物体位置姿勢推定コンポーネント
マニュアル

国立大学法人　大阪大学

Rev. 1.4

目次

1　コンポーネントの概要 3
2　開発環境 4
3　コンポーネントのインタフェース仕様 5
3.1　InPort 5
3.2　OutPort 5
3.3　Service Provider 5
3.4　Configuration 6
4　Service Provider のインタフェース情報 7
4.1　ModelInfoService 7
5　座標系 8
6　参照画像 9
6.1　参照画像作成方法 9
6.2　参照画像設定記述方法 10
7　実行環境の設定 12
7.1　実行環境のインストール 12
8　コンポーネントの動作確認のための事前準備 14
9　改訂履歴 17
10　免責事項 17

1　コンポーネントの概要

本コンポーネントは予め取得した画像をモデルとして，アピ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．SIFT 特徴を用いることで，日常環境における照明条件の変化や物体の見え方におけるスケール変化，回転変化に強固な物体検出を実現する．

本コンポーネントでは，計算コストの高い SIFT 演算部に GPU による処理を適用することで, 処理の高速化に努めている. そのため, 本コンポーネントを利用するためには NVIDIA 社の CUDA に対応した GPU を利用している必要がある．

【注意事項】

なお，本コンポーネントは知能化プロジェクト共通カメラインタフェースに対応したコンポーネントであるため，画像入力のコンポーネントは共通カメラインタフェースに対応しているものである必要がある．

2　開発環境

本コンポーネントは以下の環境で開発・動作検証を行っている.

OS	Windows7
Compiler	Microsoft Visual C++ 2008
OpenRTM	1.0.0
CUDA SDK	4.0
CUDA toolkit	4.0
OpenCV	2.3.1

3　コンポーネントのインタフェース仕様

3.1　InPort

ポート名	データ型	備考
SingleImage	TimedCameraImage	カメラでキャプチャした RGB 画像を入力する.

3.2　OutPort

ポート名	データ型	備考
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.

3.3　Service Provider

ポート名	インタフェース型	インスタンス名	備考
Model	AcceptModelService	ModelAcceptor	物体モデル ID を指定する

3.4 Configuration

名前	データ型	備考
CfgName	string	参照画像設定ファイル名及びパスを指定する
SIFT_Ethreshold	double	カメラ画像に対する SIFT 特徴量の設定項目
SIFT_Sigma	double	カメラ画像に対する SIFT 特徴量の設定項目
SIFT_nLevels	Int	カメラ画像に対する SIFT 特徴量の設定項目
SIFT_nOctaves	Int	カメラ画像に対する SIFT 特徴量の設定項目
Display	string	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	物体名を記述することで，仕様する参照画像を指定する．

4　Service Provider のインタフェース情報

4.1　ModelInfoService

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，仕様参照モデルを変更することができる．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できるようになる．

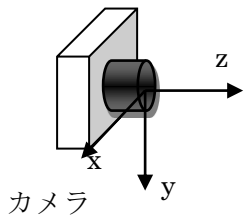
本インタフェースが定義されている IDL ファイルは以下のとおりである．

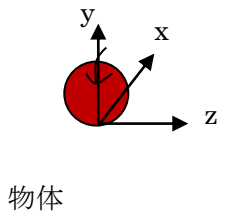
```
// interface for model provider
interface AcceptModelService{
        typedef sequence<octet> OctetSeq;
        // set model
        //@param model: a model of something
        void setModel(in OctetSeq model)
                raises(ServicePort::ServiceException);
};
```

ObjectInfoService.idl

5　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なっている.





6　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．

6.1　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の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．

6.2　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できる.
以下に記述コマンドを示す.

