

機能仕様書

オープンソース版作業対象認識モジュール群

OpenVGR

Ver. 0.9.0

2011 年 12 月 21 日

独立行政法人 産業技術総合研究所

## 改版履歷

[illegible]

## 目次

1. はじめに .....	1
1. 1. 本書の適用範囲 .....	1
1. 2. 関連文書 .....	1
1. 3. 本書を読むにあたって .....	1
2. 機能仕様 .....	2
2. 1. 機能概要 .....	2
2. 2. モジュール群構成 .....	2
2. 3. ターゲットハードウェア .....	2
3. RTC 仕様 .....	3
3. 1. Measure3DComp (3次元距離計測 RTC) .....	3
3. 1. 1. 機能概要 .....	3
3. 1. 2. 動作環境 .....	3
3. 1. 3. ポート情報 .....	3
3. 1. 4. コンフィグレーション .....	4
3. 1. 5. 入出力データフォーマット .....	4
3. 1. 6. サービスポート I/F 仕様 .....	7
3. 1. 7. 設定ファイル .....	7
3. 2. MultiCameraComp (ステレオ画像取得 RTC) .....	9
3. 2. 1. 機能概要 .....	9
3. 2. 2. 動作環境 .....	9
3. 2. 3. ポート情報 .....	9
3. 2. 4. コンフィグレーション .....	10
3. 2. 5. 入出力データフォーマット .....	10
3. 2. 6. サービスポート I/F 仕様 .....	10
3. 2. 7. 設定ファイル .....	10
3. 3. MultiDispComp (画像表示 RTC) .....	11
3. 3. 1. 機能概要 .....	11
3. 3. 2. 動作環境 .....	11
3. 3. 3. ポート情報 .....	11
3. 3. 4. コンフィグレーション .....	12
3. 3. 5. 入出力データフォーマット .....	12
3. 4. RecognitionComp (作業対象認識 RTC) .....	13
3. 4. 1. 機能概要 .....	13
3. 4. 2. 動作環境 .....	13
3. 4. 3. ポート情報 .....	13
3. 4. 4. コンフィグレーション .....	15
3. 4. 5. 入出力データフォーマット .....	15

3. 4. 6. サービスポート I/F 仕様 .....	18
3. 4. 7. 設定ファイル .....	18
3. 5. RecognitionResultViewerComp (認識結果表示 RTC) .....	22
3. 5. 1. 機能概要 .....	22
3. 5. 2. 動作環境 .....	22
3. 5. 3. ポート情報 .....	22
3. 5. 4. コンフィグレーション .....	23
3. 5. 5. 入出力データフォーマット .....	23
3. 5. 6. サービスポート I/F 仕様 .....	23
3. 5. 7. 設定ファイル .....	23
4. ツール仕様 .....	24
4. 1. multicalib (カメラキャリブレーションツール) .....	24
4. 1. 1. 機能概要 .....	24
4. 1. 2. 動作環境 .....	24
4. 1. 3. 使用方法 .....	24
4. 1. 4. データファイルのフォーマット .....	25
4. 2. VGRModeler (モデル作成ツール) .....	26
4. 2. 1. 機能概要 .....	26
4. 2. 2. 動作環境 .....	26
4. 2. 3. コマンドラインオプション .....	26
4. 2. 4. 設定ファイル .....	26
5. 特記事項 .....	29
5. 1. カメラの座標系 .....	29

# 1. はじめに

## 1. 1. 本書の適用範囲

本書はロボットビジョンを活用する技術者を対象に、カメラと画像処理によって周囲の環境から特定の物体を見つけるためのソフトウェアについて解説する。ソフトウェアは作業対象認識モジュール群であり、その RT コンポーネントとツールのインターフェースを各項目に記述する。なお、本書では、通信機能を持つプログラムを RT コンポーネント、単独で機能するプログラムをツールと呼ぶ。その二つを総称する場合にモジュールと呼ぶ。また、RT コンポーネントの表記として RTC を用いる。

## 1. 2. 関連文書

表 1-1 関連文書

No.	文書名	備考
1	はじめにお読みください	
2	操作手順書	

## 1. 3. 本書を読むにあたって

本書で取り扱うソフトウェアを「作業対象認識モジュール群」と総称する。作業対象認識モジュール群のあらましは「はじめにお読みください」を参照すること。

ここでは、読者として RTC を作成するプログラマを想定する。本書は、RTC に基づく画像処理ソフトウェアについての技術文書である。読むにあたり、読者には RT ミドルウェアと画像処理に関する初歩的な知識が必要とされる。

