

```
#####  
# ARToolKit マーカ検出・位置姿勢推定コンポーネントの使い方 #  
# 2011/03/30 #  
#####
```

1. 概要

本プログラムは ARToolKit を利用したマーカ検出・位置姿勢推定 RT コンポーネントです。画像の入出力には TimedCameraImage データ型と CameraCaptureService インタフェースを利用します。

2. 依存ライブラリ

ARToolKit 2.72.1 (その他のバージョンでは未検証)

OpenGL, GLUT, GLEW

OpenCV 2.0.0 以上

3. コンパイル

(1)ARToolKit のコンパイル

ARToolKit-2.72.1.tgz をダウンロードする。

ダウンロード先 : <http://sourceforge.net/projects/artoolkit/files/artoolkit/>

```
$ sudo aptitude update
```

```
$ sudo aptitude install libglut3 libglut3-dev
```

```
$ sudo aptitude install libxi-dev libxmu-dev libjpeg-dev
```

```
$ sudo aptitude install patch
```

```
$ tar xvfz ARToolKit-2.72.1.tgz
```

同封の artk-v4l2-2.72.1.20101026.patch を ARToolKit を展開したところと同じディレクトリに置き、パッチを当てる。

```
$ patch -p1 -d ARToolKit < artk-v4l2-2.72.1.20101026.patch
```

```
$ cd ARToolKit
```

```
$ ./Configure
```

```

-----

Select a video capture driver.
  1: Video4Linux
  2: Video4Linux+JPEG Decompression (EyeToy)
  3: Video4Linux2
  4: Digital Video Camcorder through IEEE 1394 (DV Format)
  5: Digital Video Camera through IEEE 1394 (VGA NONCOMPRESSED
Image Format)
  6: GStreamer Media Framework
Enter : 3

Color conversion should use x86 assembly (not working for 64bit)?
Enter : y
Do you want to create debug symbols? (y or n)
Enter : n
Build gsub libraries with texture rectangle support? (y or n)
GL_NV_texture_rectangle is supported on most NVidia graphics cards
and on ATi Radeon and better graphics cards
Enter : y

```

kernel アップデートを実施した際には、以下、二つの処理は再度実行する必要がある

```

$ sudo ln -s /usr/src/linux-headers-`uname -r`/include/asm-generic/linkage.h
/usr/src/linux-headers-`uname -r`/include/asm/linkage.h
$ make

```

(2)その他の依存ライブラリのインストール

```

$ sudo aptitude update
$ sudo aptitude install libglew1.5 libglew1.5-dev
$ sudo aptitude install libcv4 libhighgui4 libcvaux4 libcv-dev libhighgui-dev
libcvaux-dev opencv-doc

```

(3)Makefile の編集

- GUI の有無の指定

Makefile 内の WITH_GUI を yes にするとウィンドウを表示し、出力画像を描画する。

- ARToolKit の PATH を指定する

```
ARTOOLKIT_INC_DIR = /<ARToolKit Root>/include
```

```
ARTOOLKIT_LIB_DIR = /<ARToolKit Root>/lib
```

(4)コンパイル

```
cd src/
```

```
make -f Makefile.ARMarkerDetectSimple
```

4. 実行

(1)AR マーカの準備

現状では、<ARToolKit Root>/bin/mk_patt プログラムを利用する必要がある。

手順 1. <ARToolKit Root>/bin/calib_camera2 によりマーカモデルの登録を行うカメラのカメラパラメータを求める。この際に、
<ARToolKit Root>/patterns/calib_dist.pdf のキャリブレーションパターン
(6x4 の格子点) を使用する。

手順 2. <ARToolKit Root>/bin/mk_patt プログラムに先ほど生成したカメラパラメータを読み込ませ、登録したいマーカをカメラに見せた上でマーカをファイルを生成する。

