

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

部分エッジ画像認識モジュール

V e r . 2 . 2

2012年1月23日

(株) 東芝

改版履歷

[illegible]

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
4. 3. 画像の登録	8
5. 実行	10
5. 1. ネームサーバの起動	10
5. 2. RTSystemEditorの起動	10
5. 3. カメラ撮影	11
5. 3. 1. カメラの起動	11
5. 3. 2. RTCの起動	11
5. 3. 3. 接続	11
5. 3. 4. 活性化	12
5. 3. 5. 画像の認識、位置・姿勢出力	12
5. 3. 6. RTCの終了手順	13
6. 特記事項	14

1. はじめに

1. 1. 本書の適用範囲

本書は、部分エッジ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。部分エッジ画像認識モジュールは、カメラで複数の物体（スプーン、ナイフなど）の部分的な曲線を検出し、予め登録した曲線データと照合し、その物体の重心位置・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	部分エッジ画像認識モジュール 機能仕様書	部分エッジ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	Windows (Xp,sp3 で動作確認済み)
開発言語	C++
コンパイラ	VisualC++2008
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/)

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 PartialEdge 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			
└ OpenC V 2.2	C++	OpenC V 2.2 関連ファイル	
└ OpenRTM	C++	RTM 本体	
└ 1.0		Ver1.0.0	
└ examples			
└ C++			
└ PartialEdgeConsumer	C++	部分エッジテスト用	
└ PartialEdge	C++	部分エッジ関連ファイル	
└ Train	C++	画像データベース	

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)、OpenCV2.2、

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

あらかじめ、部分エッジ画像認識モジュールを実行する 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/>

3. 2. 実機環境

本知能モジュールは、市販のU S Bカメラ(画素数 640x480)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。