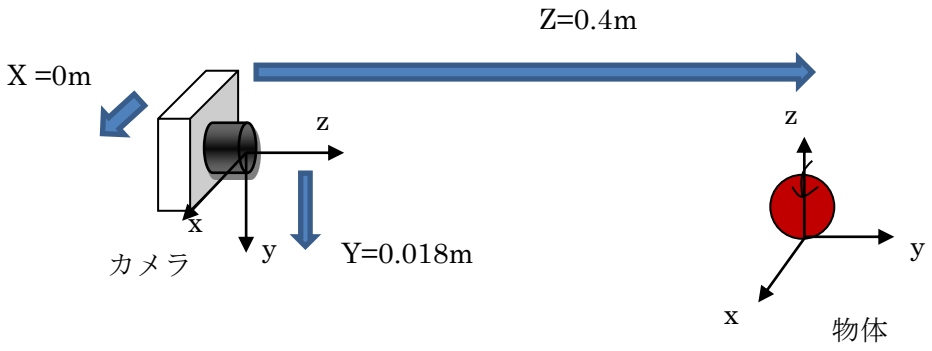
[物体 ID]
パラメータ 1 =[ファイルパス]
パラメータ 2=[参照画像内の対象物の領域]
パラメータ 3=[外部パラメータ　1 行目]
パラメータ 4=[外部パラメータ　2 行目]
パラメータ 5=[外部パラメータ　3 行目]

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座変換行列の一行目
パラメータ 4	CameraExtrinsic1	カメラ座標系から物体座標系への同座変換行列の二行目
パラメータ 5	CameraExtrinsic2	カメラ座標系から物体座標系への同座変換行列の三行目

