

SingleArUcoComp 仕様書

名城大学理工学研究科

メカトロニクス工学専攻

2018 年 2 月 2 日

内容

1.	はじめに	3
1.1	コンポーネント概要	3
1.2	本書を読むにあたって	3
1.3	動作環境	3
1.4	開発環境	3
2.	RTC 仕様	4
2.1	インタフェース仕様	4
2.2	座標系	4
2.3	ArUco.idl	5
	arUcoDataImage	5
	arUcoData	5
	arUcoPose3D	6
	arUcoPoint2D	6
	rotationMat	6
	markerCorner	7
	pixelPoint2D	7
2.4	独自 IDL を使う際の注意点	7
3.	RTC の導入	8
3.1	OpenCV3.2 の導入	8
3.2	導入	8

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 OpenCV 内に実装されている ArUco マーカの位置・姿勢の推定を行う。

1.2 本書を読むにあたって

本書は、RT ミドルウェアに関する基礎知識を有した利用者を対象としている。

1.3 動作環境

以下の RTC の動作確認環境を以下に示す。

OS	Ubuntu16.04
RT ミドルウェア	OpenRTM-aist-1.1.2
OpenCV	OpenCV3.2

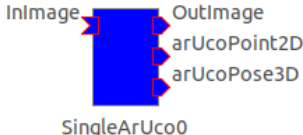
1.4 開発環境

以下の RTC の開発環境を以下に示す。

OS	Ubuntu16.04
RT ミドルウェア	OpenRTM-aist-1.1.2
OpenCV	OpenCV3.2
言語	C++

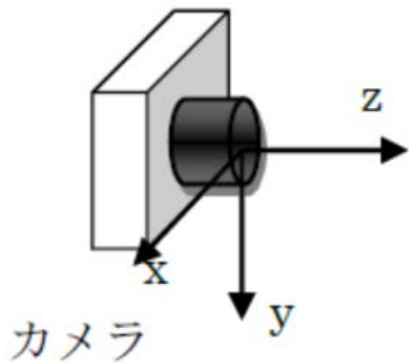
2. RTC 仕様

2.1 インタフェース仕様

RTC の名称			
SingelArUco			
入力ポート			
名称	データ型	説明	
InImage	Img/TimedCameraImage	入力画像	
出力ポート			
名称	データ型	説明	
OutImage	Img/TimedCameraImage	検出結果画像	
arUcoPoint2D	arUco/arUcoPoint2D	マーカの画像内情報を出力	
arUcoPose3D	arUco/arUcoPose3D	マーカの 3 次元情報を出力	
主なコンフィグレーション			
名称	データ型	デフォルト	説明
Dictionary	String	ORIGINAL	マーカのディクショナリの指定
MarkerSize	double	0.05	マーカのサイズ[m]

2.2 座標系

カメラの座標系は下図のようにになっている。ArUco マーカの座標原点はマーカの中心にある。



2.3 ArUco.idl

本章では、独自データポートを宣言している、ArUco.idl について説明をする。

sequence は配列を表し、double 型の配列の場合、

sequence<double>

と表記される。

arUcoDataImage

本コンポーネントでは未実装のデータポート。マーカ ID、マーカコーナ、検出元画像、回転行列、並進行列を含むデータポート。マーカ関連の情報をすべて同期させて取ることを想定。

データ名	データ型	説明
tm	RTC::Time	タイムスタンプ
ids	sequence<long>	マーカ ID
markerCorners	sequence<markerCorners>	マーカコーナ
data	Img::CameraImage	検出元画像
rotations	sequence<rotationMat>	回転行列
translates	sequence<RTC::Vector3D>	並進行列

arUcoData

本コンポーネントでは未実装のデータポート。マーカ ID、マーカコーナ、回転行列、並進行列を含むデータポート。arUcoDataImage から検出元画像を除外したデータポート。巢すべて同期して取りたいが、画像は重いのでいない場合を想定。