詳しくは、「11.参考資料」の「工学ナビ ARToolKit を使った拡張現実感プログラミング」を参照。

(2)マーカ設定ファイルの記述

「5.設定」の「marker_info_filename」の項を参照

現在は、<ARToolKit Root>/patterns/pattHiro.pdf および

<ARToolKit Root>/patterns/pattMulti.pdf の二パターンのマーカを検出する
マーカ設定ファイルが付属しております。

(3)コンポーネントの実行

```
./ARMarkerDetectSimple
```

(4)マーカ検出が実行される条件

AR マーカ検出 RTC は画像データを受信するとマーカ検出処理が実行される。

マーカ検出処理が実行される条件（画像データが受信される条件）は、以下のいずれかの条件に合致する時である。

動作条件	キャプチャ RTC 動作状態	AR マーカ検出 RTC 動作状態	ObjectRecognitionService Consumer
1	画像送信	-	-
2	指令待ち待機	continuous_flag = 1	-
3	指令待ち待機	continuous_flag = 0	setModelID()呼び出し

(5)コンポーネント接続例

- example_img/example1.png

キャプチャ RTC を能動的かつ連続的に実行させ、ある一つのマーカ ID に対してのみ位置姿勢推定を実行し、マーカ位置姿勢を取得する場合。指定したいマーカ ID は、ARMarkerDetectSimple コンポーネントのコンフィギュレーションにより指定します。

- example_img/example2.png

キャプチャ RTC を能動的かつ連続的に実行させ、複数のマーカ ID に対して位置姿勢推定を実行し、検出されたマーカ全ての位置姿勢を取得する場合。

- example_img/example3.png

キャプチャ RTC を能動的かつ連続的に実行させ、複数のマーカ ID に対して位置姿勢推定を実行し、ある指定した一つのマーカ ID の位置姿勢を取得する場合。指定したいマーカ ID は、ObjectRecognitionService の Consumer 側 (ARMarkerDetectTest コンポーネント) により指定します。

- example_img/example4.png

キャプチャ RTC を受動的に実行させ、キャプチャ指令はマーカ検出指令に応じて発行させる場合。

マーカ検出指令は検出したいマーカ ID と共に ObjectRecognitionService の Consumer 側 (ARMarkerDetectTest コンポーネント) より送信する。

ObjectRecognitionService からマーカ ID を設定すると画像キャプチャ指令 (画像 1 枚) が発行される。

5. 設定

コンフィギュレーションセットを設定する

- marker_info_filename

ARToolKit マーカ情報を記述するファイル名を指定するオプション

検出したいマーカセットファイルを指定 ID 番号と共に 1 行で記述する.

例)

```
# ID MarkerSetFilename
0 Data/multi/marker.dat
10 Data/multi/marker2.dat
```

マーカセットファイルの記述方法

```
#the number of patterns to be recognized      <-- # 行はコメント
2                                              <-- ファイルに記述されたマーカの数
                                              下方に記述するマーカ設定はこの数だけ記
                                              述する
                                              マーカ 1 個で記述する場合は, 1 を指定する
                                              こと

#marker 1
Data/multi/patt.a                        <-- ARToolKit マーカファイル名
40.0                                     <-- マーカの矩形サイズ [mm]
0.0 0.0                                 <-- マーカ原点のオフセット量 (0,0)がマーカ中心
1.0000 0.0000 0.0000 0.0000          <-- マーカ原点の三次元位置姿勢
0.0000 1.0000 0.0000 0.0000          単位行列を指定するとマーカ位置を原点とする
0.0000 0.0000 1.0000 0.0000          カメラ位置姿勢を出力することになる

#marker 2
Data/multi/patt.b
40.0
0.0 0.0
1.0000 0.0000 0.0000 100.0000 <-- 全てのマーカは剛体上に張り付けられて
                                いる (相対的な位置姿勢は変化しない)
0.0000 1.0000 0.0000 0.0000      として扱われる
0.0000 0.0000 1.0000 0.0000
```

- **default_detect_marker_id**

デフォルトの検出マーカ ID を指定するオプション

サービスポート **RecognitionService** より ID が指定された場合は、サービスポート側が優先される。

- **camera_parameter_filename**

カメラパラメータファイル名を指定するオプション

ただし、入力ポートの **TimedCameraImage** データ型からカメラパラメータを取得した場合は、データポートの値に上書きされる。

- **continuous_flag**

実行時に連続検出を有効化するオプション

0 を指定すると、アクティブ時はサービスポート **ObjectRecognitionService** より、**setModelID0**が指定されるまで何もせず待機状態となる。

1 を指定すると、カメラキャプチャ **RTC** のサービスポート **CameraCaptureService** に連続キャプチャを指定し、マーカ検出も連続的に実行する。