RT ミドルウェア、RTC については下記を参照のこと。

OpenRTM-aist Official Website:

<http://www.openrtm.org/>

## 2. 機能仕様

### 2. 1. 機能概要

このモジュール群の主要な機能はモデルに基づく物体認識である。画像中からモデルによく合致するオブジェクトを探し出して、その位置姿勢を出力する。モデルの照合は 3 次元空間で行われる。環境の情報をステレオカメラから取得し、立体視の原理に基づいて 3 次元空間の情報を復元して、モデルの特徴と照合する。したがって認識結果もまた 3 次元の位置姿勢となる。

### 2. 2. モジュール群構成

このモジュール群は、以下の 7 個のモジュールに機能が分担されている。

- 撮影（ステレオ画像取得 RTC）
- 画像の表示（画像表示 RTC）
- 計測（3 次元距離計測 RTC）
- 認識（作業対象認識 RTC）
- 結果の表示（認識結果表示 RTC）
- 較正（カメラキャリブレーションツール）
- モデルの作成（モデル作成ツール）

このモジュール群を利用する RTC には、モデルを指定して、認識結果を受け取る機能が必要とされる。このとき認識させるモデルとして、あらかじめモデル作成ツールによってモデルデータファイルを作成しておく必要がある。

### 2. 3. ターゲットハードウェア

このモジュール群は、ステレオ画像を撮影するために IEEE 1394b 仕様に準拠するデジタルカメラを 3 台必要とする。Point Grey Research 社製 Flea2 カメラによってソフトウェアの動作が確認されている。

## 3. RTC 仕様

### 3. 1. Measure3DComp (3 次元距離計測 RTC)

#### 3. 1. 1. 機能概要

3 次元距離計測 RTC は、ステレオ画像を入力し、ステレオ相関法により計算した 3 次元距離画像を出力する。

#### 3. 1. 2. 動作環境

この RTC は次の環境で動作する。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix (32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++)
依存パッケージ	OpenCV 2.0

#### 3. 1. 3. ポート情報



図 1 3 次元距離計測 RTC

##### A) データポート (InPort)

名称	型	データ長	説明
ImageIn	TimedMultiCameraImage	不定	ステレオ画像取得 RTC が出力するステレオ画像データ。

## B) データポート (OutPort)

名称	型	データ長	説明
Data3DOut	TimedStereo3D	不定	3次元距離計測データ。画像および3次元の点群データが含まれる。

## C) サービスポート (Provider)

サービス名	インターフェース名	説明
Reconstruct3D	Reconstruct3Dservice	外部の RTC から 3次元距離計測データの要求を受信する。入力ポートにデータが入力されたら、計測処理を実行する。

## D) サービスポート (Consumer)

サービス名	インターフェース名	説明
CameraCapture	CameraCaptureService	ステレオ画像取得 RTC に対して撮像開始トリガを送信する。

## 3. 1. 4. コンフィグレーション

なし。

## 3. 1. 5. 入出力データフォーマット

## A) ステレオ画像データ

入力ポートのステレオ画像データは次のデータフォーマットに従う。



Img.idl (抜粋):

```
module Img {
struct ImageData
{
    long width;
    long height;

    ColorFormat format;
    sequence<octet> raw_data;
};

/* camera image */
struct CameraIntrinsicParameter
{
    double matrix_element[5];
    sequence<double> distortion_coefficient;
};

struct CameraImage
{
    RTC::Time captured_time;
    ImageData image;
    CameraIntrinsicParameter intrinsic;
    Mat44 extrinsic;
};

struct MultiCameraImage
{
    sequence<CameraImage> image_seq;
    long camera_set_id;
};

struct TimedMultiCameraImage
{
    RTC::Time tm;
    MultiCameraImage data;
    long error_code;
};
}; /* module */
```