データ名	データ型	説明
tm	RTC::Time	タイムスタンプ
ids	sequence<long>	マーカ ID
markerCorners	sequence<markerCorners>	マーカコーナ
rotations	sequence<rotationMat>	回転行列
translates	sequence<RTC::Vector3D>	並進行列

arUcoPose3D

本コンポーネントでは、出力データポートの arUcoPose3D で用いられているデータ型。3次元に関するデータを中心に含む。

データ名	データ型	説明
tm	RTC::Time	タイムスタンプ
ids	sequence<long>	マーカ ID
rotations	sequence<rotationMat>	回転行列
translates	sequence<RTC::Vector3D>	並進行列

arUcoPoint2D

本コンポーネントでは、出力データポートの arUcoPoint2D で用いられているデータ型。2次元に関するデータを中心に含む。

データ名	データ型	説明
tm	RTC::Time	タイムスタンプ
ids	sequence<long>	マーカ ID
markerCorners	sequence<markerCorners>	マーカコーナ

rotationMat

マーカの回転行列を表すデータ型。

データ名	データ型	説明
R11	double	回転行列の 1 行 1 列要素
R12	double	回転行列の 1 行 2 列要素
R13	double	回転行列の 1 行 3 列要素
R21	double	回転行列の 2 行 1 列要素
R22	double	回転行列の 2 行 2 列要素
R23	double	回転行列の 2 行 3 列要素
R31	double	回転行列の 3 行 1 列要素
R32	double	回転行列の 3 行 2 列要素
R33	double	回転行列の 3 行 3 列要素

markerCorner

マーカのコーナの画像内座標を表すデータ型。

データ名	データ型	説明
point1	pixelPoint2D	ピクセル位置 (x,y)
point2	pixelPoint2D	ピクセル位置 (x,y)
point3	pixelPoint2D	ピクセル位置 (x,y)
point4	pixelPoint2D	ピクセル位置 (x,y)

pixelPoint2D

ピクセル位置 (x,y) をあらわすデータ型。

データ名	データ型	説明
point1	pixelPoint2D	ピクセル位置 (x,y)
point2	pixelPoint2D	ピクセル位置 (x,y)

2.4 独自 IDL を使う際の注意点

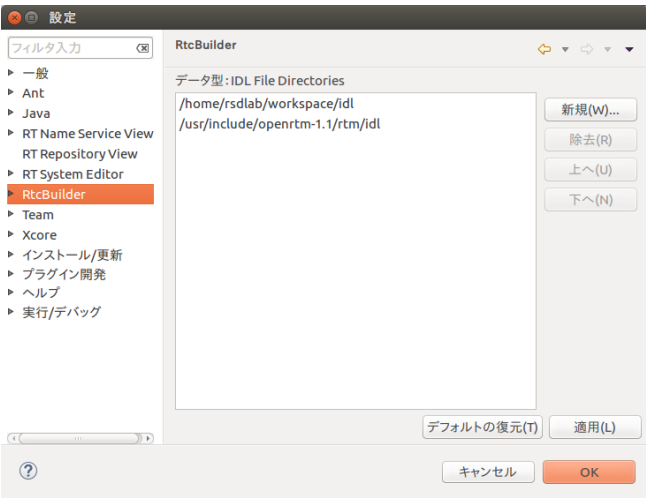
今回使用した ArUco.idl のような、ユーザが独自に定義した IDL を使ったコンポーネントを RTCBuilder で作成する場合、IDL の登録が必要である。

RTCBuilder を開き、

ウィンドウ->設定

を開く

設定ウィンドウが開いたら、RTCBuilder を選択肢し、新規から独自 IDL があるディレクトリを指定する。



3. RTC の導入

3.1 OpenCV3.2 の導入

ロボットシステムデザイン研究室のホームページを参照

(http://www2.meijo-u.ac.jp/~kohara/cms/technicalreport/ubuntu1404_opencv32_setup)

ホームページでは Ubuntu14.04 への導入だが、Ubuntu16.04 にも導入することができる。ただし、不必要なパッケージも含まれている。

3.2 導入

ダウンロードしてきたファイルコンポーネントファイルの階層で、以下のコマンドを実行する。

```
$ mkdir build  
$ cd build  
$ cmake ../  
$ make
```

実行ファイルは build/src 内に生成される。