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，レンズの中心の高さ
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      = images/aporo.png #image of reference pattern
RefImgRegion = 272:76:76:186        # image region of the reference pattern
CameraExtrinsic0 = 1.0    0.00    0.00    0.000    # pose of the reference pattern
CameraExtrinsic1 = 0.0    0.00    -1.00    0.018 #[Rco Tco]
CameraExtrinsic2 = 0.0    1.00    0.00    0.40
```

なお，＃以降はコメントとして無視される.

7　実行環境の設定

7.1　実行環境のインストール

- OpenCV ライブラリ

  別紙の OpenCV インストールマニュアルを参照

- CUDA ライブラリ

  (配布元：http://developer.nvidia.com/cuda-toolkit-40)

[インストール手順]

① 上記サイトから以下のファイルをダウンロードし，インストール

32bit，64bit はインストールする PC に対応したバージョンをインストール

なお，（ア）については，デスクトップ PC，ノート PC いずれを利用するかによって変わるので，利用環境に合わせたドライバを選択する必要があるので，注意が必要.

(ア) Developer Drivers for WinVista and Win7

もしくは

Notebook Developer Drivers for WinVista and Win7

(イ) CUDA Toolkit

(ウ) GPU Computing SDK code samples

② 環境変数 Path に"C:¥ProgramData¥NVIDIA Corporation¥NVIDIA GPU Computing SDK 4.0¥C¥bin¥win32¥Release"を追加

③ C:¥ProgramData¥NVIDIA Corporation¥NVIDIA GPU Computing SDK 4.0¥C¥common