TimedMultiCameraImage 構造体は入れ子構造になっており、以下の情報が含まれる。

- TimedMultiCameraImage
  - tm: タイムスタンプ
  - data: ステレオ画像データ
    - image\_seq: 各カメラの画像データ
      - captured\_time: 画像の撮影時刻
      - image: 画像データ
        - width: 画像の幅 [pixel]
        - height: 画像の高さ [pixel]
        - format: 白黒(CF\_GRAY)またはカラー(CF\_RGB)
        - raw\_data: 画像の高さ [pixel]
      - intrinsic: 内部パラメータ
        - matrix\_element: 内部パラメータ行列の上三角成分
        - distortion\_coefficient: 歪み係数
      - extrinsic: 外部パラメータ(世界座標系からカメラ座標系への変換行列)
    - camera\_set\_id: カメラセットを識別するための任意の番号
    - error\_code: エラーコード (正常時は 0)

## B) 3次元距離計測データ

出力ポートの3次元距離計測データは次のデータフォーマットに従う。

```
struct TimedStereo3D
{
    RTC::Time tm;
    Stereo3D data;
    long error_code;
};
```

また、Stereo3D 構造体は次のデータフォーマットに従う。

```
struct Stereo3D
{
    TimedPointCloud obj;
    TimedMultiCameraImage img;
};

struct TimedPointCloud
{
    RTC::Time tm;
    PointCloud data;
    long error_code;
};

struct PointCloud
{
    long id;
    Img::Mat44 T;
    sequence<Img::Vec3> point;
    sequence<Img::Vec3> color;
};
```

### 3. 1. 6. サービスポート I/F 仕様

#### A) Reconstruct3DService

関数名	reconstruct			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	なし			
説明	3次元距離計測処理を開始する。			

#### B) CameraCaptureService

関数名	take_one_frame			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	なし			
説明	ステレオ画像取得 RTC に対し画像出力の要求を送信する。			

### 3. 1. 7. 設定ファイル

この RTC では、設定ファイルによって 3 次元距離計測に必要なパラメータを設定する。設定ファイルの名称は固定で、"measure3d\_config.d"である。このファイルは、実行ファイル Measure3DComp と同じディレクトリに置かれなくてはならない。最初にテンプレートのファイルが置かれているので、必要であれば値を編集することができる。

設定ファイルの各行には空白またはタブで区切られた二つ以上の項目が書かれ、最初の項目はパラメータ名のいずれか、二番目の項目は値（整数値）でなければならない。存在しないパラメータ名あるいは整数値以外の値が書かれた場合にはエラーになる。三番目以降にコメント等を書くことは構わない。その場合、三番目以降の項は単に無視される。

設定ファイルにおいては、すべてのパラメータが設定されなければならない。同じパラメータの設定を二度以上した場合には最後に書かれた値が有効になる。

以下に、各パラメータの簡単な説明を書く。なお、これらのパラメータはすべて OpenCV の関数 cvFindStereoCorrespondenceBM0に渡される引数の CvStereoBMState 構造体のメンバーである。詳細は、上記関数の説明にゆだねる。

名称	デフォルト値	説明
preFilterType	0	未使用
preFilterSize	33	平滑化ウィンドウサイズ、大きいほどスムーズになる
preFilterCap	33	平滑化のクリップサイズ
SADWindowSize	33	相関法のウィンドウサイズ
minDisparity	0	最小ディスパリティ値
numberOfDisparities	128	ディスパリティ範囲
textureThreshold	10	ディスパリティを計算するためのテクスチャの強度
uniquenessRatio	15	探索範囲内のすべての $d$ において $\text{sad}(d) \geq \text{sad}(d^*) \cdot (1 + uR/100)$ のときにディスパリティ $d^*$ を許可する。 つまり、この値が大きいと似た場所があったときにその値が不可になるということ。
speckleWindowSize	0	スペckルノイズのウィンドウサイズ
speckleRange	10	上記ウィンドウ内で許可するディスパリティの差
trySmallerWindows	1	1 がセットされると、遅くなるかわりに正確になる。

## 3. 2. MultiCameraComp（ステレオ画像取得 RTC）

### 3. 2. 1. 機能概要

複数の IEEE 1394b カメラでキャプチャした画像をまとめ、3 次元距離計測 RTC などの入力となるステレオ画像データを作成・出力する。

### 3. 2. 2. 動作環境

本 RTC の動作環境は以下の通りである。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix (32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++)
依存パッケージ	libdc1394-22, OpenCV 2.0