6. 入出力仕様

6.1 入力ポート

- **image**: 画像データ

-- **TimedCameraImage** 詳細のデータ構造は **Img.idl** を参照。

--- **RTC::Time tm** # RTM 形式の時間(キャプチャ時間と同じ)

--- **CameraImage data** # カメラ画像データ

---- **RTC::Time captured_time** # キャプチャ時間

---- **ImageData image** # 画像データ

---- **CameraIntrinsicParameter intrinsic** # カメラの内部パラメータ

----- **double matrix_element[5]** # カメラの内部行列

a[0] a[1] a[3]

0 a[2] a[4]

0 0 1

----- **sequence<double> distortion_coefficient** # レンズ歪パラメータ

OpenCV のものと同じ, k1, k2, p1, p2 [, k3]

---- **Mat44 extrinsic** # カメラの外部パラメータ

--- **long error_code** # エラーコード (使用していない)

6.2 出力ポート

- imageAR: 直方体を重畳した AR 画像
- TimedCameraImage 詳細のデータ構造は `Img.idl` を参照.
- RTC::Time tm # RTM 形式の時間(キャプチャ時間と同じ)
- CameraImage data # カメラ画像データ
- RTC::Time captured_time # キャプチャ時間
- ImageData image # 画像データ
- CameraIntrinsicParameter intrinsic # カメラの内部パラメータ
- double matrix_element[5] # カメラの内部行列
- a[0] a[1] a[3]
- 0 a[2] a[4]
- 0 0 1
- sequence<double> distortion_coefficient # レンズ歪パラメータ
- OpenCV のものと同じ, k1, k2, p1, p2 [, k3]
- Mat44 extrinsic # カメラの外部パラメータ
- long error_code # エラーコード (使用していない)

- markerPose: 指定されたマーカ ID の位置姿勢
- TimedDoubleSeq (データ数 12)
- 以下の行列があった場合,
- r1 r2 r3 t1
- r4 r5 r6 t2
- r7 r8 r9 t3
- 0 0 0 1
- r1, r2, r3, t1, r4, r5, r6, t2, r7, r8, r9, t3
- として出力する.

- markerPoseAll: 検出されたマーカ全てのマーカ ID および位置姿勢
- TimedObjectPoseMatSeq 詳細のデータ構造は `MyDataType.idl` を参照.
- RTC::Time tm # RTM 形式の時間
- sequence<ObjectPoseMat> data # ID 付き位置姿勢のデータ列
- long id # マーカ ID
- sequence<double> pose # 位置姿勢 (データ数 12) markerPose の data
- と同じフォーマット

6.3 サービスポート (Provider)

- インタフェース名:RecognitionService インタフェース型:RecognitionService
- long getModelID(); # 検出対象として設定されているモデル ID(マーカ ID)を取得する
- void setModelID(in long ModelID); # 検出対象のモデル ID(マーカ ID)を設定する

6.4 サービスポート (Consumer)

- インタフェース名:CameraCaptureService
インタフェース型:Img::CameraCaptureService
- void take_one_frame(); # 画像 1 枚のみキャプチャ
- void take_multi_frames(in long num); # 画像を指定枚数キャプチャ
- void start_continuous(); # 画像の連続取得開始
- void stop_continuous(); # 画像の連続取得終了

7. RTC::Logger ログ

- ERROR
- Unable to create an instance of MarkerDetect Class
MarkerDetect クラスのインスタンスの生成に失敗した際に出力される。
(onInitialize)
- Could not load camera parameter [<filename>]
カメラパラメータファイルの読み込みに失敗した際に出力される。(onActivated)
- Could not load marker config [<filename>]
AR マーカ情報記述ファイルを読み込めなかった際に出力される。(onActivated)
- Unable to set camera parameter.
カメラパラメータの再設定に失敗した際に出力される。(onExecute)
- Camera parameter has not be setted yet. OutPort of ImageAR is disabled.
カメラパラメータが設定される前に描画初期化を呼び出した際に出力される。
(onExecute)
マーカ位置姿勢推定の成否には関係ないが、AR 画像 imageAR の出力が正常に行われない場合がある。
- Unable to use framebuffer object. OutPort of ImageAR is disabled.
利用している GPU で Frame buffer object が利用できない場合に出力される。
(onExecute)
マーカ位置姿勢推定の成否には関係ないが、AR 画像 imageAR の出力が正常に行われない場合がある。

