

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

ステレオ楕円画像認識モジュール

V e r . 2.2

2012 年 1 月 24 日

(株) 東芝

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
1. 4. 動作環境	1
2. ディレクトリ構成	2
3. ソフトウェアインストール	3
3. 1. 基本環境	3
3. 1. 1. OpenRTM-aist-1.0.0 RELEASE (C++)	3
3. 1. 2. OpenCV	3
3. 1. 3. Simple DirectMedia Layer(SDL)	3
3. 1. 4. 1394-based DC Control Library	3
3. 2. 実機環境	5
4. 事前準備	6
4. 1. IEEE1394 カメラのドライバインストール	6
4. 2. カメラのセッティング	6
5. 実行	7
5. 1. ネームサーバの起動	7
5. 2. RTSystemEditorの起動	7
5. 3. カメラ撮影	8
5. 3. 1. カメラの起動	8
5. 3. 2. RTCの起動	8
5. 3. 3. 接続	8
5. 3. 4. 活性化	8
5. 3. 5. R T Cの終了手順	8
6. 特記事項	9

1. はじめに

1. 1. 本書の適用範囲

本書は、ステレオ楕円画像認識モジュールの操作手順について記述した文書である。ステレオ楕円画像認識モジュールは、ステレオカメラで複数の物体（皿やコップ）の円弧（楕円）を検出し、予め登録した半径データと照合して、その位置・姿勢を算出するものである。

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	ステレオ楕円画像認識モジュール 機能仕様書	ステレオ楕円画像認識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 : 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	Linux(Ubuntu Linux 10.04 にて動作を確認)
開発言語	C++
コンパイラ	GCC 4.3.3
RT ミドルウェア／バージョン	OpenRTM-aist 1.0.0
依存パッケージ	OpenCV(http://sourceforge.net/projects/opencvlibrary/) Simple DirectMedia Layer(SDL) (http://www.libsdl.org/) 1394-based DC Control Library (http://sourceforge.net/projects/libdc1394/)

2. ディレクトリ構成

本書では下表のディレクトリ構成のもと話をする。ディレクトリ構成が下表と異なる場合は、適宜その環境に合わせた修正が必要になる。

ディレクトリ	言語	内容	備考
/home/user			
└StereoEllipse	C++	楕円認識コンポーネント関連ディレクトリ	
/usr/lib	C++	OpenC V 関連ライブラリ	
/usr/include/opencv	C++	OpenC V 関連インクルードファイル	
ト			

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 (1.0.0)、OpenCV、Simple DirectMedia Layer(SDL)、

1394-based DC Control Library は上記のディレクトリ構成内にインストールすること。

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

3. 1. 1. OpenRTM-aist-1.0.0 RELEASE (C++)

コマンドラインから次のように入力し、OpenRTM 1.0.0 のインストールを行う。

```
$wget
http://www.openrtm.org/OpenRTM-aist/download/install_scripts/pkg_install_ubuntu
.sh
$ chmod +x pkg_install_ubuntu.sh
$ sudo ./pkg_install_ubuntu.sh
※途中で数回インストールするかどうか聞かれるので、"y"を入力してインストールを進める。
```

3. 1. 2. OpenCV

apt-get コマンドを実行して OpenCV のインストールを行う。

```
$ sudo apt-get install libcv4 libcv-dev libcvaux4 libcvaux-dev libhighgui4
libhighgui-dev
```

3. 1. 3. Simple DirectMedia Layer(SDL)

apt-get コマンドを実行して SDL のインストールを行う。

```
$ sudo apt-get install libsdl1.2debian-pulseaudio libsdl1.2-dev
```

3. 1. 4. 1394-based DC Control Library

1394 カメラから画像をキャプチャする際に使用する libdc1394 のインストールを行う。
Ellipse_BoK の pvr-linux.{h|cpp}は古い 1.x 系の libdc1394 を必要とするので、ソースコードを取得しコンパイルを行う。

はじめに libdc1394 のコンパイルに必要なライブラリを apt-get でインストールする。

```
$ sudo apt-get build-dep libdc1394-22
```

