

# 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. 1.1

# 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	Service Consumer .....	6
3.5	Configuration .....	7
4	Service Provider のインタフェース情報 .....	8
4.1	ImageInfoService .....	8
4.2	CameraInfoService .....	8
4.3	ModelInfoService .....	10
5	座標系 .....	11
6	参照画像 .....	12
6.1	参照画像作成方法 .....	12
6.2	参照画像設定記述方法 .....	13
7	カメラパラメータ .....	15
7.1	カメラパラメータ記述方法 .....	15
8	実行環境の設定 .....	17
8.1	実行環境のインストール .....	17
9	コンポーネントの動作確認 .....	18
10	改訂履歴 .....	20
11	免責事項 .....	20

## 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．

## 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	Windows Vista, WindowsXP
Compiler	Microsoft Visual C++ 2008
OpenRTM	0.4.2
OpenCV	1.1pre
依存パッケージ	Microsoft Visual C++ 2005 再頒布可能パッケージ (x86)

※OpenCV のバージョンは 1.1 のみで動作する.

### 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
ImageData	TimedOctetSeq	単眼カメラまたはステレオカメラの左カメラでキャプチャした RGB 画像を入力する.
ImageData2	TimedOctetSeq	ステレオカメラの右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	TimedOctetSeq	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
ResImgInfo	ImageInfoService	ResImgInfo	検出及び位置姿勢推定結果を描画した RGB 画像のフォーマットを提供する

### 3.4 Service Consumer

ポート名	インタフェース型	インスタンス名	備考
CamInfo	CameraInfoService	CameraInfoService	単眼カメラまたはステレオカメラの左カメラのカメラパラメータを取得する
CamInfo	ImageInfoService	ImageInfoService	単眼カメラまたはステレオカメラの左カメラでキャプチャする画像フォーマットを取得する
CamInfo2	CameraInfoService	CameraInfoService2	ステレオカメラの右カメラのカメラパラメータを取得する
CamInfo2	ImageInfoService	ImageInfoService2	ステレオカメラの右カメラでキャプチャする画像フォーマットを取得する

### 3.5 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
ImgResHeight	int	結果画像の縦サイズを指定する. 現状では, 入力画像と同じサイズを指定.
ImgResWidth	int	結果画像の横サイズを指定する. 現状では, 入力画像と同じサイズを指定.
LeftSIFT_Ethreshold	double	単眼カメラまたはステレオカメラの左カメラ画像に対する SIFT 特徴量の設定項目
LeftSIFT_Sigma	double	単眼カメラまたはステレオカメラの左カメラ画像に対する SIFT 特徴量の設定項目
LeftSIFT_nLevels	int	単眼カメラまたはステレオカメラの左カメラ画像に対する SIFT 特徴量の設定項目
LeftSIFT_nOctaves	int	単眼カメラまたはステレオカメラの左カメラ画像に対する SIFT 特徴量の設定項目
RightSIFT_Ethreshold	double	ステレオカメラの右カメラ画像に対する SIFT 特徴の設定項目
RightSIFT_Sigma	double	ステレオカメラの右カメラ画像に対する SIFT 特徴の設定項目
RightSIFT_nLevels	int	ステレオカメラの右カメラ画像に対する SIFT 特徴の設定項目
RightSIFT_nOctaves	int	ステレオカメラの右カメラ画像に対する SIFT 特徴の設定項目
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 ImageInfoService

ImageInfoService インタフェースは画像のフォーマットを提供する.

```
// pixel format
typedef enum { PIXEL_UNKNOWN=0, // default value (not meant for use)
    PIXEL_GRAY, // grayscale (1 channel /pixel) (the smallest)
    PIXEL_GRAYA, // grayscale + alpha (2 channels/pixel)
    PIXEL_RGB, // RGB (3 channels/pixel)
    PIXEL_RGBA, // RGB + alpha(4 channels/pixel)
    PIXEL_BGR, // BGR (3 channels/pixel)
    PIXEL_YUV, // YUV (3 channels/pixel)
    PIXEL_YUYV, // Y and alternating UV (2 channels/pixel)
    PIXEL_INT, // integer values (1 channels/pixel)
    PIXEL_FLOAT, // float values (1 channels/pixel)
    PIXEL_VOIDP, // void pointer (1 channels/pixel)
    PIXEL_F3V // float[3] (3 channels/pixel)
} pixel_t;
```

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.

```
// image information
interface ImageInfoService {
    // get image size and format
    short getImageWidth();
    short getImageHeight();
    short getPixelFormat();
};
```