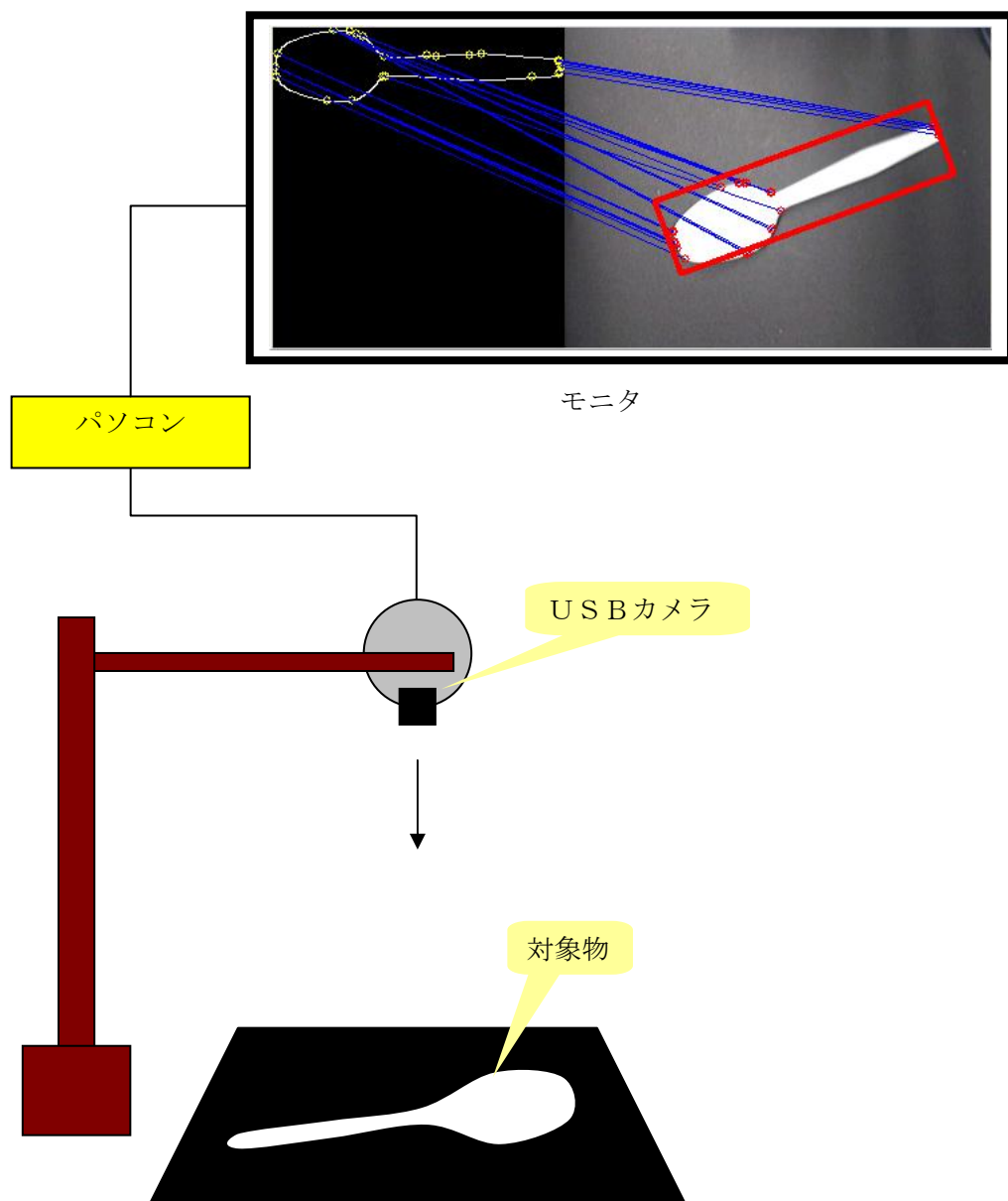


図 1 本R T Cのハードウェア構成図

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. 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。

4. 3. 画像の登録

PartialEdgeComp.exe を立ち上げる。PartialEdgeComp のコンソールパネルで、図 2 のように Do you make model DB? (y/n)

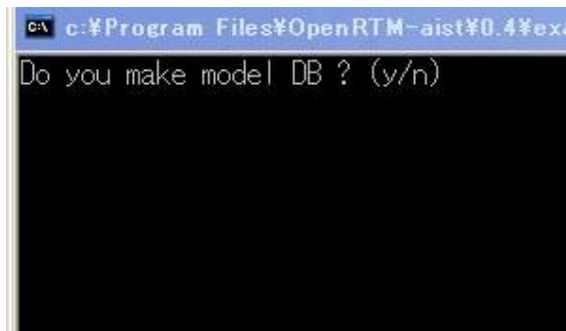


図 2 PartialEdgeComp のコンソールパネル

と問われるので「y」と入力し、Enter キーを押すと図 3 のウインドウ表示が現れる。

黒い紙など影の映らない背景を用意して、対象物を置き、カメラ画像の中央に対象物が映るようにする。「s」キーを入力すると物体が登録され、重心座標や方向、部分エッジなどが保存される。

図 3 の「frame」と書かれたウインドウに原画像が表示され、「canny」と書かれたウインドウにエッジ抽出結果が表示される。このウインドウには、閾値調整用のトラックバーがあるので、適宜、ノイズが出ないように調整する。「counters」ウインドウには、そのエッジ抽出結果が表示される。「s」キー入力後に「counters2」が表示され、対象物を囲むバウンダリーボックスと重心位置、および、物体方向が表示される。マウスで白いバウンダリーボックスの角を指定すると、バウンダリーボックス位置が変更されるとともにその中心である重心も変更される。従って、任意の位置に重心を登録することができる。「counters2」のウインドウを移動するときに、画面内をクリックすると、重心位置が変わってしまうので、注意が必要である。「s」キーを押すたびに画像や重心情報は、画像データベースに上書きされ

