次世代ロボット知能化技術開発プロジェクト

操作手順書

単眼位置姿勢計測・表示モジュール

Ｖｅｒ．１.３

２０１２年2月８日

（株）東芝

改版履歴

Ver.	改版日	改版内容
1.0	2011/4/1	新規作成
1.1	2011/4/7	Naming Sever の説明図を RTM1.0 版に変更
1.2	2011/4/8	画面の画素数を 320x240 から 640x480 に修正
1.3	2012/2/8	バイナリ公開に向け、特記事項を変更

目次

１．　はじめに 4
　１．１．　本書の適用範囲 4
　１．２．　関連文書 4
　１．３．　本書を読むにあたって 4
　１．４．　動作環境 4
２．　ディレクトリ構成 5
３．　ソフトウェアインストール 6
　３．１．　基本環境 6
　３．２．　実機環境 7
４．　事前準備 8
　４．１．　ＵＳＢカメラのドライバインストール 8
　４．２．　カメラのセッティング 8
５．　実行 9
　５．１．　カメラの起動 9
　５．２．　Nameサーバー起動 9
　５．３．　MarkerRecognition RTC起動 9
　５．４．　ObjectPositionTest RTC 11
　５．５．　RecognitionServiceTest RTC 12
　５．６．　FingerPositionTest RTC 13
　５．７．　接続 14
　５．８．　活性化 15
　５．９．　マーカ形状の選択 15
　５．１０．　ロボット手先位置座標系の回転 18
　５．１１．　結果表示 19
　５．１２．　ＲＴＣの終了手順 19
６．　特記事項 20

１．　はじめに

１．１．　本書の適用範囲

本書は、単眼位置姿勢計測表示モジュールの操作手順について記述した文書である。単眼位置姿勢計測表示モジュールは、予め登録された多角形マーカの三次元位置・姿勢を算出し、センス系統一ＩＦの仕様で出力するものである。

１．２．　関連文書

本書の関連文書は 下表の通り。

No.	文書名	備考
1	単眼位置姿勢計測表示モジュール 機能仕様書	単眼位置姿勢計測表示モジュールの機能仕様について記載。

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

URL：http://www.openrtm.org/

１．４．　動作環境

検証に用いた動作環境は以下のとおりである。

動作 OS	Windows (Xp,sp2 で動作確認済み)
開発言語	C++
コンパイラ	VisualC++2008、VisualC++2010
RT ミドルウェア／バージョン	OpenRTM-aist 1.0.0
依存パッケージ	OpenCV2.2 http://sourceforge.net/projects/opencvlibrary/ Freeglut http://sourceforge.jp/projects/sfnet_freeglut/releases/

２．　ディレクトリ構成

本書では下表のディレクトリ構成を推奨するが、OpenCV2.2 の dll ファイルを MarkerRecognition のディレクトリに入れておけば、この他の構成でも起動可能である。

ディレクトリ	言語	内容	備考
C:¥ProgramFiles			
├ OpenＣＶ2.2	C++	OpenＣＶ2.2 関連ファイル	
├ OpenRTM-aist	C++	RTM 本体	
├1.0		Ver1.0.0	
├examples			
├C++			
├MarkerRecognition	C++	単眼位置姿勢計測・表示モジュール	
├MarkerRecognitionComp.exe	C++	単眼位置姿勢計測・表示モジュール本体	
├FingerPositionTestComp.exe	C++	指先位置テストモジュール	
├ObjectPositionTestComp.exe	C++	物体位置姿勢表示モジュール	
├RecognitionServiceTestComp.exe	C++	マーカ形状変更モジュール	
├ViewSimulatorTestComp.exe	C++	ビューシュミレータ用モジュール	
├ViewSimulatorTestViewerComp.exe	C++	ビューシュミレータ用モジュール	
├freeglut.dll		ＣＧ用動的ライブラリ	
├MarkerRecognition.dll		単眼位置姿勢計測・表示動的ライブラリ	
├rtc.conf		rtc.conf	
├（OpenCV2.2 系 dll）			

３．　ソフトウェアインストール

動作に必要なソフトウェアを以下に記す。

OpenRTM（1.0）、OpenCV2.2、Freeglut

は上記のディレクトリ構成内にインストールすること（下記 3.1 参照）。

全てのソフトウェアのインストールが完了したら再起動すること。

３．１．　基本環境

あらかじめ、MarkerRecognition を実行する PC 本体に USB カメラが接続されていることを確認してください。

Windows 環境に下記のライブラリを順番にインストールしてください。

- OpenRTM-aist-1.0.0-RELEASE
- http://www.openrtm.org/pub/Windows/OpenRTM-aist/cxx/OpenRTM-aist-1.0.0-RELEASE_vc9_100212.msi
- Python 2.6.4
- http://www.python.org/ftp/python/2.6.4/python-2.6.4.msi
- PyYAML
- http://pyyaml.org/download/pyyaml/PyYAML-3.09.win32-py2.6.exe
- OpenRTM-aist-Python-1.0.0-RELEASE
- http://www.openrtm.org/pub/Windows/OpenRTM-aist/python/OpenRTM-aist-Python-1.0.0.msi
- rtctree
- http://www.openrtm.org/pub/OpenRTM-aist/tools/1.0.0/rtctree-3.0.0.win32.exe
- rtsprofile
- http://www.openrtm.org/pub/OpenRTM-aist/tools/1.0.0/rtsprofile-2.0.0.win32.exe
- rtshell
- http://www.openrtm.org/pub/OpenRTM-aist/tools/1.0.0/rtshell-3.0.0.win32.exe
- OpenCV2.2

  http://sourceforge.net/projects/opencvlibrary/
- Freeglut

  http://sourceforge.jp/projects/sfnet_freeglut/releases/

３．２．　実機環境

本知能モジュールは、市販のＵＳＢカメラ(画素数 640×480)の使用を想定しており、OpenCVのキャプチャ関数と接続可能なものなら、どのカメラでもかまわない。
図１に本ＲＴＣのハードウェア構成図を示す。