- WARN
 - Unable to finish detection process normally.
AR マーカ検出が正常に終了しなかった際に出力される. (onExecute)
この警告出力時はマーカの位置姿勢が計算されていない.
 - AR image is empty
AR 重畳画像が空であった場合に出力される. (onExecute)
- INFO
 - Create an instance of MarkerDetect Class
MarkerDetect クラスのインスタンスを生成した際に出力される. (onInitialize)
 - Initialize glut
glut の初期化が完了した際に出力される. (onInitialize)
 - Destroy an instance of MarkerDetect Class
MarkerDetect クラスのインスタンスを生成した際に出力される. (onFinalize)
 - Marker config including <num> markers was loaded
AR マーカ情報記述ファイルを読み込んだ際に出力される. (onActivated)
 - AR marker data is released
AR マーカ情報は破棄した際に出力される. (onDeactivated / onAborting)
 - Send command of "start continuous" via CameraCaptureService.
サービスポート CameraCaptureService に「start continuous」コマンドを発行した際に出力される. (onActivated)
 - Send command of "stop continuous" via CameraCaptureService.
サービスポート CameraCaptureService に「stop continuous」コマンドを発行した際に出力される. (onDeactivated)
- DEBUG
 - Send command of "take one frame" via CameraCaptureService.
サービスポート CameraCaptureService に「take one frame」コマンドを発行した際に出力される. (onExecute)
 - Get & read image.
画像データを受信した際に出力される. (onExecute)
 - Color conversion GRAY -> RGB
画像データをグレースケールから RGB に変換した際に出力される. (onExecute)
 - Specified Marker ID [<marker id>] is detected
指定されたマーカ ID が検出された際に出力される. (onExecute)
 - Total of detected marker is <marker num>
検出されたマーカセットの数が出力される. (onExecute)

-- Marker ID [<marker id>] is detected
 登録済みマーカーが検出された際に、全てのマーカー ID に対して出力される.
 (onExecute)

-- Overlay CG image
 CG 画像を重畳処理した際に出力される. (onExecute)

-- AR image displayed
 AR 重畳画像が表示された際に出力される. (onExecute)

- TRACE

-- ARToolKit perspective matrix
 <r1> <r2> <r3> <t1>
 <r4> <r5> <r6> <t2>
 <r7> <r8> <r9> <t3>
 ARToolKit distortion factor
 <k1> <k2> <p1> <p2>
 読み込まれたカメラパラメータの情報が出力される. (onActivated / onExecute)

-- Width <width>, Height <height>, Channels <channels>
 読み込まれた画像データの幅, 高さ, チャンネル数が出力される. (onExecute)

-- Pose of specified marker ID [<marker id>]
 <r1> <r2> <r3> <t1>
 <r4> <r5> <r6> <t2>
 <r7> <r8> <r9> <t3>
 指定されたマーカー ID が検出された際に位置姿勢が出力される. (onExecute)

-- Pose of marker ID [<marker id>]
 <r1> <r2> <r3> <t1>
 <r4> <r5> <r6> <t2>
 <r7> <r8> <r9> <t3>
 検出された全てのマーカー ID の位置姿勢が出力される. (onExecute)

8. 開発環境

- ubuntu 10.04
- OpenRTM-aist 1.0.0
- omniORB 4.1.2-1
- RTCBuilder 1.0.0
- G++コンパイラ 4.4.3
- OpenCV 2.2.0 / 2.0.0
- ARToolKit 2.72.1
- GLUT 3.7-25
- GLEW 1.5.2-0

9. ディレクトリ構成

- src : RTCBuilder で生成したひな形に実処理を実装したソースコード
- src/Data : ARToolKit 用データ格納ディレクトリ
- src/log : RTC::Logger ログの出力用ディレクトリ
- test : テスト用プログラム
- example_img : RTC 動作例の画像 (RTC 接続例, マーカ検出時の画像)

10. テスト用プログラム

10.1 ARMarkerDetect テスト用コンポーネント (ar_marker_detect_test)

10.1.1 概要

位置姿勢の入出力テストを行うコンポーネントである.

