

次世代ロボット知能化技術開発プロジェクト  
作業知能（社会・生活分野）の開発

機能仕様書  
PA10 と組み合わせた  
カメラキャリブレーション RTC 群

V e r . 0 . 2 1

2 0 1 1 年 8 月 1 2 日

（独）産業技術総合研究所

知能システム研究部門タスクビジョン研究グループ

## 改版履歷

[illegible]

# 目次

|  |    |
|--|----|
| 改版履歴.....                                    | i  |
| 目次.....                                      | ii |
| 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. モジュール構成 .....                          | 2  |
| 2. 3. ターゲットハードウェア .....                      | 2  |
| 3. RTC仕様 .....                               | 3  |
| 3. 1. VVVCross (クロスマーク検出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. 入出力データフォーマット.....                   | 4  |
| 3. 2. VVVCalibPA10 (キャリブレーション処理統制RTC) .....  | 6  |
| 3. 2. 1. 機能概要 .....                          | 6  |
| 3. 2. 2. 動作環境 .....                          | 6  |
| 3. 2. 3. ポート情報.....                          | 6  |
| 3. 2. 4. コンフィグレーション .....                    | 7  |
| 3. 2. 5. 入出力データフォーマット.....                   | 7  |
| 3. 2. 6. 設定ファイル .....                        | 8  |
| 3. 2. 7. 操作方法 .....                          | 8  |
| 3. 3. VVVCdata2Calib (カメラキャリブレーションRTC) ..... | 9  |
| 3. 3. 1. 機能概要 .....                          | 9  |
| 3. 3. 2. 動作環境 .....                          | 9  |
| 3. 3. 3. ポート情報.....                          | 9  |
| 3. 3. 4. 入出力データフォーマット.....                   | 10 |
| 3. 4. VVVProjMatWriter (射影行列保存RTC) .....     | 11 |
| 3. 4. 1. 機能概要 .....                          | 11 |
| 3. 4. 2. 動作環境 .....                          | 11 |
| 3. 4. 3. ポート情報.....                          | 11 |
| 3. 4. 4. コンフィグレーション .....                    | 12 |
| 4. 特記事項.....                                 | 13 |

# 1. はじめに

## 1. 1. 本書の適用範囲

本書はロボット向けミドルウェア OpenRTM 上で、3 次元点群と対応する画像座標の組からカメラの射影行列を算出するモジュールについて記述したものである。

## 1. 2. 関連文書

本書は以下の文書と関連がある。

表 1-1 関連文書

| No. | 文書名                                 | 備考           |
|-----|-------------------------------------|--------------|
| 1   | ロボットアーム(PA10)分解運動速度制御モジュール<br>動作仕様書 |              |
| 2   | 作業対象物認識モジュール群動作仕様書                  | ステレオ画像取得 RTC |

## 1. 3. 本書を読むにあたって

本書は RT ミドルウェア、RT コンポーネント(以下、RTC)に関する基本知識を備えた利用者を対象としている。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/openrtm/ja/>

## 2. 機能仕様

### 2. 1. 機能概要

本 RTC 群は、ロボットアーム制御モジュール群を用いてハンドに取り付けたマーカの位置を定められた位置へ動かし、環境に据え付けられたステレオカメラを用いて観測することで、カメラの幾何学的キャリブレーションを行う。これによってステレオカメラを用いた 3 次元計測をロボットの座標系で行うことができる。

### 2. 2. モジュール構成

本知能モジュール群はクロスマーク検出 RTC (VVVCross)、キャリブレーション処理統制 RTC (VVVCalibPA10)、カメラキャリブレーション RTC (VVVCdata2Calib)、射影行列保存 RTC (VVVProjMatWriter) で構成される (図 1)。これらをステレオ画像取得 RTC 及び分解運動速度制御モジュール群の操作制御コンポーネント (move) と接続して用いる。