ImageInfoService.idl

### 4.2 CameraInfoService

CameraInfoService では, 接続されているカメラに関する情報を他のコンポーネントに伝えるための機能を提供する. 本インタフェースを規定している IDL ファイルは以下のとおりである.



```

// camera parameter information
interface CameraInfoService{
    // definition of intrinsic parameter's size
    const long INTRINSIC_PARAMETER_SIZE = 6;
    // definition of distortion parameter's size
    const long DISTORTION_PARAMETER_SIZE = 5;
    // intrinsic parameter data
    typedef double IntrinsicParameter [INTRINSIC_PARAMETER_SIZE];
    // distortion parameter data
    typedef double DistortionParameter [DISTORTION_PARAMETER_SIZE];
    // get intrinsic parameter
    boolean getIntrinsicParameter(out IntrinsicParameter data);
    // get distortion parameter
    boolean getDistortionParameter(out DistortionParameter data);
};

// stereo camera parameter information
interface StereoCameraInfoService{
    // definition of extrinsic parameter's size
    const long EXTRINSIC_PARAMETER_SIZE = 12;
    // extrinsic parameter
    typedef double ExtrinsicParameter [EXTRINSIC_PARAMETER_SIZE];
    // get extrinsic parameter between cameras
    boolean getExtrinsicParameterBC(out ExtrinsicParameter data);
};

```

CameraInfoService.idl

### 4.3 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。

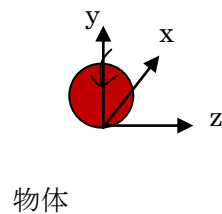
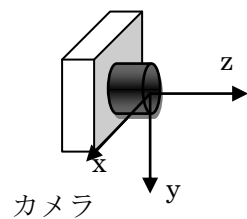
本インタフェースが定義されている IDL ファイルは以下のとおりである。

```
// 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);
};
```

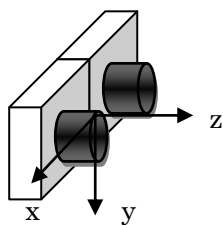
ObjectInfoService.idl

## 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になっている。



ステレオカメラを利用する場合は，並行ステレオと仮定している．ステレオカメラにおけるカメラ座標系は仮想的に二つのカメラの中心と仮定する．



ステレオカメラ

## 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．また，ステレオカメラの場合，参照画像は左右画像を一枚に統合させておく．

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(左カメラ)]

パラメータ 3=[参照画像内の対象物の領域(右カメラ)]

パラメータ 4=[外部パラメータ 1 行目]

パラメータ 5=[外部パラメータ 2 行目]

パラメータ 6=[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(左カメラ、単眼カメラ画像)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:終点 x 座標:終点 y 座標
パラメータ 3	RefImgRegion2	参照画像(右カメラ画像)における物体領域の指定 指定方法は RefImgRegion と同じ
パラメータ 4	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 5	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 6	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

```
[toppo]
RefImgFile    = images/ref_toppo.png #image of reference pattern
RefImgRegion = 330:118:81:144 # image region of the reference pattern
RefImgRegion2 = 663:121:81:144
CameraExtrinsic0 = 1.0  0.00  0.00  0.000  # pose of the reference pattern
CameraExtrinsic1 = 0.0  0.00  -1.00 -0.0175 #[Rco Tco]
CameraExtrinsic2 = 0.0  1.00  0.00  0.55
```

なお，#以降はコメントとして無視される．

## 7 カメラパラメータ

通常は CameraInfoService により, 他コンポーネントからカメラパラメータを取得するが, CameraInfoService が他コンポーネントと接続されていない場合は, パラメータ設定ファイルからカメラパラメータを取得する. カメラパラメータ設定は参照画像設定と同じファイルへ記述する.

### 7.1 カメラパラメータ記述方法

カメラパラメータ設定ファイルに, 以下のコマンドを記述する.

[カメラ設定]  
パラメータ名 = パラメータ 1, パラメータ 2, ...

設定ファイルでは, 次のコマンドを記述することができる.

コマンド種類	コマンド	備考
カメラ設定	StereoCamera	ステレオカメラにおける設定
パラメータ名	CameraStereoBaseline	ステレオカメラのベースライン
パラメータ名	CameraDefaultSIFTPattern	デフォルトで使用する物体 ID コンフィギュレーションと不整合な場合は無視される
カメラ設定	StereoCameraLeft	ステレオカメラの左カメラにおける設定
カメラ設定	StereoCameraRight	ステレオカメラの右カメラにおける設定
パラメータ名	CameraResolution	カメラでキャプチャする画像の解像度 記述例: 640 x 480
パラメータ名	CameraIntrinsic0	カメラ内部パラメータの一行目を設定
パラメータ名	CameraIntrinsic1	カメラ内部パラメータの二行目を設定
パラメータ名	CameraDistortion	カメラの歪みパラメータを設定

尚, #を付けるとコメントアウトできる.