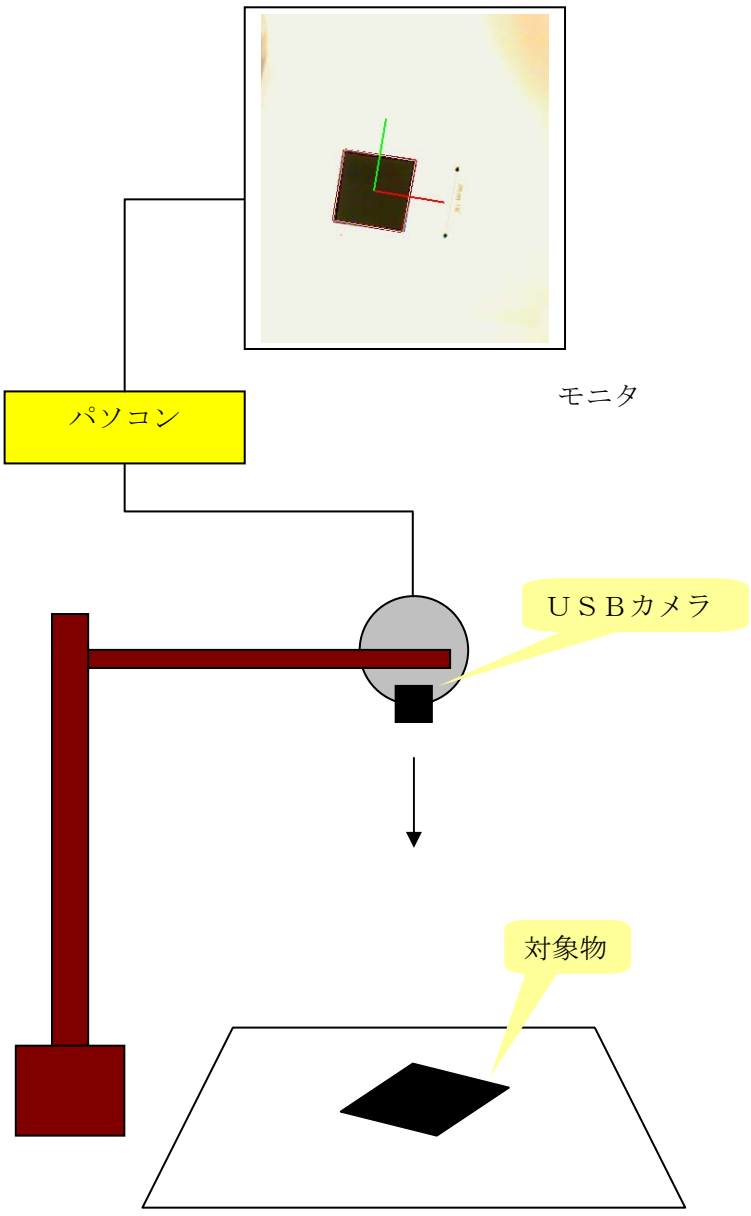

図１　本ＲＴＣのハードウェア構成図

４．　事前準備

実際に動作させる前に行う設定を以下に記す。

４．１．　ＵＳＢカメラのドライバインストール

適宜ＣＤ－ＲＯＭなどからドライバーをインストールする。

４．２．　カメラのセッティング

カメラをスタンドやロボットなどに搭載し、対象物がカメラから撮影できることを確認する。

５．　実行

５．１．　カメラの起動

カメラのスイッチがある場合はＯＮする。

５．２．　Name サーバー起動

ポート番号を****に指定して CORBA ネームサーバを起動する。

```
Starting omniORB omniNames: PCname:****

Thu Mar __ :__:__ 2011:

Starting omniNames for the first time.
Wrote initial log file.
Read log file successfully.
Root context is IOR:010000002b00000049444c3a6f6d672e6f72672f436f734e616d696e672f
4e616d696e67436f6e746578744578743a312e3000000100000000000000700000000010102000e00
00003133332e3139362e38392e383600f90a0b0000004e616d6553657276696365000300000000000
0000080000000100000000545441010000001c0000000100000001000100010000000100010509010
0100010000000901010003545441080000059699d4d01000c2c
Checkpointing Phase 1: Prepare.
Checkpointing Phase 2: Commit.
Checkpointing completed.
```

図 2　ネームサーバの表示

５．３．　MarkerRecognition RTC 起動

MarkerRecognition RTC を Activate すると図 3 の画面が表示される。
（marker_recognition_test_activate.bat ですべてのＲＴＣをまとめて起動することができる。）