### 3. 2. 3. ポート情報

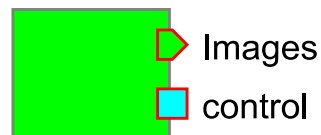


図 2 ステレオ画像取得 RTC

#### A) データポート（InPort）

なし。

#### B) データポート（OutPort）

名称	型	データ長	説明
images	TimedMultiCameraImage	1	ステレオ画像データ

#### C) サービスポート（Provider）

サービス名	インターフェース名	説明
control	CameraCaptureService	撮像開始トリガ入力

## D) サービスポート (Consumer)

なし。

### 3. 2. 4. コンフィグレーション

名称	型	デフォルト値	説明
camera_calib_file	string	camera_calib.yaml	カメラキャリブレーションデータのファイル名
camera_set_id	int	1	カメラセットを識別する任意の ID
camera_setting_file	string	ieee1394board.0	カメラの設定ファイル名

### 3. 2. 5. 入出力データフォーマット

出力データポートの TimedMultiCameraImage 構造体は、3.1.5(A)のものと同一である。

### 3. 2. 6. サービスポート I/F 仕様

## A) CameraCaptureService

関数名	take_one_frame			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	なし			
説明	画像取得の指示（トリガ）を与える。取得した画像はデータポート images より出力される。			

### 3. 2. 7. 設定ファイル

ステレオ画像取得 RTC は、初期設定状態でカレントディレクトリにある次の設定ファイルを参照する。

- ieee1394board.0: カメラ設定ファイル(詳細は操作手順書を参照のこと)
- camera\_calib.yaml: カメラキャリブレーションデータ

camera\_calib.yaml はカメラキャリブレーションツール(→4. 1. )を用いて生成することができる。

### 3. 3. MultiDispComp（画像表示 RTC）

#### 3. 3. 1. 機能概要

データポートより入力された画像データをディスプレイに表示する。

#### 3. 3. 2. 動作環境

本 RTC の動作環境は以下の通りである。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix(32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++)
依存パッケージ	OpenCV 2.0

#### 3. 3. 3. ポート情報



図 3 画像表示 RTC

##### A) データポート（InPort）

名称	型	データ長	説明
images	TimedMultiCameraImage	1	ステレオ画像データ

##### B) データポート（OutPort）

なし。

##### C) サービスポート（Provider）

なし。

##### D) サービスポート（Consumer）

なし。

### 3. 3. 4. コンフィグレーション

名称	型	デフォルト値	説明
image_dir	string	images	入力画像データ保存用ディレクトリ名
image_save_mode	int	0	保存モードの選択 (0: 保存しない、その他: 保存する)

### 3. 3. 5. 入出力データフォーマット

入力データポートの TimedMultiCameraImage 構造体は、3.1.5(A)のものと同一である。



## 3. 4. RecognitionComp（作業対象認識 RTC）

### 3. 4. 1. 機能概要

作業対象認識 RTC は、モデルによく合致するオブジェクトを画像から探し出す。モデルの照合は 3 次元であり、認識結果はオブジェクトの位置姿勢を表す。入力はステレオ画像データを含む距離計測データとモデルであり、出力は認識結果および距離計測データである。また認識結果表示 RTC を呼び出して、結果が視覚的に分かるように認識されたオブジェクトの投影画像を画面表示する。

### 3. 4. 2. 動作環境

この RTC は次の環境で動作する。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix(32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++)
依存パッケージ	OpenCV 2.0

### 3. 4. 3. ポート情報

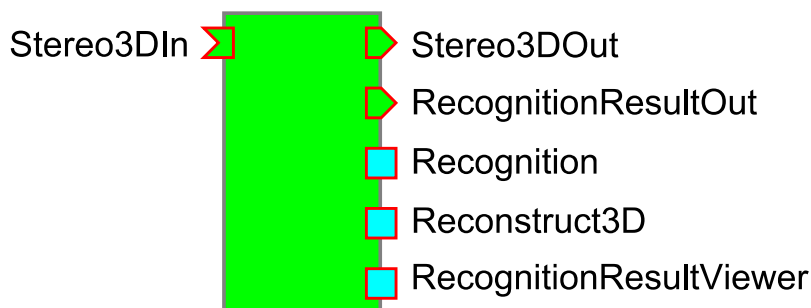


図 4 作業対象認識 RTC

## A) データポート (InPort)

名称	型	データ長	説明
Stereo3DIn	TimedStereo3D	不定	3次元距離計測 RTC が出力する 3次元距離計測データ。ステレオ画像や 3次元の点群データが含まれる。

