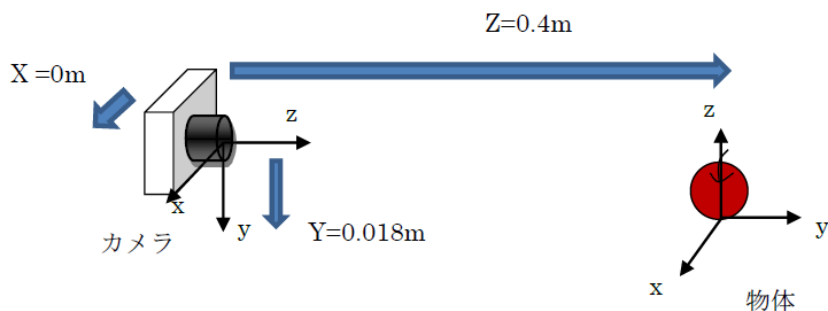
以下に記述例を示す

下図のように物体の位置・姿勢を表現するカメラ座標系と物体座標系を設定した場合の外部パラメータは以下のようになる.

X : カメラ座標原点からの物体座標原点の **x** 座標の距離・・・レンズの中心に来たほうがよいので, 0 が望ましい.

Y : カメラ座標原点からの物体座標原点の **y** 座標の距離・・・カメラと物体を同じ平面に置いた場合, レンズの中心高さ

Z : カメラ座標原点からの物体座標原点の **z** 座標の距離・・・カメラと物体との距離



上記の位置関係で $(X,Y,Z)=(0,0.018,0.40)$ の場合の参照画像の記入例は以下のようになる.

```
[aporo]
RefImgFile = image/aporo.png #image of reference pattern
RefImgReguib = 272:76:76:186 #image region of the reference pattern
CameraExtrinsic0 = 1.00 0.00 0.00 0.000 #pose of the reference
CameraExtrinsic1 = 0.00 0.00 -1.00 0.018 #[Rco Tco]
CameraExtrinsic2 = 0.00 1.00 0.00 0.400
```

なお, #以降はコメントとして無視される.