以下に記述例を示す

```
[StereoCamera]
```

```
CameraStereoBaseline = 0.12
```

```
CameraDefaultSIFTPattern = pretz
```

```
[StereoCameraLeft]
```

```
CameraParameter.Intrinsic1: 594.128,      0, 319.988
```

```
CameraParameter.Intrinsic2:      0, 600.2, 240.004
```

```
CameraParameter.Distortion: 9.14364e-07 # invertible distortion model
```

```
[StereoCameraRight]
```

```
RightCameraParameter.Intrinsic1: 590.434,      0, 320.028
```

```
RightCameraParameter.Intrinsic2:      0, 599.779, 240.011
```

```
RightCameraParameter.Distortion: 8.73809e-07 # invertible distortion model
```



## 8 実行環境の設定

### 8.1 実行環境のインストール

- OpenCV ライブラリ (配布元 : <http://sourceforge.net/projects/opencvlibrary/>)

[インストール手順]

- ① 上記サイトなどで OpenCV\_1.1pre1a.exe をダウンロードし, 実行
- ② 環境変数 Path に”C:¥Program Files¥OpenCV¥bin”を追加

- 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=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]

- ① 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

※OpenCV1.1 の動作のために以下のパッケージもダウンロードする

- Microsoft Visual C++ 2005 再頒布可能パッケージ (x86) (配布元 : <http://www.microsoft.com/downloads/details.aspx?familyid=32BC1BEE-A3F9-4C13-9C99-220B62A191EE&displaylang=ja>)

[インストール手順]

- ① 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++ 2005 SP1 再頒布可能パッケージ (x86) (配布元 : <http://www.microsoft.com/downloads/details.aspx?FamilyID=200b2fd9-ae1a-4a14-984d-389c36f85647&displaylang=ja>)

[インストール手順]

- ① 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

## 9 コンポーネントの動作確認

### 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

#### (ア) ObjectSensorComp.exe

物体位置姿勢推定コンポーネントの実行ファイル

#### (イ) cfg フォルダ

参照画像，カメラパラメータを記述したファイルを格納するフォルダ．中にこれらのパラメータを記述した `sample.cfg` ファイルがある．

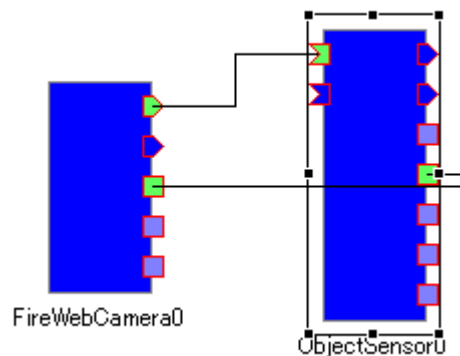
#### (ウ) images フォルダ

参照画像を格納するフォルダ．この中にはサンプルとして，`ref_toppo.png` が入っている．尚，サンプル画像は Point Gray 社製 Bumblebee2 により撮影した．