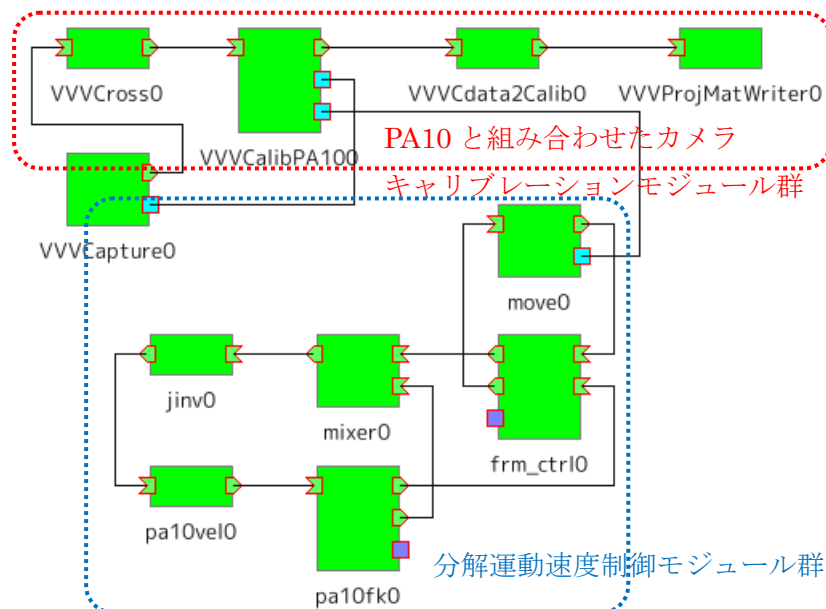


図 1 キャリブレーション RTC 群との接続例

### 2. 3. ターゲットハードウェア

本モジュールは、ロボットアームに取り付けたマーカの位置を制御するために分解運動速度制御モジュール群に含まれる操作制御コンポーネントを用いる。従って、このコンポーネントと同一の機能を持つ RTC とそれに対応するロボットアームが必要である。

## 3. RTC仕様

### 3. 1. VVVCross（クロスマーク検出RTC）

#### 3. 1. 1. 機能概要

入力されたステレオ画像データから、それぞれの画像に対してクロスマーク（図 2）を検出し、その画像座標を出力する。

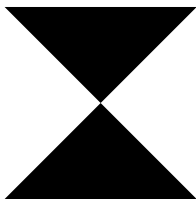


図 2 クロスマーク

#### 3. 1. 2. 動作環境

本知能モジュールの動作環境は以下の通りである。

|                 |                                  |
|-----------------|----------------------------------|
| 動作 OS           | Ubuntu 10.04 LTS (x86)           |
| 開発言語            | C, C++                           |
| コンパイラ           | gcc-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         | 特になし                             |

#### 3. 1. 3. ポート情報



図 3 クロスマーク検出 RTC

##### A) データポート（InPort）

| 名称   | 型         | データ長 | 説明        |
|------|-----------|------|-----------|
| epbm | TimedEPBM | 1    | 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         |
|-------|----------------|------|------------|
| coord | TimedDoubleSeq | 2*n  | 画像座標×カメラ台数 |

## 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 |
|-------|-----------|----|
| なし    |           |    |

## D) サービスポート (Consumer)

なし。

## 3. 1. 4. 入出力データフォーマット

EPBM.idl:

```
typedef sequence<octet> EPBMData;
struct TimedEPBM {
    RTC::Time tm;
    RTC::Time timestamp;
    EPBMData data;
};

interface EPBMSource {
    void sendEPBM();
};
```

EPBMDataはnetpbm形式<sup>1</sup>の画像データをカメラ台数分並べたバイトストリームである。

EPBM データ例 :

|         |   |                |
|---------|---|----------------|
| P5      | } | netpbm 形式画像データ |
| 640 480 |   |                |
| 255     |   |                |
| (画像データ) |   |                |
| P5      | } | 2 台目カメラデータの始まり |
| 640 480 |   |                |
| :       |   |                |
|         |   | netpbm 形式画像データ |

出力されるデータ列 :

|      |      |      |      |      |      |     |
|------|------|------|------|------|------|-----|
| col1 | row1 | col2 | row2 | col3 | row3 | ... |
|------|------|------|------|------|------|-----|