る。画像登録が終わったら、エスケープキー「ESC」を押し、画像認識に移ることができる（事前に **Naming** サーバを立ち上げておく必要がある）。

なお、登録に使ったエッジ画像や重心位置情報は、「**Train**」と書かれたディレクトリに保存される。登録が終わったら、一旦、このコンソールを閉じる。

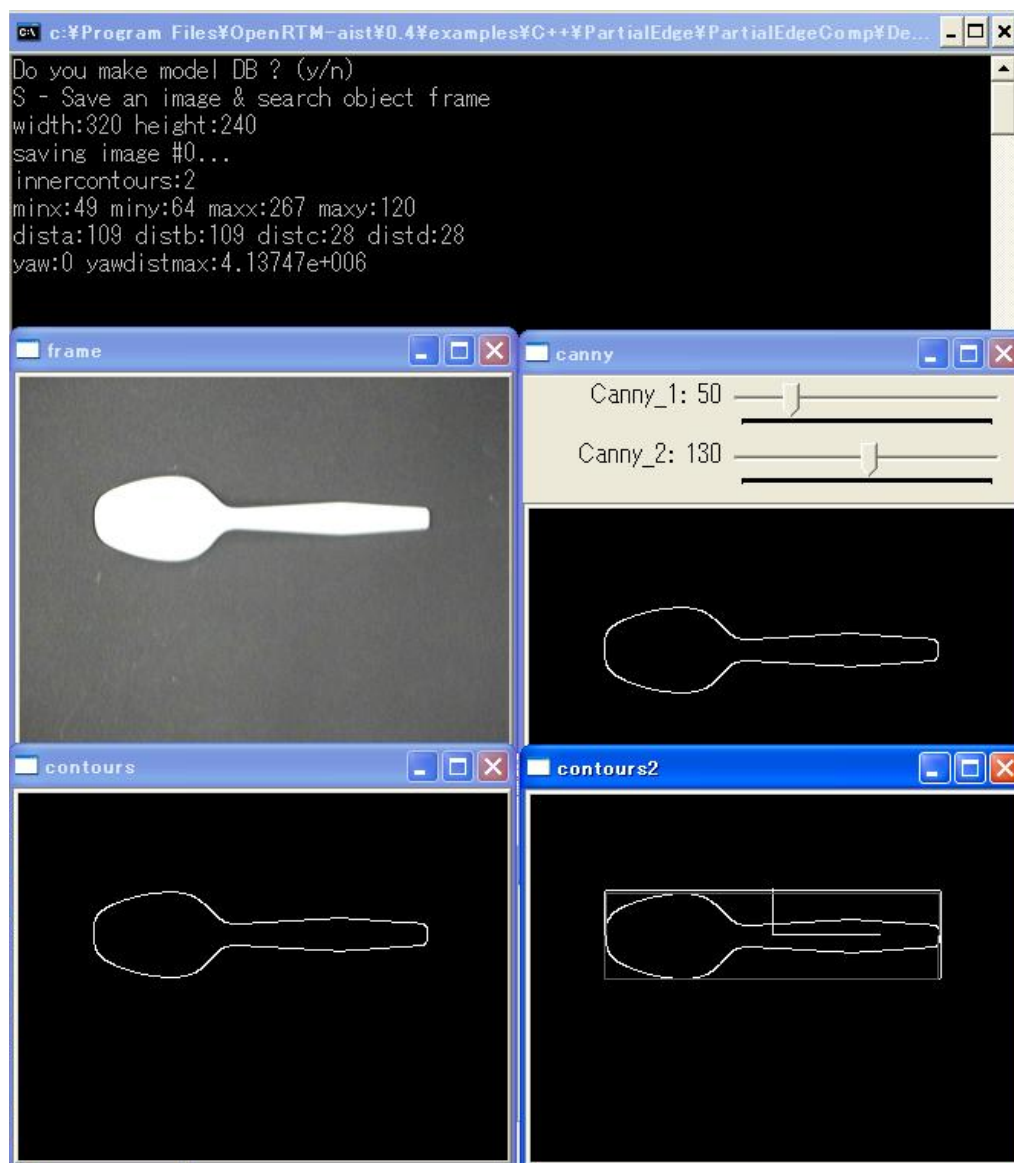


図 3 対象物画像の登録画面

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
00000800000001000000000545441010000001c000000010000000100010001000000010001050901
0100010000000090101000354544108000000059699d4d01000c2c
Checkpointing Phase 1: Prepare.
Checkpointing Phase 2: Commit.
Checkpointing completed.
```

図 4 ネームサーバの表示

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

カメラのスイッチがある場合はONする。

5. 3. 2. RTC の起動

PartialEdgeComp、PartialEdgeConsumerComp を起動。

PartialEdgeComp のコンソールで「n」と入力すると RTC がスタートする(図 5)。

PartialEdgeConsumerComp はそのままスタートする。