## B) データポート (OutPort)

名称	型	データ長	説明
RecognitionResultOut	TimedRecognitionResult	不定	認識結果。位置姿勢、評価値、モデル ID などの情報が含まれる。複数の認識候補がある場合は、先頭から有力な順に並べられる。
Stereo3DOut	TimedStereo3D	不定	入力ポートで受け取ったものと同じ距離計測データ。

## C) サービスポート (Provider)

サービス名	インターフェース名	説明
Recognition	RecognitionService	外部の RTC からモデル ID で指定し、一連の画像処理を開始する。 モデル ID が指定されると、認識設定ファイルから認識設定を読み込み、距離計測データの要求を行う。

## D) サービスポート (Consumer)

サービス名	インターフェース名	説明
Reconstruct3D	Reconstruct3Dservice	3次元距離計測 RTC に対して距離計測データの送信を要求する。入力ポートにデータが入力されたら、認識処理を実行する。
RecognitionResultViewer	RecognitionResultViewerService	認識結果表示 RTC を呼び出し、認識結果を画面に表示する。

### 3. 4. 4. コンフィグレーション

名称	型	デフォルト値	説明
RecogModelListPath	string	“modelList.txt”	モデルリストファイル名 3.4.7 参照
RecogParameterFilePath	string	“recogParameter.txt”	認識パラメータファイル名 3.4.7 参照
DebugText	int	0	中間結果テキスト情報の ファイル出力の選択 (0:しない、その他:する)
DebugImage	int	0	中間結果画像情報のファ イル出力の選択 (0:しない、その他:する)
DebugDisplay	int	0	中間結果の画像情報表示 の選択 (0:しない、その他:する) 表示した画像はファイル 保存される

### 3. 4. 5. 入出力データフォーマット

#### A) 3次元距離計測データ

入力ポートおよび出力ポートの TimedStereo3D 構造体は、3.1.5(B)のものと同一である。

#### B) 認識結果

出力ポートの TimedRecognitionResult 構造体は、次のデータフォーマットに従う。

```
typedef RTC::TimedDoubleSeq TimedRecognitionResult;
```

TimedRecognitionResult の内容は、20 個の浮動小数点を一組のデータとする 0 組以上のデータ列である。一組のデータの内訳は、先頭から順に次の通りである。

0	カメラセット ID	撮影を行ったステレオカメラの番号
1	モデル ID	認識に用いたモデルの番号
2	認識候補番号	0 から始まる通し番号。有力な認識候補ほど若い番号。
3	座標系番号	座標系を表す番号。予約されたデータ項目。
4	評価値	確かさの評価値。大きいほど有力な認識候補。
5	エラーコード	エラーコード（後述）
6	予備 1	未定義。予約されたデータ項目。
7	予備 2	未定義。予約されたデータ項目。
8	$R_{00}$	（後述・以下同様）
9	$R_{01}$	
10	$R_{02}$	
11	$T_x$	
12	$R_{10}$	
13	$R_{11}$	
14	$R_{12}$	
15	$T_y$	
16	$R_{20}$	
17	$R_{21}$	
18	$R_{22}$	
19	$T_z$	

なお  $R$  と  $T$  は次の  $4 \times 4$  の行列の要素を意味する。これは座標変換行列であり、モデルからオブジェクトへの回転と移動を表す。

$$\begin{pmatrix} R_{00} & R_{01} & R_{02} & T_x \\ R_{10} & R_{11} & R_{12} & T_y \\ R_{20} & R_{21} & R_{22} & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

カメラ ID、モデル ID、認識候補番号、座標系番号、エラーコードは整数値をとる。カメラ ID は、入力ポートで受信した `TimedMultiCameraImage` の `camera_set_id` と同じである。モデル ID は認識されたモデルの番号である。座標系番号は座標系を区別するための番号であり、現在は必ず予約された番号 0 になる。

## C) エラーコード

エラーコードは次のように定義されている。

番号	意味
-1	関数の引数が不正
-2	メモリが不足している
-3	ファイルのオープンに失敗した
-4	不正なファイルフォーマット
-101	不正な画像サイズ。画像の幅または高さが 0
-102	画像の不足。入力された画像が 1 枚以下
-103	ステレオ画像のサイズが同じではない
-104	ステレオ画像のカラー形式が同じではない
-105	モデル番号に対応するファイルが見つからない