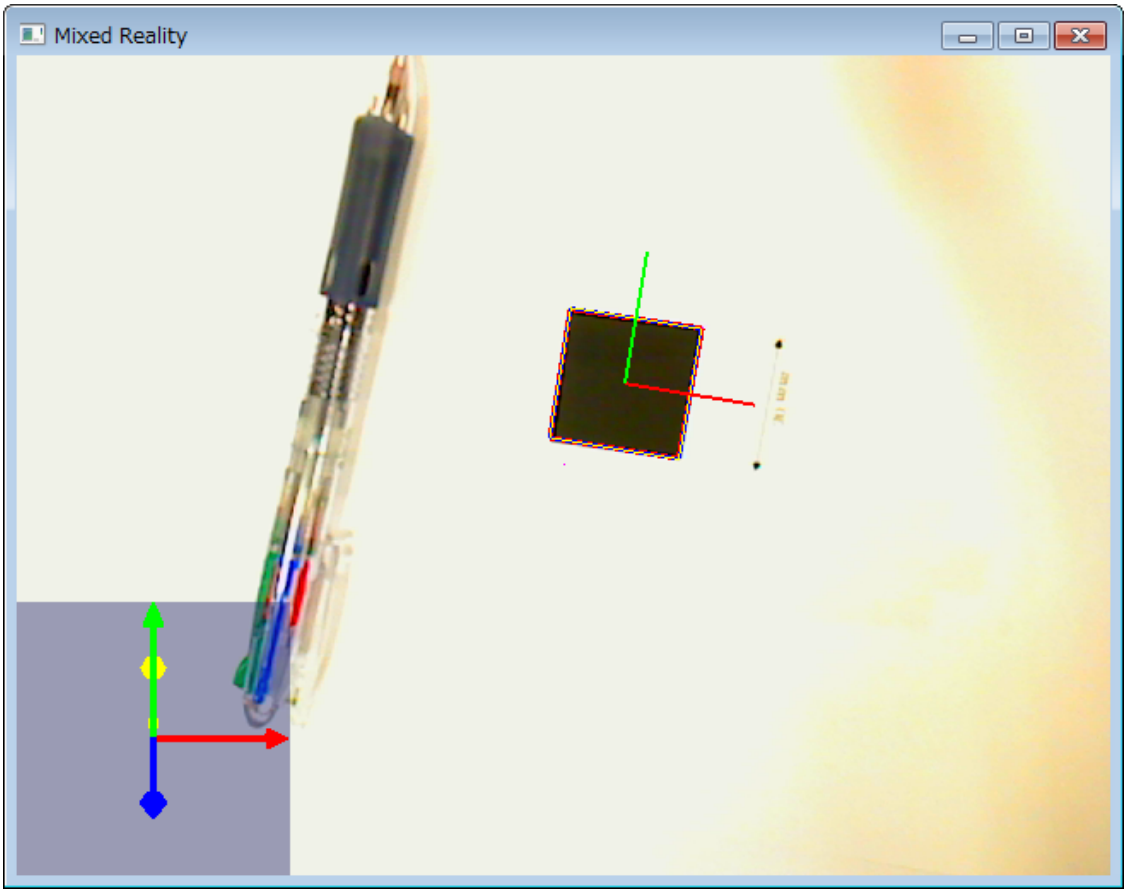

図 3　MarkerRecognition RTC

MarkerRecognition RTC には入力された画像に、マーカを認識した結果を重畳表示する機能がある。また、画面左下に finger_position ポートに入力されたロボットの手先姿勢の方向を表示する。

５．４．　ObjectPositionTest RTC

ObjectPositionTest RTC を Activate すると図４の画面が表示される。 ObjectPositionTest RTC は MarkerRecognition RTC から出力されたマーカ認識結果を受信し、コンソールに表示する。