次に、libdc1394 が使用する libraw1394 を configure & make する。 libraw1394 は

Ubuntu のパッケージにも用意されているが、libdc1394 は 1.x 系の libraw1394 を想定しているため Ubuntu に用意されているパッケージを使用せず、ソースコードからコンパイルする必要がある。

```
$ wget
http://jaist.dl.sourceforge.net/project/libraw1394/libraw1394/1.2.0/libraw1394-1.2.0.tar.gz
$ tar xvfz libraw1394-1.2.0.tar.gz
$ cd libraw1394-1.2.0
$ ./configure
$ make
$ sudo make install
$ sudo make dev
$ sudo chown 0666 /dev/raw1394
$ cd ..
※/usr/local/以下にインストールされる
```

最後に libdc1394 のソースコードを取得し、configure & make を行う。

```
$ wget
http://jaist.dl.sourceforge.net/project/libdc1394/libdc1394/1.2.2/libdc1394-1.2.2.tar.gz
$ tar xvfz libdc1394-1.2.2.tar.gz
$ cd libdc1394-1.2.2
$ CFLAGS="-I/usr/local/include" LDFLAGS="-L/usr/local/lib" ./configure
$ make
$ sudo make install
```

ライブラリファイルのインストールを行った後は、以下の手順で ldconfig を設定する必要がある。

```
$ sudo sh -c 'echo "/usr/local/lib" >> /etc/ld.so.conf'
$ sudo /sbin/ldconfig
```

3. 2. 実機環境

本知能モジュールは、ステレオカメラとして、現在Point Grey Research社※のFlea及びFlea2を対象にしている。前述のようにIEEE1394 ケーブルを介してP Cに接続する。