### 3. 4. 6. サービスポート I/F 仕様

#### A) RecognitionService

関数名	setModelID			
引数	名称	型	I/O	説明
	ModelID	long	Input	認識すべきモデルの番号
戻り値	値			説明
	なし			
説明	新たにモデルを指定し、認識をはじめとする画像処理を開始する。			

関数名	getModelID			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	long			認識すべきモデルの番号
説明	現在、認識の対象として設定されているモデルの番号を取得する。			

#### B) Reconstruct3DService

関数名	reconstruct			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	なし			
説明	3次元距離計測 RTC に対し、3次元距離計測データの送信を要求する。			

### 3. 4. 7. 設定ファイル

この RTC では、RTC のコンフィグレーションによって次の設定ファイルを指定する。

- モデルリストファイル
- 認識パラメータファイル

## A) モデルリストファイル

モデルリストファイルは、モデルに付けた一意の番号とモデルのファイル名の組を一覧にしたテキストファイルである。番号とファイル名は空白で区切り、一行に一組を記述する。以下にモデルリストファイルの例を示す。

105	./model/can190_rtvcm.wrl
109	./model/cube_rtvcm.wrl
117	./model/cookiebox_rtvcm.wrl
402	./model/w2rtvcm.wrl
403	./model/w3rtvcm.wrl
404	./model/w4rtvcm.wrl
409	./model/w9rtvcm.wrl

## B) 認識パラメータファイル

認識パラメータファイルは、認識処理の動作を左右するパラメータをテキストファイルに記述する。設定できる内容について以下に示す。またここに記載されている数値は、パラメータが明示的に与えられなかった場合のデフォルト値である。

## ● RecognitionComponent 用認識パラメータ

## ① 認識に使用するステレオペア

StereoPair	0	0: 左眼 & 右眼
		1: 左眼 & 検証
		2: 右眼 & 検証
		3: 3 眼 (OR)
		4: 3 眼 (AND)

## ② 最大出力候補数 (正の整数)

OutputCandNum	20
---------------	----

## ● 2 次元特徴抽出用パラメータ

## ① エッジ検出アルゴリズム (0: Sobel 3×3, 1: Sobel 5×5)

EdgeDetectFunction	0
--------------------	---

## ② 検出するエッジの最低微分強度 (正の実数)

EdgeStrength	50
--------------	----

## ③ 直線を当てはめるときの最大誤差 ([pixel])

MaxErrorOfLineFit 0.5

- ④ 直線の特徴点を抽出する区間の重複可能な最大比率 (0.0～1.0)

OverlapRatioLine 0.7

- ⑤ 頂点端点間距離の閾値 ([pixel])

HDMax 10.0

- ⑥ 頂点を構成する二次元直線の最小長 ([pixel])

MinLengthLine2D 15.0

- ⑦ 同一の線分と見なす最大端点距離 ([pixel])

MaxDistanceSimilarLine -1.0

-1.0：同一線分を取り除かない

0.0：端点の完全に一致する線分を取り除く

それ以外：指定距離内は同一線分として取り除く

● 楕円検出パラメータ

- ① 楕円誤差評価を行う変数の切り替え(0: 平均誤差, 1: 最大誤差)

IW\_Condition 0

- ② 楕円検出中の最短長さ ([pixel])

IW\_MinLength 20

- ③ 楕円検出後に短い楕円を排除する際の最短長さ ([pixel])

IW\_PostMinLength 50

- ④ 楕円検出中に許す楕円の最小半径 ([pixel])

IW\_MinShortRadPrev 1.0

- ⑤ 楕円検出後に許す楕円の最小半径 ([pixel])

IW\_MinShortRadPost 2.0

- ⑥ 楕円検出中の平均誤差閾値 ([pixel])

IW\_ThMeanError 0.4

- ⑦ 楕円マージ中の平均誤差閾値 ([pixel])

IW\_ThMeanErrorMerging 0.44



- ⑧ 楕円検出中の最大誤差閾値 ([pixel])

IW\_ThMaxError 1.0

- ⑨ 楕円マージ中の最大誤差閾値 ([pixel])

IW\_ThMaxErrorMerging 1.1

- ⑩ 直線あてはめ誤差 ([pixel])

IW\_MinSD 0.5

- ⑪ 点列の積和を計算する際のオフセットモード

IW\_OffsetMode 1 0: static (track 全体で同じオフセット)  
1: dynamic (点列ごとに動的に平均値を反映)