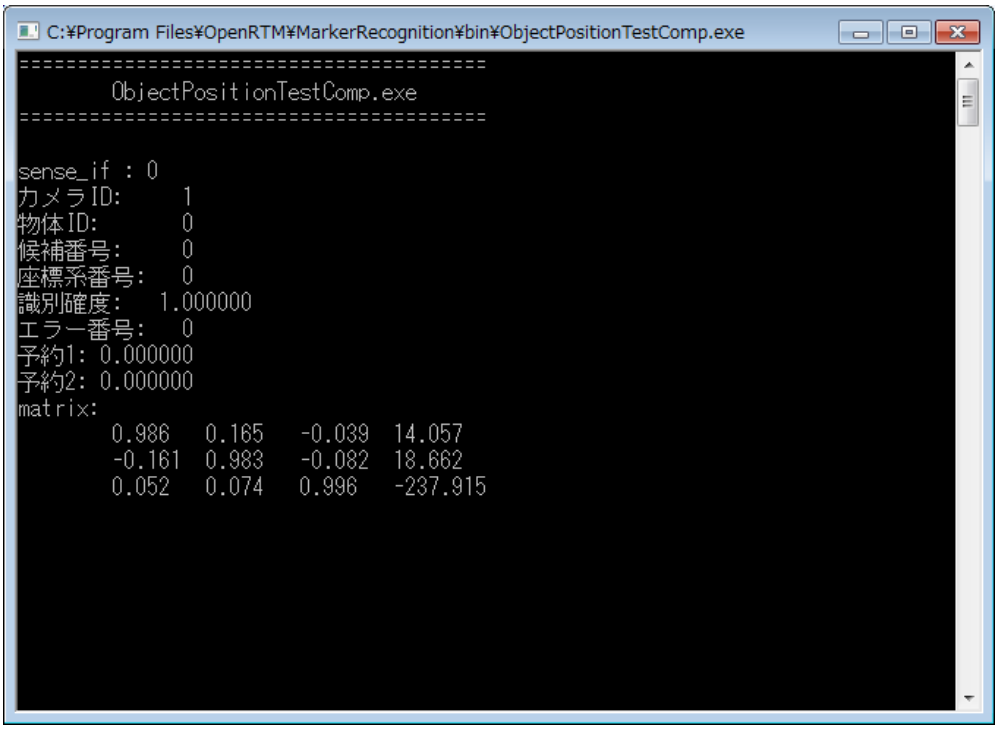

図４　ObjectPositionTest RTC

５．５．　RecognitionServiceTest RTC

RecognitionServiceTest RTC を Activate すると図５の画面が表示され、コンソールからの入力待ち状態になる。MarkerRecognition RTC は RecognitionService ポートを通じて認識対象のマーカ形状を設定・取得することができる。

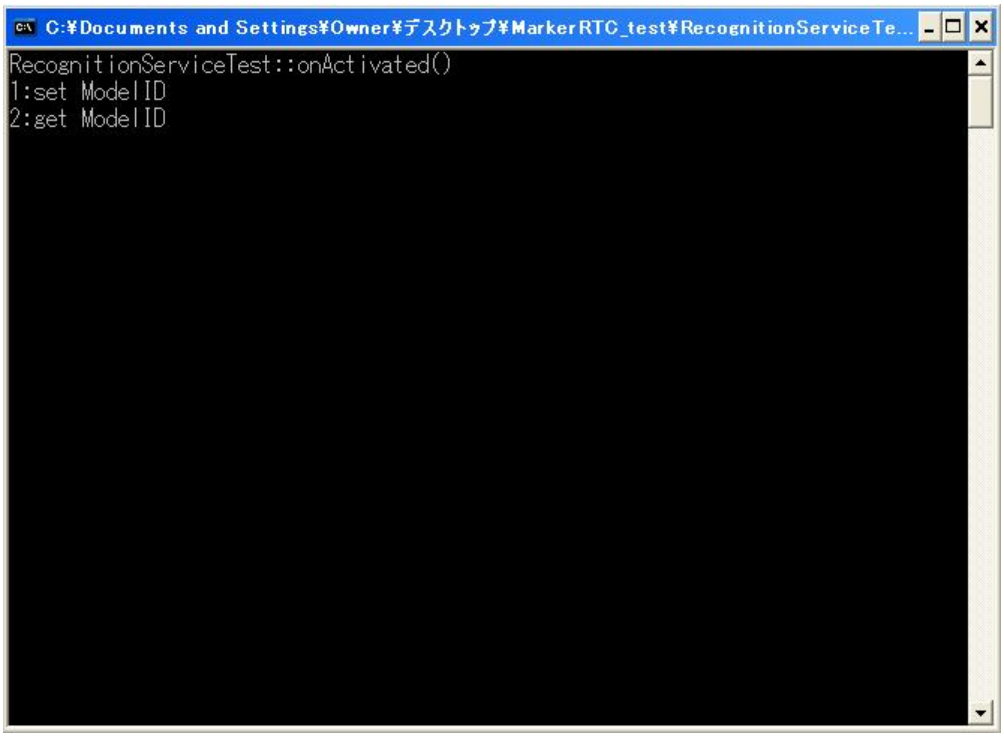

図５　RecognitionServiceTest (入力待ち状態)

５．６．　FingerPositionTest RTC

FingerPositionTest RTC を Activate すると図 6 の画面が表示される。

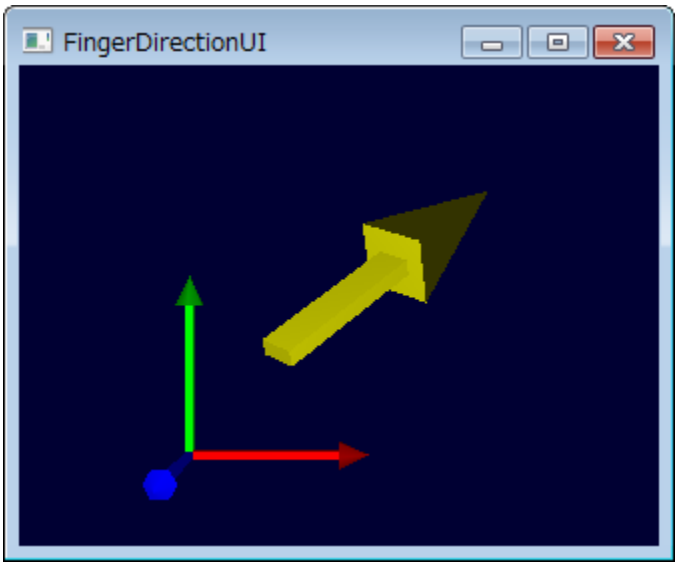

図 6　FingerPositionTest RTC

FingerPositionTest RTC は相対位置決め制御モジュールからの出力をエミュレートするための RTC で、画面上に表示されている黄色の矢印の姿勢をロボットの手先姿勢として finger_position ポートから出力する。画面上には黄色の矢印と、赤、緑、青のグローバル座標系を表す矢印が表示される。赤、緑、青の矢印はそれぞれ OpenGL の座標系である X 軸、Y 軸、Z 軸の方向を表している。