ARMarkerDetectSimple が出力ポートより送信する位置姿勢データを取得・表示する.

10.1.2 依存ライブラリ

特になし

10.1.3 コンパイル

```
cd test/ar_marker_detect_test
make -f Makefile.ARMarkerDetectTest
```

10.1.4 実行

```
./ARMarkerDetectTestComp
```

10.1.5 設定

コンフィギュレーションセットを設定する

- detect_marker_id

検出したいマーカ ID を指定.

指定したマーカ ID をサービスポート **ObjectRecognitionService** を通して指定する.

10.1.6 入出力仕様

10.1.6.1 入力ポート

- **markerPose**: 指定したマーカ ID の位置姿勢

-- **TimedDoubleSeq** (データ数 12)

以下の行列があった場合,

r1 r2 r3 t1

r4 r5 r6 t2

r7 r8 r9 t3

0 0 0 1

r1, r2, r3, t1, r4, r5, r6, t2, r7, r8, r9, t3

として出力する.

- **markerPoseAll**: 検出されたマーカ全てのマーカ ID および位置姿勢

-- **TimedObjectPoseMatSeq** 詳細のデータ構造は **MyDataType.idl** を参照.

--- **RTC::Time tm** # RTM 形式の時間

--- **sequence<ObjectPoseMat> data** # ID 付き位置姿勢のデータ列

---- **long id** # マーカ ID

---- **sequence<double> pose** # 位置姿勢(データ数 12) **markerPose** の **data** と同じフォーマット

10.1.6.2 サービスポート (Consumer)

- インタフェース名:**RecognitionService**

インタフェース型:**RecognitionService**

-- **long getModelID();** # 検出対象として設定されているモデル ID(マーカ ID)を取得する

-- **void setModelID(in long ModelID);** # 検出対象のモデル ID(マーカ ID)を設定する

10.1.7 RTC::Logger ログ

- WARN

-- Data size is wrong. Pose should have 12 params

受信データに含まれる位置姿勢のデータ数が一致しないときに出力される。

データ数は 4x4 行列の最後の行を抜いた 12 個の要素であるべき。

- DEBUG

-- ID was set to <src>, [<dst>]

サービスポート経由で検出対象のマーカ ID を指定した際に出力される。
(onExecute)

-- Total of detected marker is <marker num>

検出されたマーカセットの数が出力される。(onExecute)

-- Marker ID [<marker id>] is detected

受信した全てのマーカ ID に対して出力される。(onExecute)

-- Specified Marker ID [<marker id>] is detected

指定されたマーカ ID が検出された際に出力される。(onExecute)

- TRACE

-- Pose of marker ID [<marker id>]

<r1> <r2> <r3> <t1>

<r4> <r5> <r6> <t2>

<r7> <r8> <r9> <t3>

受信した全てのマーカの位置姿勢がマーカ ID と共に出力される。

(onExecute)

-- Pose of specified marker ID [<marker id>]

<r1> <r2> <r3> <t1>

<r4> <r5> <r6> <t2>

<r7> <r8> <r9> <t3>

指定したマーカ ID の位置姿勢を受信した際に位置姿勢が出力される。

(onExecute)

11. 参考資料

- ARToolKit Official Site

<http://www.hitl.washington.edu/artoolkit/>

- 工学ナビ ARToolKit を使った拡張現実感プログラミング

<http://kougaku-navi.net/ARToolKit.html>

12. ChangeLog

- 2011.03.30

描画処理エラーが発生する際のエラーログを `RTC::Logger` のログメッセージに追加.

- 2011.03.03

追加ファイルの不備、ドキュメントの不備を修正.

バージョン 0.2.0a としてリリース

- 2011.02.16

ARToolKit マーカ検出・位置姿勢推定コンポーネントのプロトタイプ版の完成

バージョン 0.2.0 としてリリース

13. 特記事項

本モジュールのライセンスは、ARToolKit のライセンス (GPL) に従います。本モジュールの改変・再配布等については、GPL に従って行ってください。

【連絡先】

奈良先端科学技術大学院大学 情報科学研究科 ロボティクス研究室

〒630-0192 奈良県 生駒市 高山町 8916-5

TEL: 0743-72-5376 FAX: 0743-72-5379 Email: robotics-staff@is.naist.jp