<sup>1</sup> Netpbm home page: <http://netpbm.sourceforge.net/>

$(col_n, row_n)$ は  $n$  台目カメラ画像で検出されたクロスマーカの画像座標である。



## 3. 2. VVVCalibPA10（キャリブレーション処理統制RTC）

### 3. 2. 1. 機能概要

アーム制御モジュールとステレオ画像取得モジュールに適宜コマンドを送信し、得られたデータ列を整形してキャリブレーション用データを生成し、出力を行う。

### 3. 2. 2. 動作環境

本知能モジュールの動作環境は以下の通りである。

|                 |                                  |
|-----------------|----------------------------------|
| 動作 OS           | Ubuntu 10.04 LTS (x86)           |
| 開発言語            | C, C++                           |
| コンパイラ           | gcc-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         | 特になし                             |

### 3. 2. 3. ポート情報

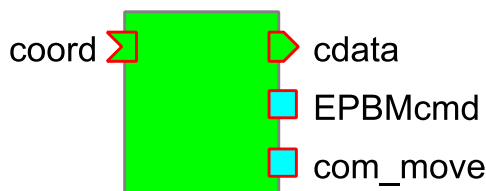


図 4 キャリブレーション処理統制 RTC

#### A) データポート（InPort）

| 名称    | 型              | データ長 | 説明         |
|-------|----------------|------|------------|
| coord | TimedDoubleSeq | 2*n  | クロスマーク検出座標 |

#### B) データポート（OutPort）

| 名称    | 型             | データ長 | 説明               |
|-------|---------------|------|------------------|
| cdata | TimedCdataSeq | 1    | カメラキャリブレーション用データ |

#### 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 |
|-------|-----------|----|
| なし    |           |    |

#### D) サービスポート (Consumer)

EPBMcmd: ステレオ画像データ撮像トリガ (EPBMSource)

com\_move: ロボットハンド目標手先位置・姿勢 (ComMv)

### 3. 2. 4. コンフィグレーション

| 名称       | 型      | デフォルト値   | 説明                     |
|----------|--------|----------|------------------------|
| dry_run  | bool   | 0        | 1 にするとアームの動作のみ行う       |
| pos_file | string | hand.pos | ハンド位置指定ファイル名           |
| wait_ms  | int    | 1000     | 動作完了からキャプチャ開始までの時間[ms] |

### 3. 2. 5. 入出力データフォーマット

クロスマーク検出座標データについては、3. 1. 4. を参照のこと。

Cdata.idl:

```
struct CdataElement{
    double x ;
    double y ;
    double z ;
    double col ;
    double row ;
};
typedef sequence<CdataElement> CdataElementSeq;
struct TimedCdataSeq{
    RTC::Time tm;
    CdataElementSeq data;
};
```

CdataElement は 3 次元点(x,y,z)、対応する画像座標(col,row)の組からなる対応点データであり、これを観測点数だけ並べたものが CdataElementSeq である。計算を実行するためには少なくとも 6 個の観測点が必要である。

### 3. 2. 6. 設定ファイル

キャリブレーション処理統制 RTC は、標準で次の設定ファイルを参照する。

- hand.pos

このファイルはハンドの位置[mm]を並べたテキストファイルであり、この座標と検出されたクロスマークの位置を使ってカメラキャリブレーションを行う。また、手先の姿勢をファイルの先頭に”# a b c” [deg]の形で記述することができる。このとき、操作制御コンポーネントに与えられる姿勢  $R$  は次の式で与えられる。

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix} \begin{bmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{bmatrix} \begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

例：

```
# 0 0 45
700 30 185
650 -20 135
650 -20 235
650 80 135
650 80 235
750 -20 135
750 -20 235
750 80 135
750 80 235
650 30 185
750 30 185
700 -20 185
700 80 185
700 30 135
700 30 235
```

### 3. 2. 7. 操作方法

モジュール群接続後、本モジュールをアクティベートすると、分解運動速度制御モードになっているか確認する旨のメッセージが表示される。ここでリターンキーを押下するとキャリブレーション処理が実行される。設定ファイルを変更した際などは、一度でデアクティベートし、再度アクティベートする。