画面上に表示されている黄色の矢印はマウスのドラッグ操作で回転することができる。 C キーを押すと黄色の矢印の回転方向をリセットされる。

５．７．　接続

RTSystemEditor を起動する。RTSystemEditor 上で NameServiceView にネームサーバが起動していることを確認する。

marker_recognition_test_activate.bat で各 RTC を起動させ、図 1 のような接続構成で RTC を接続する。

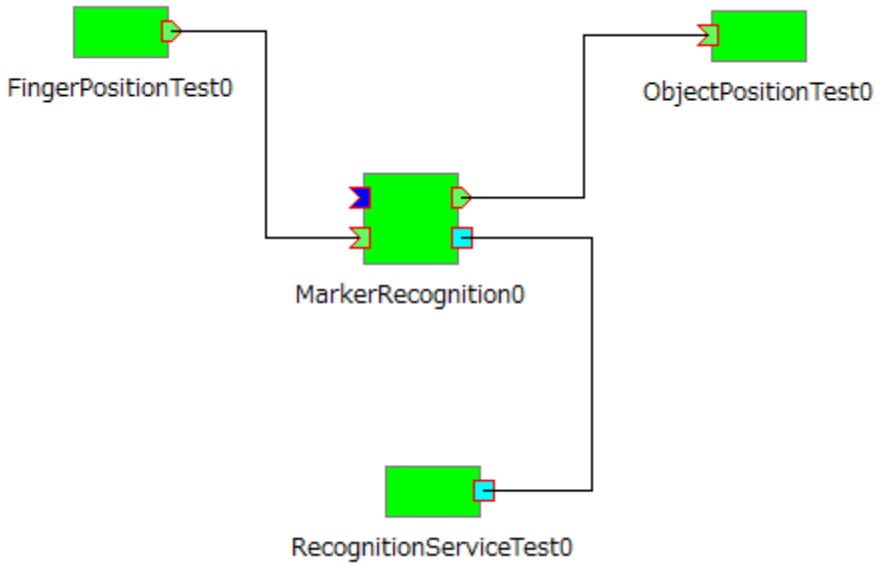

図 8　RTC の接続構成

５．８．　活性化

接続が完了したら、RTSystemEditor 上で RTC を選択し、サブメニューから”activate”を選択することで活性化する。

５．９．　マーカ形状の選択

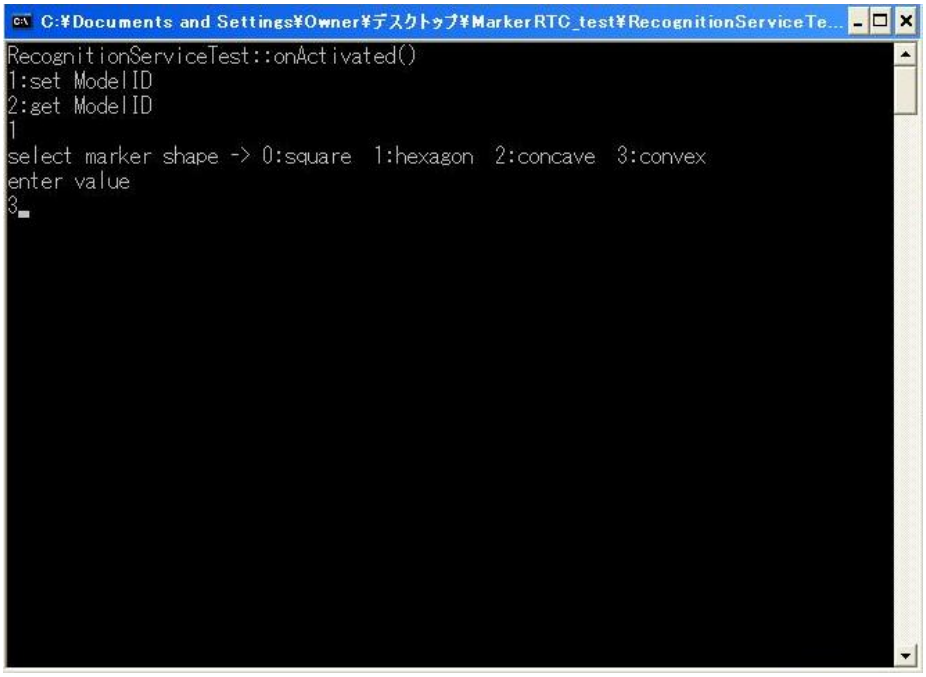

図 9　マーカ形状選択画面

RecognitionServiceTest RTC の、入力待ち状態で 1 を入力し、"1:set Model ID"を選択した場合、図 9 のように認識対象とするマーカの種類を入力することができる。0～3 の値を入力すると RecognitionService ポートを通じて MarkerRecognition RTC に認識対象のマーカを設定することができる。