3 RTC の導入・動作確認

3.1 OpenCV3.1 の導入

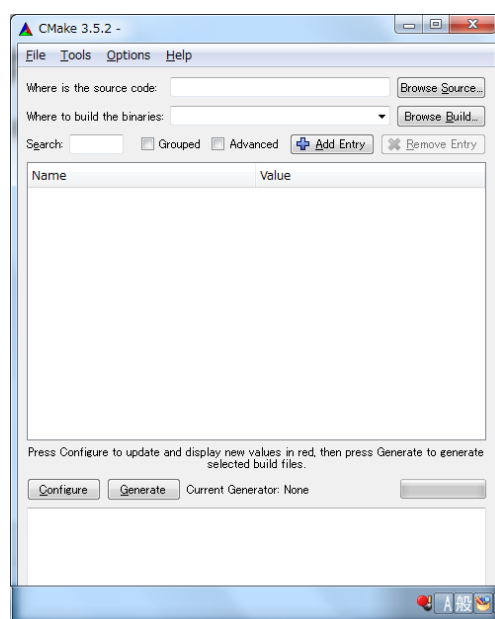
別紙の OpenCV3.1 導入マニュアルを参照

3.2 導入

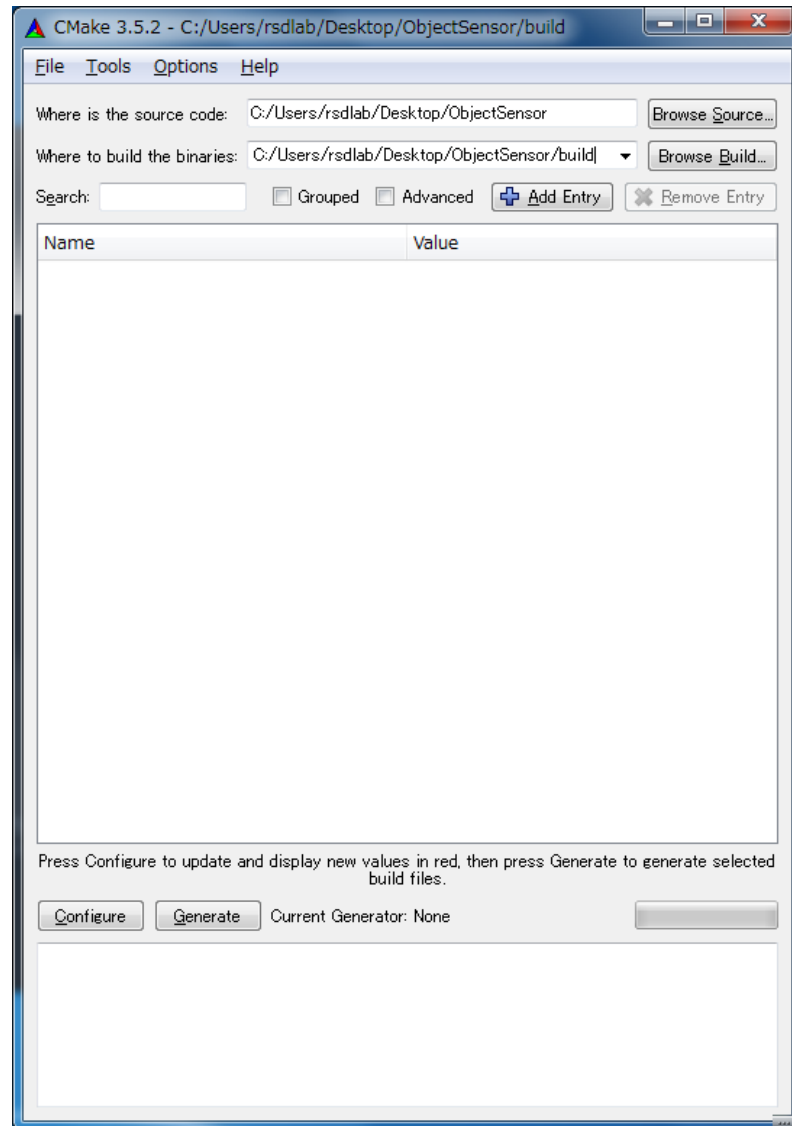
- ① Zip フォルダを展開．フォルダ内部は以下のようにになっている

| 名前 | 更新日時 | 種類 | サイズ |
|---------------------|------------------|-----------------|-------|
| cfg | 2016/12/07 5:03 | ファイル フォル... | |
| cmake | 2016/12/07 5:03 | ファイル フォル... | |
| doc | 2016/12/07 5:03 | ファイル フォル... | |
| idl | 2016/12/07 5:03 | ファイル フォル... | |
| images | 2016/12/07 5:04 | ファイル フォル... | |
| include | 2016/12/07 5:03 | ファイル フォル... | |
| src | 2016/12/07 5:03 | ファイル フォル... | |
| .project | 2016/09/19 1:46 | PROJECT ファイル | 1 KB |
| CMakeLists.txt | 2016/09/19 3:20 | TXT ファイル | 4 KB |
| COPYING | 2016/09/19 3:20 | ファイル | 35 KB |
| COPYING.LESSER | 2016/09/19 3:20 | LESSER ファイル | 8 KB |
| ObjectSensor.conf | 2016/10/31 13:14 | CONF ファイル | 5 KB |
| README.ObjectSensor | 2016/09/19 3:20 | OBJECTSENSOR... | 7 KB |
| rtc.conf | 2016/09/19 3:20 | CONF ファイル | 15 KB |
| RTC.xml | 2016/09/19 15:46 | XML ドキュメント | 8 KB |

- ② CMake(cmake-gui)を立ち上げる



- ③ Where is the source code に展開したファイルを指定。Where to build the binaries に展開したファイルに build を付け足した場所を指定する。