- ⑫ 楕円検出時に使うエッジ画像の選択

IW\_SwLineEllipse 3 1: 直線のあるエッジ画像のみを使う  
2: 直線を消したエッジ画像のみを使う  
3: 両方使う

- ⑬ 古いマージ関数を使うフラグ

IW\_SwOldMergeFunc 1 0: 使わない  
1: 使う

- ステレオ対応用パラメータ

- ① 頂点、線分のなす角度の最小値 ([deg])

AMin 85

- ② 頂点、線分のなす角度の最大値 ([deg])

AMax 95

- ③ ステレオ対応誤差 ([mm]<sup>1</sup>)

StereoError 5.0

- 3次元復元のパラメータ

- ① 検出しない特徴

NoSearchFeatures 0 0: 全て検出  
1: 頂点を検出しない  
2: 楕円を検出しない

<sup>1</sup> カメラキャリブレーション時の単位系に依存

## 3. 5. RecognitionResultViewerComp（認識結果表示 RTC）

### 3. 5. 1. 機能概要

認識結果表示 RTC は、認識結果を視覚的に分かりやすく画面表示する。作業対象認識 RTC の認識結果は位置姿勢を表す数値データであり、見た目から内容を直感的に理解できない。この RTC はステレオ画像、認識結果、モデルが入力されると、モデルを認識された位置へ投影し画像と重ね合わせて緑色のワイヤースケルトンで表示する。

認識結果が空の場合、すなわち作業対象認識 RTC がオブジェクトを見つけられなかった場合には、入力された画像のみを表示する。

### 3. 5. 2. 動作環境

この RTC は次の環境で動作する。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix(32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE (C++)
依存パッケージ	OpenCV 2.0

### 3. 5. 3. ポート情報



図 5 認識結果表示 RTC

A) データポート (InPort)

なし。

B) データポート (OutPort)

なし。

## C) サービスポート (Provider)

サービス名	インターフェース名	説明
RecognitionResult Viewer	RecognitionResultViewerService	ステレオ画像、認識結果を受けて、認識結果画像を作成し、画面に表示する。

## D) サービスポート (Consumer)

なし。

## 3. 5. 4. コンフィグレーション

名称	型	デフォルト値	説明
RecogModelListPath	string	“modelList.txt”	モデルリストファイル名 3.5.7 参照

## 3. 5. 5. 入出力データフォーマット

この RTC に入出力データはない。

## 3. 5. 6. サービスポート I/F 仕様

## A) RecognitionResultViewerService

関数名	display			
引数	名称	型	I/O	説明
	frame	TimedMultiCameraImage	Input	ステレオ画像
	pos	TimedRecognitionResult	Input	認識結果
戻り値	値			説明
	なし			
説明	認識結果に従いモデルを回転・平行移動させ、画像に投影して表示する。			

サービスポートで使用する構造体は、3.1.5(A)(B)と同じである。

## 3. 5. 7. 設定ファイル

この RTC には、RTC のコンフィグレーションによって次の設定ファイルを指定する。

- モデルリストファイル（作業対象認識 RTC と同一）

## 4. ツール仕様

### 4. 1. multicalib（カメラキャリブレーションツール）

#### 4. 1. 1. 機能概要

単純な繰り返しパタンを持つ参照平面の位置・姿勢を変えながら複数回観測したデータを用いて、各カメラの内部パラメータ及び相対位置・姿勢の推定を行う。

#### 4. 1. 2. 動作環境

本ツールの動作環境は以下の通りである。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix(32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	なし
依存パッケージ	OpenCV 2.0

#### 4. 1. 3. 使用方法

キャリブレーション用データを `calib.xcdata` とすると、次のコマンドによってステレオ画像取得 RTC(→ 3. 2.) で使用するカメラキャリブレーションデータ `camera_calib.yaml` が生成できる。

```
$ ./multicalib -o camera_calib.yaml calib.xcdata
```

生成されたキャリブレーションデータに於ける各カメラの相対位置・姿勢は第 1 カメラの光学中心を原点とし、画像面と平行で右方向を  $+x$ 、下方向を  $+y$ 、画面手前から奥へ向かう方向を  $+z$  とする右手座標系を用いて表される。より詳細な使用方法については操作手順書を参照のこと。