０：四角形

１：六角形

２：凹形

３：凸型

（各マーカの寸法値は、添付の別ファイルを参照）

図 10～13 に、四角形、六角形、凹形、凸型のマーカの認識結果を示す。

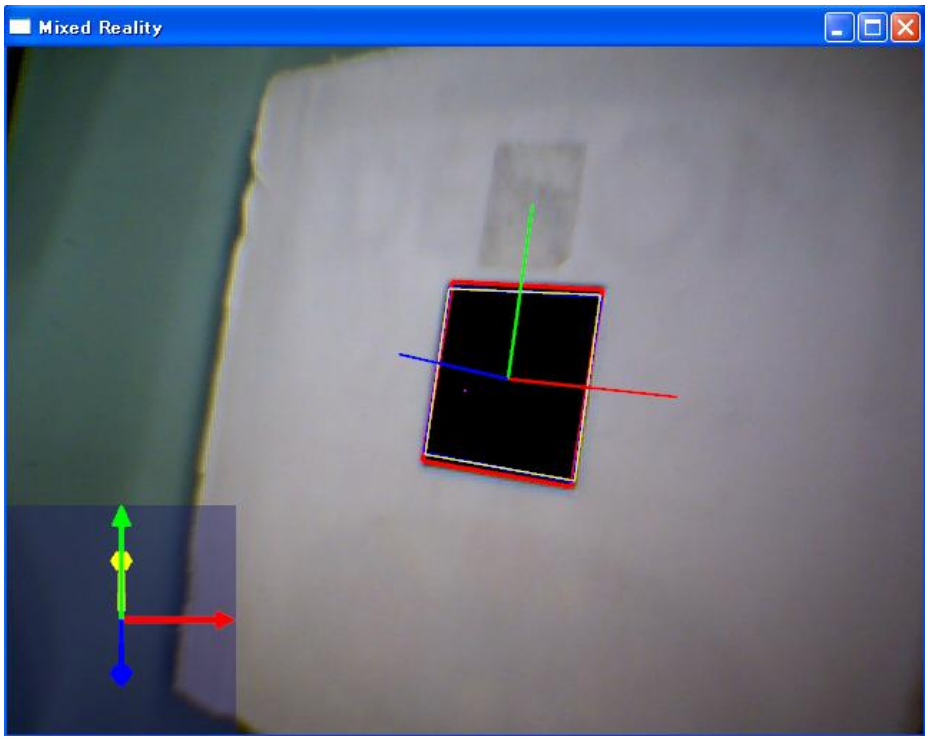

図 10　四角形マーカ

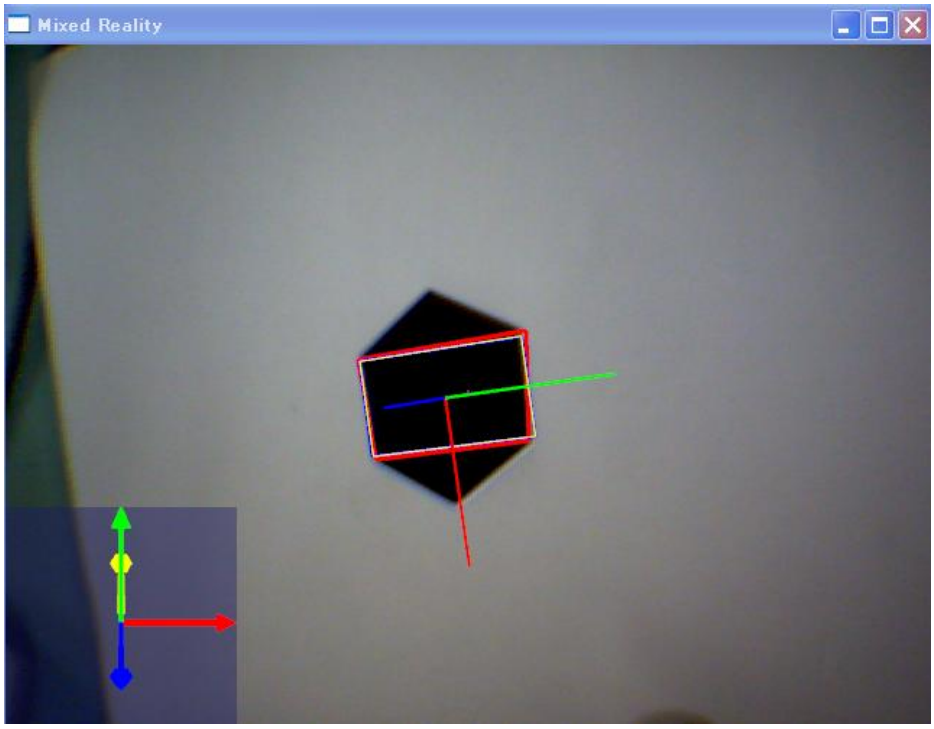

図 11　六角形マーカ

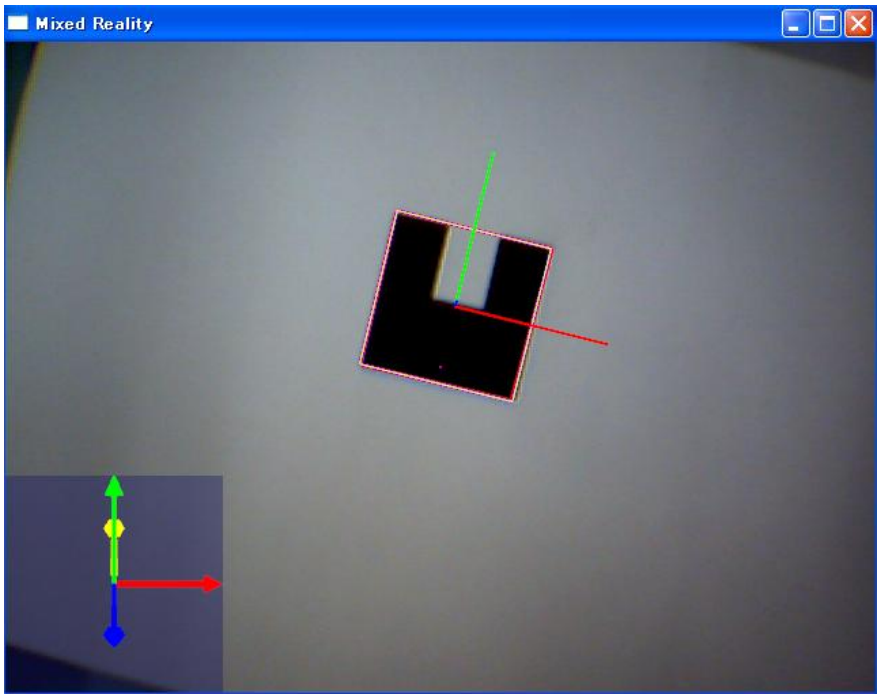

図 12　凹形マーカ

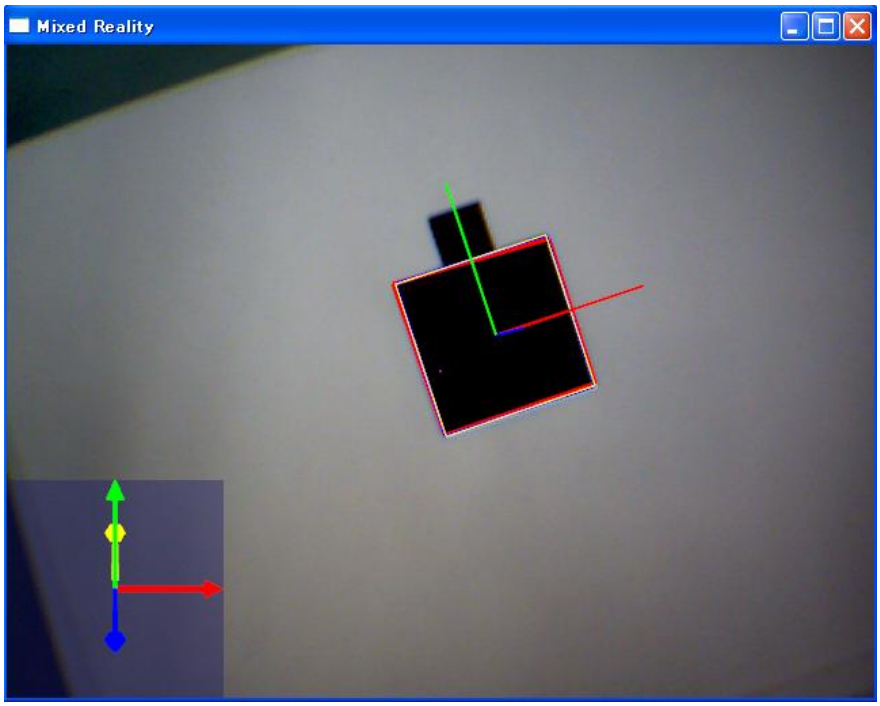

図 13　凸形マーカ

５．１０．　ロボット手先位置座標系の回転

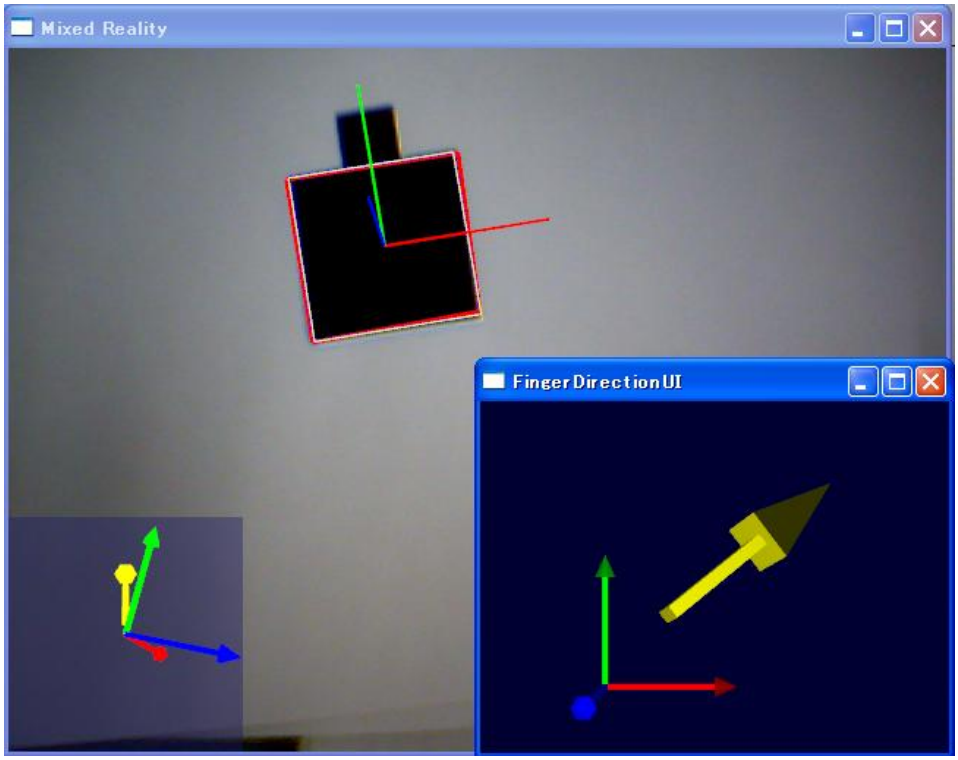

図 14　ロボット手先位置位置座標系の回転方法