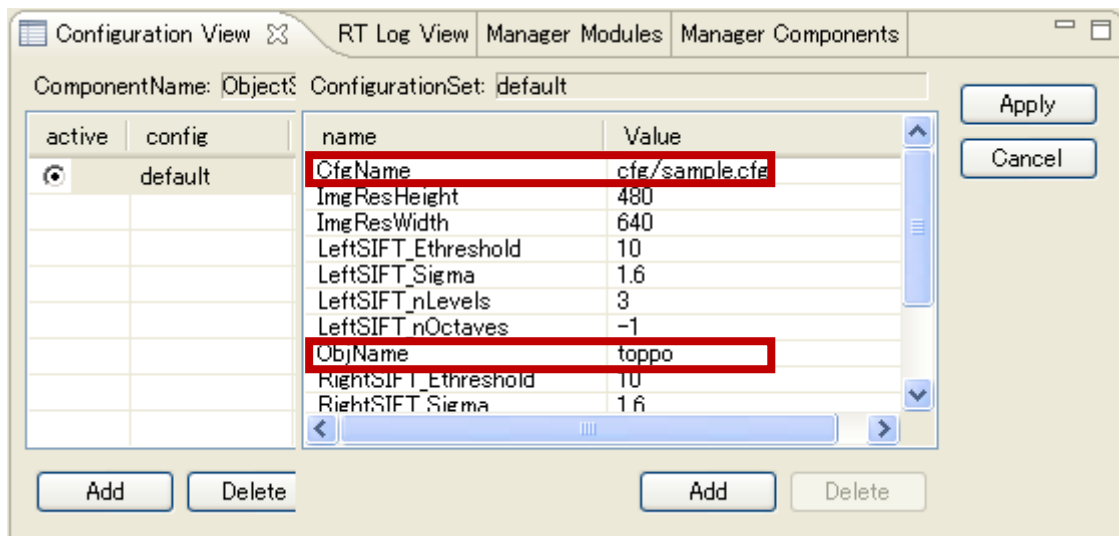
### ② ObjectSensorComp.exe を実行

### ③ 画像出力コンポーネントと接続



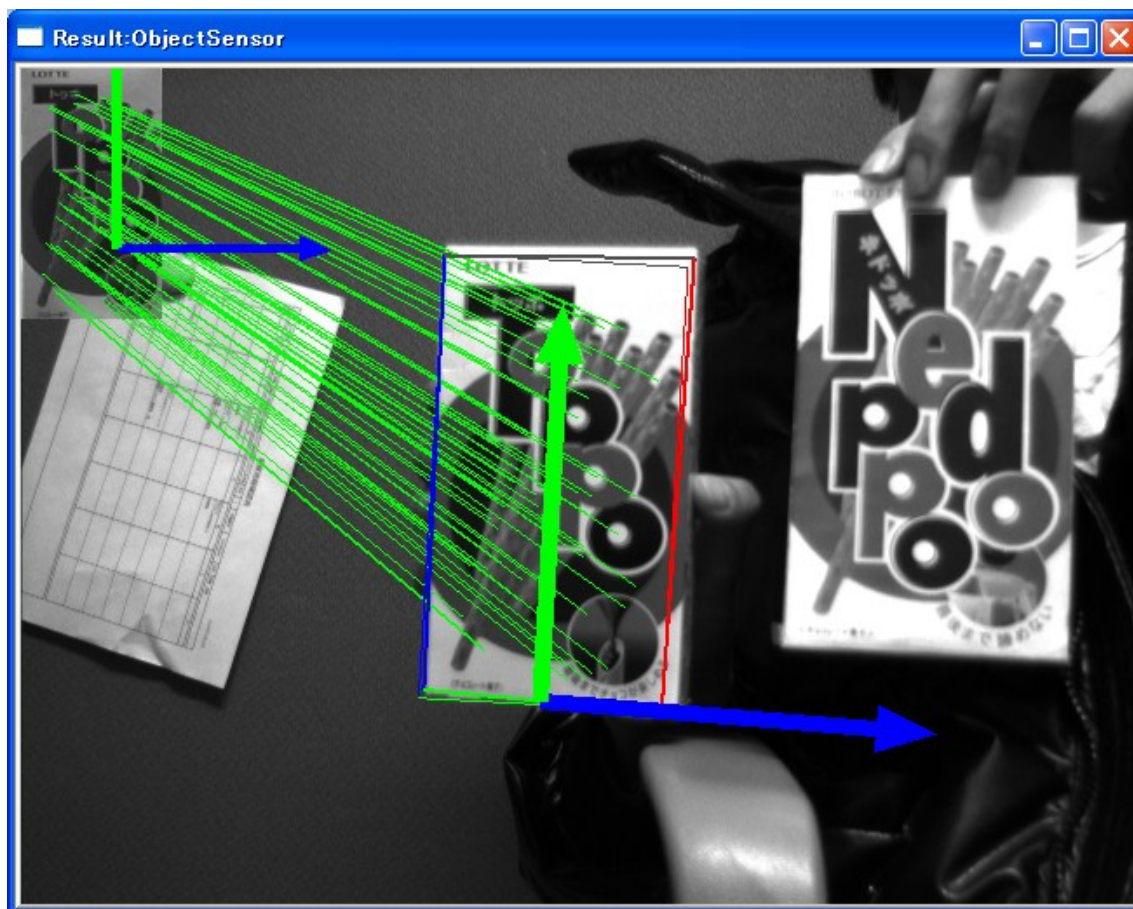
### ④ コンフィグレーションの設定

`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．



⑤ 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)



## 10 改訂履歴

Rev. 1.0 (2010/03/10) : 初版

Rev. 1.1 (2010/08/25) : 依存パッケージに関する記述を追記

## 11 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．