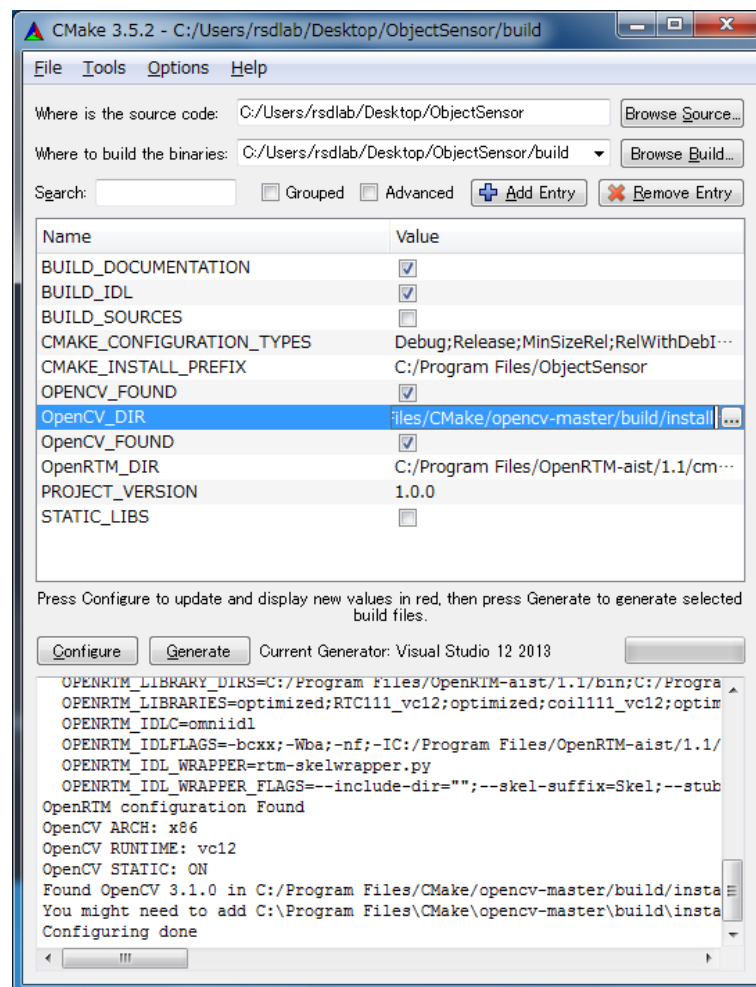


④ Configure を押す

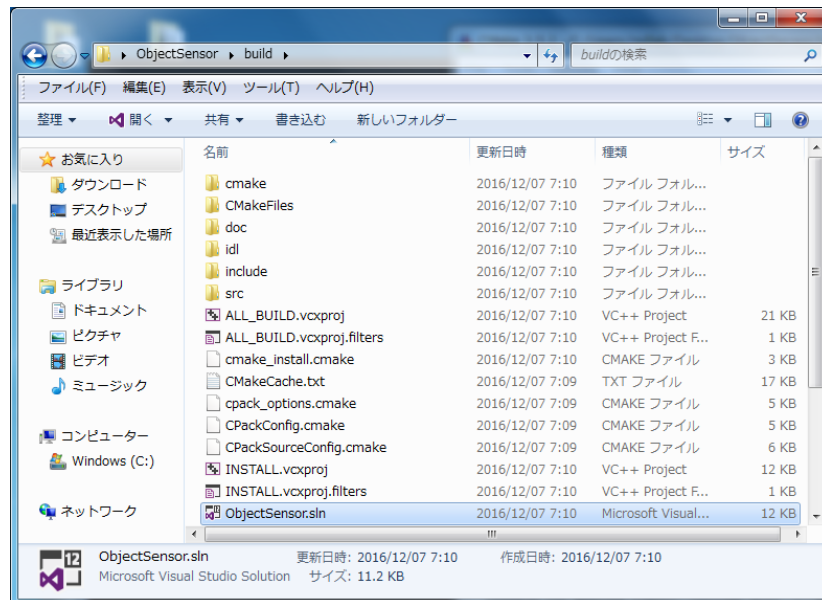
⑤ OpenCV_DIR を

[OpenCV3.1 のディレクトリ]/build/install

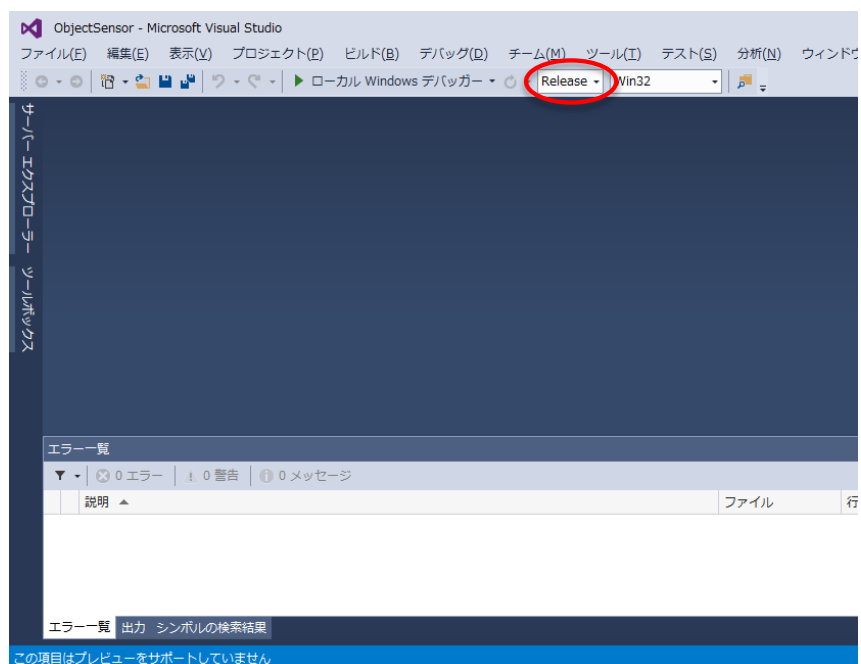
に指定する



- ⑥ Generate を押すと、build ファイルが生成される
- ⑦ build フォルダ内の ObjectSensor.sln を開く



- ⑧ Release モードにする



- ⑨ ビルドする。この時、スタートアッププロジェクトを設定していないとエラーが出るが、2回目のビルドは通る。
- ⑩ ObjectSensor 内の「cfg」と「image」ファイルを build/src/Release 内にコピーする。
(上側がコピー前、下側がコピー後)