#### 4. 1. 4. データファイルのフォーマット

キャリブレーションのためのデータファイルは、以下のテキストファイルである。

MLTCLB 1

格子点の間隔[mm] 格子パタンの回転周期[deg]

観測回数 カメラ台数

第 1 回観測、第 1 カメラの観測点数

パターン x 座標[mm] パタン y 座標[mm] 画像 x 座標[pixel] 画像 y 座標[pixel]

:

第 1 回観測、第 2 カメラの観測点数

:

第 1 回観測、第 m カメラの観測点数

:

第 2 回観測、第 1 カメラの観測点数

:

(以下同様)

## 4. 2. VGRModeler (モデル作成ツール)

### 4. 2. 1. 機能概要

モデル作成ツールは、直方体または円柱の形状を表現するモデルを作成するコマンドラインプログラムである。モデルは VRML ファイルに埋め込まれた独自の形式で表現される。モデルを作成するには、あらかじめ形状のパラメータを記述したモデル設定ファイルを、テキストファイル形式で用意する必要がある。

### 4. 2. 2. 動作環境

このモジュールは、以下の動作環境で動作する。

動作 OS	Ubuntu 10.04 LTS Desktop 日本語 Remix(32bit 環境)
開発言語	C, C++
コンパイラ	GNU Compiler Collection 4.4.3
RT ミドルウェア／バージョン	なし
依存パッケージ	なし

### 4. 2. 3. コマンドラインオプション

モデル作成ツールには、次のコマンドラインオプションがある。

- i (ファイル名) : 入力ファイル名。モデル設定ファイル名を指定する。
- o (ファイル名) : 出力ファイル名。モデルのファイル名を指定する。

例としてモデル設定ファイル名を `in.txt`、モデルのファイル名を `out.wrl` とする場合、次のように VGRModeler をコンソールから実行する。もし `out.wrl` が既に存在する場合、上書きされるので注意すること。

```
$ VGRModeler -i in.txt -o out.wrl
```

### 4. 2. 4. 設定ファイル

このツールに与えるモデル設定ファイルは、以下のフォーマットで記述する。

- (1) 直方体を出力する場合

- ・ 行の先頭に"B"
- ・ 半角スペースを空け、カンマ区切りで幅 (x 軸方向)、奥行き (y 軸方向)、高さ(z 軸方向) ([mm]) を記述

記述例：幅 200 [mm], 奥行き 100 [mm], 高さ 300 [mm]の直方体

```
B 200,100,300
```

(2) 円筒を出力する場合

- ・ 行の先頭に"C"
- ・ 半角スペースを空け、カンマ区切りで半径と高さ ([mm]) を記述
- ・ 上の円の法線が紫色、下の円の法線が黄色で表示される

記述例：半径 100 [mm], 高さ 300 [mm]の円筒

```
C 100,300
```

(3) 円を出力する場合

- ・ 行の先頭に"C"
- ・ 半角スペースを空け、円筒と同じくカンマ区切りで半径と高さ ([mm]) を記述
- ・ 回転の指定が無い場合、法線は z 軸のプラス方向に向く法線の向きを変えるには、回転コマンドで 180 度回転させる

記述例：法線が z 軸マイナス方向に向いた半径 300 [mm]の円

```
C 300,0
R 0,180,0
```

(4) モデルの回転（直方体と円筒共通）

- ・ 回転はモデルの重心で行われる
- ・ 行の先頭に"R"
- ・ 半角スペースを空け、カンマ区切りで x 軸の回転角, y 軸の回転角, z 軸の回転角を度数で記述
- ・ 複数行に分割して記述可能

記述例：直方体を x 軸に 90 度, y 軸に 45 度, z 軸に 45 度回転

```
B 200,100,200
R 45,0,45
R 45,45,0
```

(5) モデルの移動（直方体と円筒共通）

- ・ 行の先頭に"T"

- ・半角スペースを空け、カンマ区切りで  $x, y, z$  の移動量 ([mm]) を記述
- ・複数行に分割して記述可能

記述例：円筒を  $z$  軸に 45 度回転し、 $x$  軸に 10 [mm],  $y$  軸に 20 [mm],  $z$  軸に 30 [mm] 移動

```
C 10, 30  
R 0, 0, 45  
T 10, 20, 30
```



## 5. 特記事項

### 5. 1. カメラの座標系

multicalib（カメラキャリブレーションツール）によって定義されるカメラキャリブレーションデータの座標系は次のようになる。