にある cutil_vs2008.sln を開いて，Debug/Release 共にビルドし cutil32.lib cutil32D.lib 等を生成.

④ Visual Studio のパスの設定をする.

ツール→オプションからプロパティおよびソリューション→VC++ ディレクトリを開き，以下を設定.

・インクルードファイルに追加

C:¥ProgramData¥NVIDIA Corporation¥NVIDIA GPU Computing SDK 4.0¥C¥common¥inc

・ライブラリファイルに追加

C:¥ProgramData¥NVIDIA Corporation¥NVIDIA GPU Computing SDK 4.0¥C¥common¥lib¥Win32

C:¥Program Files¥NVIDIA GPU Computing Toolkit¥CUDA¥v4.0¥lib¥Win32

- CUDA VS Wizard のインストール
  http://sourceforge.net/projects/cudavswizard/files/cudavswizard/CUDA_VS_Wizard_2.0%20Release/
  こちらのサイトの CUDA_VS_Wizard_W32.2.0.zip をダウンロードしてインストール．Visual Studio で CUDA のファイルをコンパイルするために必要となる．

- Microsoft Visual C++ 2008 再頒布可能パッケージ (x86)（配布元：http://www.microsoft.com/downloads/details.aspx?familyid=9b2da534-3e03-4391-8a4d-074b9f2bc1bf&displaylang=ja）

[インストール手順]

上記サイトからファイルをダウンロードし，インストール

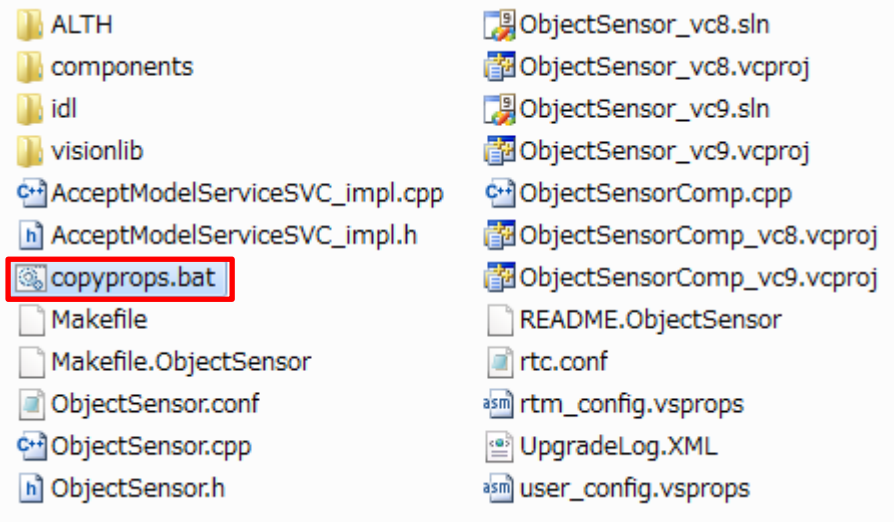
- Microsoft Visual C++ 2008 SP1 再頒布可能パッケージ (x86)（配布元：http://www.microsoft.com/downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=ja）

[インストール手順]

上記サイトからファイルをダウンロードし，インストール

8　コンポーネントの動作確認のための事前準備