※U R L

<http://www.viewplus.co.jp/product/01/01.html>

図 1 に本R T Cのハードウェア構成図を示す。

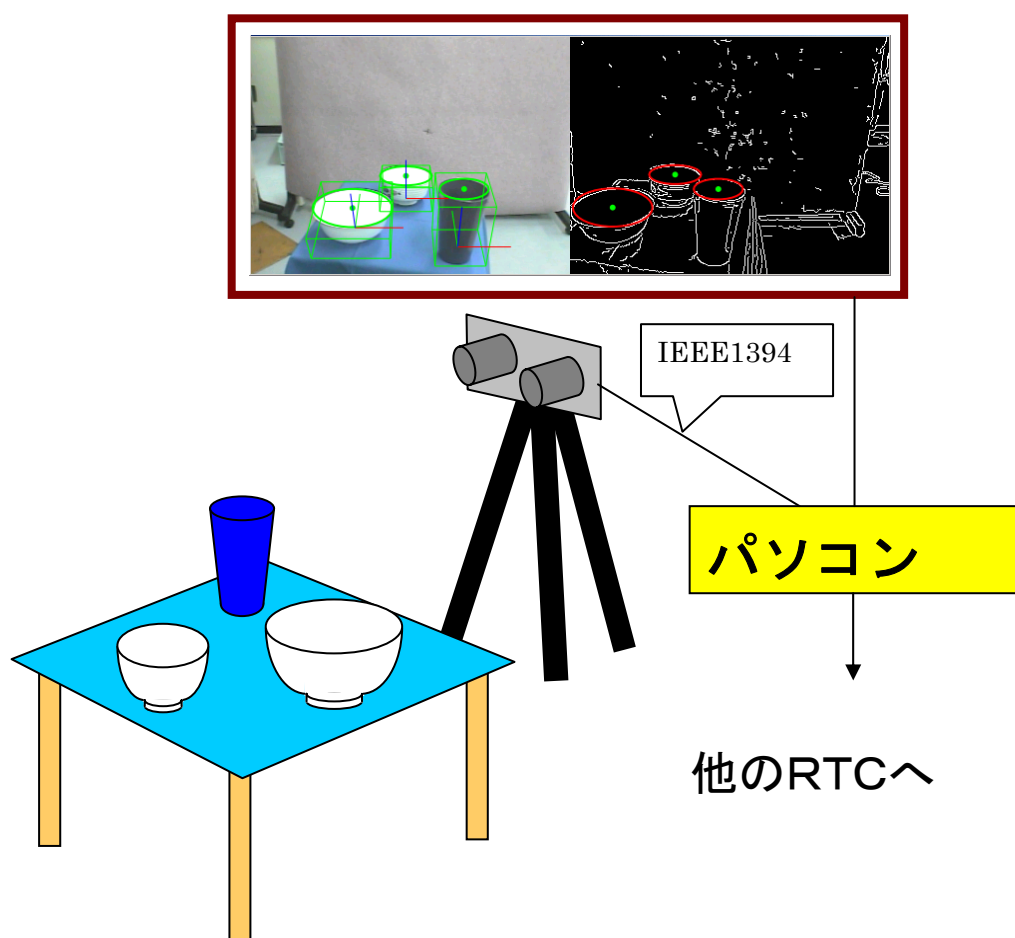


図 1 本R T Cのハードウェア構成図

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. IEEE1394 カメラのドライバインストール

Point Grey Research 社、Flea 及び Flea2 に添付された CD-ROM 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

二つのカメラを雲台にのせ、三脚などで固定し、対象物が二つのカメラから撮影できることを確認する。

5. 実行

5. 1. ネームサーバの起動

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
```

```
4e616d696e67436f6e746578744578743a312e300000010000000000000070000000010102000e00
```

```
00003133332e3139362e38392e383600f90a0b0000004e616d655365727669636500030000000000
```

```
0000080000000100000000545441010000001c000000010000000100010001000000010001050901
```

```
01000100000009010100035454410800000059699d4d01000c2c
```

```
Checkpointing Phase 1: Prepare.
```

```
Checkpointing Phase 2: Commit.
```

```
Checkpointing completed.
```

5. 2. RTSystemEditor の起動

Eclipse を立ち上げその上で 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。

5. 3. カメラ撮影

5. 3. 1. カメラの起動

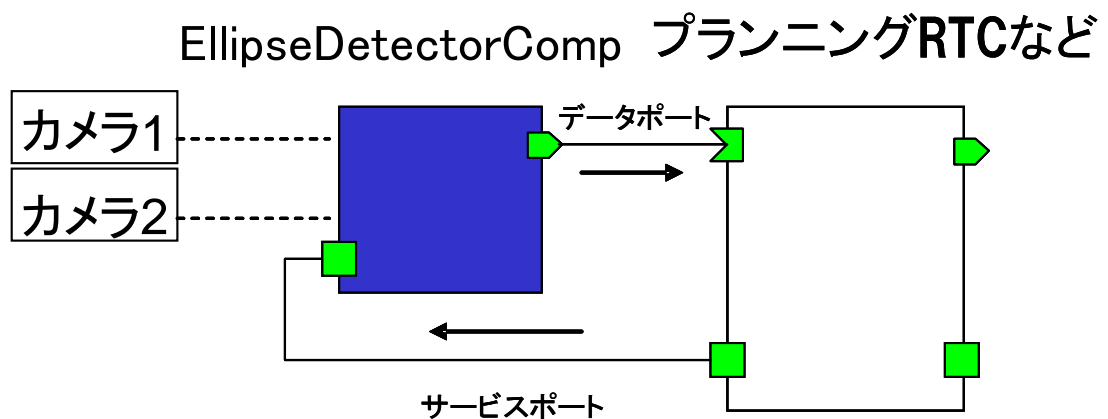
カメラのスイッチをいれる。

5. 3. 2. RTC の起動

EllipseDetectorComp、プランニング RTC 等を起動。

5. 3. 3. 接続

RTSystemEditor を用いて各 RTC を接続する。下図を参考にコンポーネントのポートを接続する。ポートの詳細は機能仕様書を、参照のこと。



5. 3. 4. 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. 3. 5. 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。

※Linux®は、Linus Torvalds 氏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。

※Ubuntu®は、Canonical Ltd. 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。