### 3. 3. VVVCdata2Calib（カメラキャリブレーションRTC）

#### 3. 3. 1. 機能概要

入力データとして与えられた 3 次元空間と画像上の対応点列から、その間の幾何的な関係を表す射影行列を算出し、出力する。

#### 3. 3. 2. 動作環境

本知能モジュールの動作環境は以下の通りである。

|                 |                                  |
|-----------------|----------------------------------|
| 動作 OS           | Ubuntu 10.04 LTS (x86)           |
| 開発言語            | C, C++                           |
| コンパイラ           | gcc-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         | 特になし                             |

#### 3. 3. 3. ポート情報

cdata  proj\_mat

図 5 カメラキャリブレーション RTC

##### A) データポート（InPort）

| 名称    | 型             | データ長 | 説明            |
|-------|---------------|------|---------------|
| cdata | TimedCdataSeq | 1    | 3 次元・画像上の対応点列 |

##### B) データポート（OutPort）

| 名称       | 型              | データ長 | 説明   |
|----------|----------------|------|------|
| proj_mat | TimedDoubleSeq | 12   | 射影行列 |

##### 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 |
|-------|-----------|----|
| なし    |           |    |

##### D) サービスポート（Consumer）

なし。

### 3. 3. 4. 入出力データフォーマット

TimedCdataSeq: 対応点データ

詳細は 3. 2. 5. を参照のこと。

TimedDoubleSeq: 射影行列

proj\_mat.data[i]を $p_i$ とすると、本 RTC の出力は次の射影行列を定める。

$$\lambda \begin{bmatrix} \text{col} \\ \text{row} \\ 1 \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 & p_7 \\ p_8 & p_9 & p_{10} & p_{11} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

### 3. 4. VVVProjMatWriter（射影行列保存RTC）

#### 3. 4. 1. 機能概要

VVVCdata2Calib によって算出された射影行列を受け取り、保存する RTC。ファイル名はベース名+”.”+n (mod num\_cameras)となる。但し、n は得られたデータに対し順に 0 番から割り振られる。

#### 3. 4. 2. 動作環境

本知能モジュールの動作環境は以下の通りである。

|                 |                                  |
|-----------------|----------------------------------|
| 動作 OS           | Ubuntu 10.04 LTS (x86)           |
| 開発言語            | C, C++                           |
| コンパイラ           | gcc-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         | 特になし                             |

#### 3. 4. 3. ポート情報

proj\_mat  

図 6 射影行列保存 RTC

##### A) データポート（InPort）

| 名称       | 型              | データ長 | 説明   |
|----------|----------------|------|------|
| proj_mat | TimedDoubleseq | 12   | 射影行列 |

##### B) データポート（OutPort）

| 名称 | 型 | データ長 | 説明 |
|----|---|------|----|
| 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 |
|-------|-----------|----|
| なし    |           |    |

##### D) サービスポート（Consumer）

なし。

### 3. 4. 4. コンフィグレーション

| 名称          | 型      | デフォルト値 | 説明          |
|-------------|--------|--------|-------------|
| calib_file  | string | Calib  | 保存ファイルのベース名 |
| num_cameras | int    | 3      | カメラ台数       |

## 4. 特記事項

本モジュールをご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本モジュールは独立行政法人 新エネルギー・産業技術総合開発機構の「次世代ロボット知能化技術開発プロジェクト」内実施者向けに評価を目的として提供するものであり、商用利用など他の目的で使用することを禁じます。
- ドキュメントに情報を掲載する際には万全を期していますが、それらの情報の正確性または利用者にとっての有用性等については一切保証いたしません。
- 利用者が本モジュールを利用することにより生じたいかなる損害についても一切責任を負いません。
- 本モジュールの変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本モジュールの情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

### 【連絡先】

独立行政法人 産業技術総合研究所

知能システム研究部門 タスクビジョン研究グループ

河井 良浩

email: irtsp-vvv@m.aist.go.jp