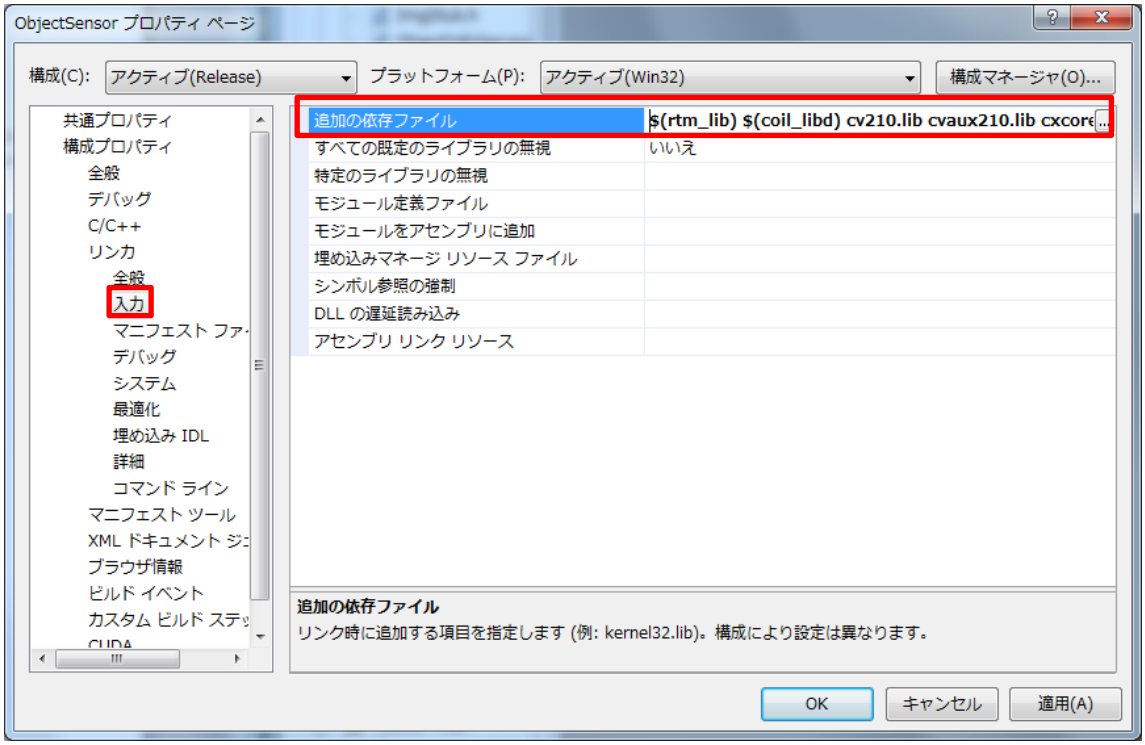
① Zip フォルダを展開．フォルダ内部の src 内部以下のようになっている．



② copyprops.bat を実行

③ ObjectSensor_vc9.sln を開き，[Release]に設定．

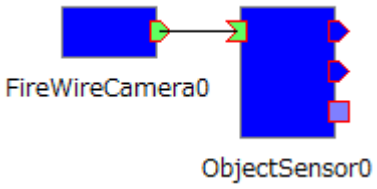
④ 違うバージョンの OpenCV（2.3.1 以外）をインストールしている方は以下を設定
ObjectSensor のプロジェクトと ObjectSensorComp のプロジェクトに対して OpenCV に対するリンカを入力する．プロジェクトのプロパティを開き，構成プロパティ→リンカ→入力の追加の依存ファイルにインストールした OpenCV に対応するリンカを入力（以下は OpenCV2.1 の場合）



⑤　ソリューションのビルドを実行

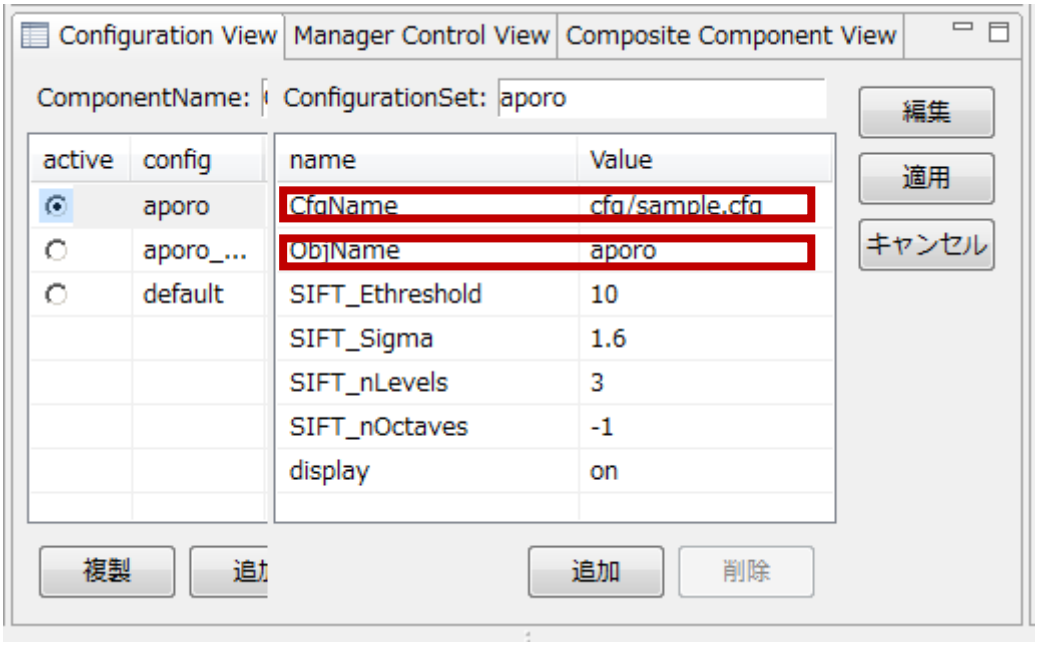
⑥　src¥components フォルダ内の ObjectSensorComp.exe を実行

⑦　カメラのコンポーネントと接続



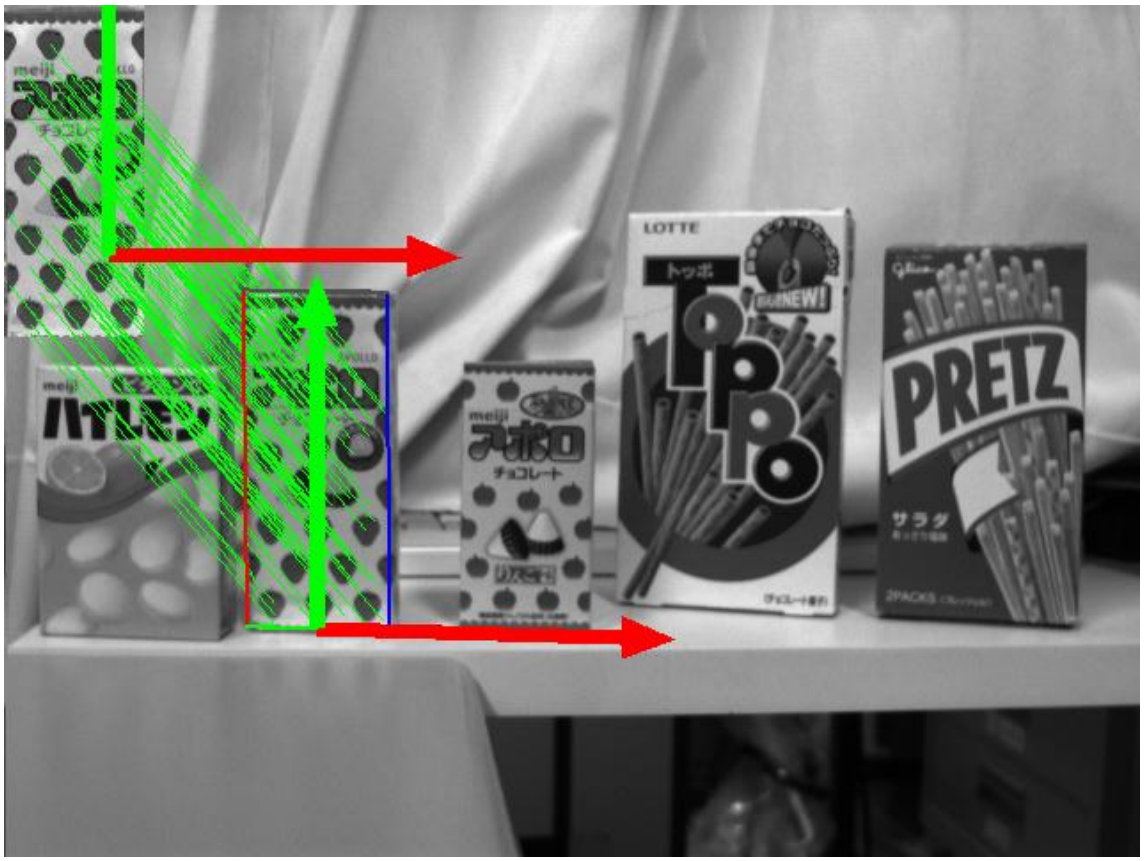
⑧　コンフィグレーションの設定

CfgName パラメータと ObjName パラメータが適切かどうか確認．外部から物体 ID を指定する場合は，ObjName パラメータが None になっていることを確認．



⑨　Activate する

物体が検出されると以下のような実行結果が表示される．(注：display パラメータが on の場合のみ)


meiji
meiji
meiji
チョコレート
LOTTE
トッポ
NEW!
PRETZ
サラダ
2PACKS

9　改訂履歴

Rev. 1.0（2010/01/15）:　初版
Rev. 1.1（2010/08/25）:　依存パッケージに関する情報を追記
Rev. 1.2（2011/07/29）:　ポートの仕様の変更，ステレオの機能を削除
Rev. 1.3（2011/12/06）:　CUDA，OpenCV ライブラリのバージョンの変更.
Rev. 1.4（2011/12/28）:　CUDA のインストール関係の記述を変更.

10 免責事項

本コンポーネントの利用により発生した，直接的又は間接的ないかなる損害についても責任を負わないものとします.