前述のように、FingerPositionTest RTC は相対位置決め制御モジュールからの出力をエミュレートするための RTC で、画面上に表示されている黄色の矢印の姿勢をロボットの手先姿勢として finger_position ポートから出力する。画面上には黄色の矢印と、赤、緑、青のグローバル座標系を表す矢印が表示される。赤、緑、青の矢印はそれぞれ OpenGL の座標系である X 軸、Y 軸、Z 軸の方向を表している。画面上に表示されている黄色の矢印はマウスのドラッグ操作で回転することができる。 C キーを押すと黄色の矢印の回転方向をリセットされる。

５．１１．　結果表示

MarkerRecognition RTC には、付随する Window 画面が幾つかあり、その中で「Square Detection Demo」と書かれた Window 画面には、マーカの認識状況とマーカ中心の位置・姿勢が表示される。前述の ObjectPositionTest RTC では、センス系統一ＩＦの仕様に基づいた出力が表示されるが、こちらでは、姿勢について、Roll、Pitch、Yaw に直された結果が表示される。

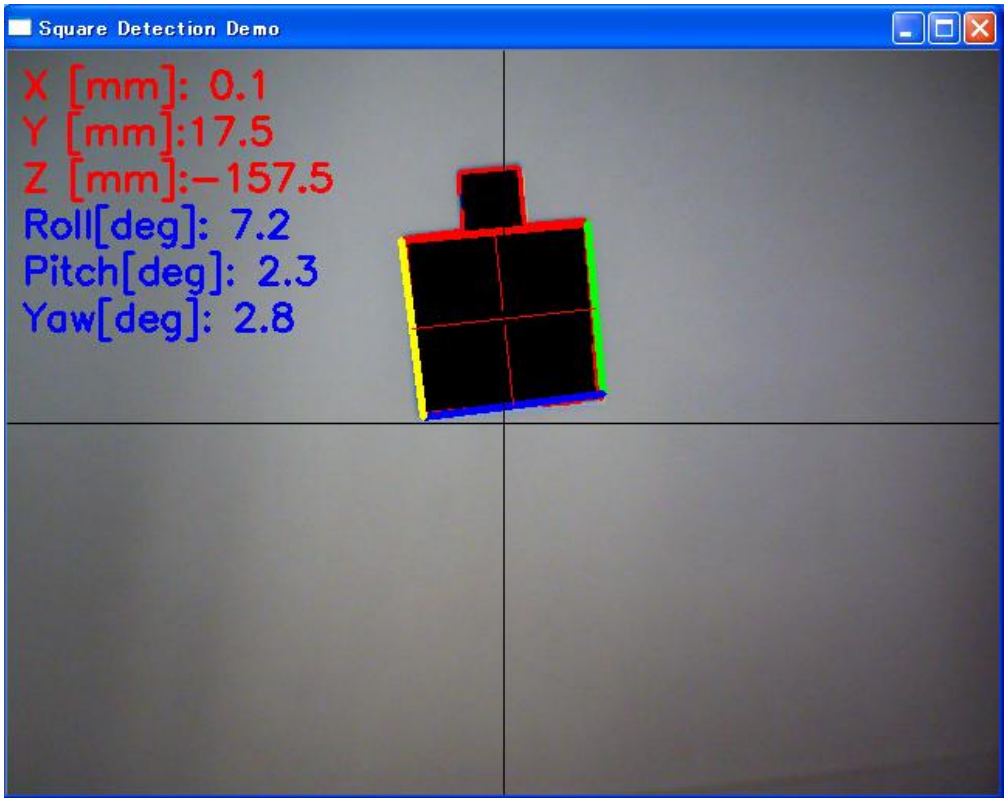

図 15　位置と姿勢角の表示

５．１２．　ＲＴＣの終了手順

RTSystemEditor 上で RTC を選択し、サブメニューから”deactivate”を選択することで終了する。

６．　特記事項

本モジュールは，別添の「エンドユーザー使用許諾契約書」の内容に、ご同意頂いた場合に限りご使用になれます。

※Windows® は、Microsoft Corporation の日本およびその他の国における商標または登録商標です。