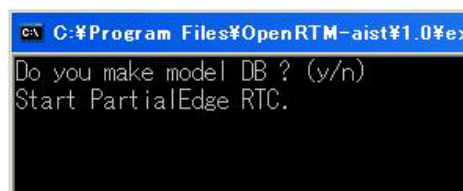


図 5 PartialEdgeComp のコンソールで「n」と入力

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、

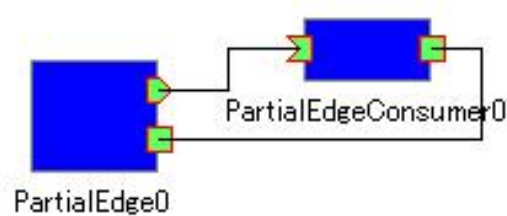


図 6 R T C の接続

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. 画像の認識、位置・姿勢出力

図 5 で Do you make model DB? (y/n)と問われたときに、「n」と入力すると既に登録された画像が呼び出され、その画像とのマッチングが行われる（図 7）。

一方、PartialEdgeConsumerComp のコンソールパネルには、図 8 の表示がされている。図 8 の 1 番と 2 番は、サービスポートを通じて登録画像番号を入力するものであるが、今回は、画像を一種類しか登録できないので、番号をセットできるが、登録画像の変更はない。3 番を入力すると、機能仕様書に記載した統一 I F 仕様で、物体の位置姿勢情報が 20 個の変数値で表示される（図 9）。これは、PartialEdgeComp の結果をデータポートで受け取ったものである。今回の実装では、部分隠れに対応するために False Positive（誤認識）の判定を厳格にしていない。これは、局所特徴量の合致比率等によって制御できる。

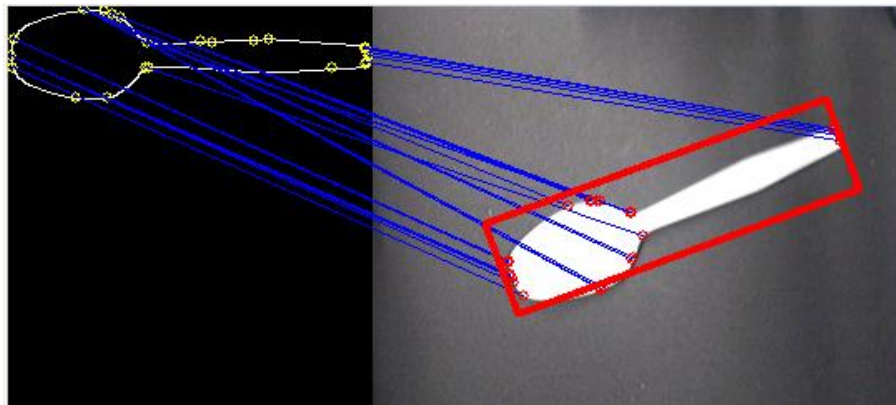


図 7 画像認識結果

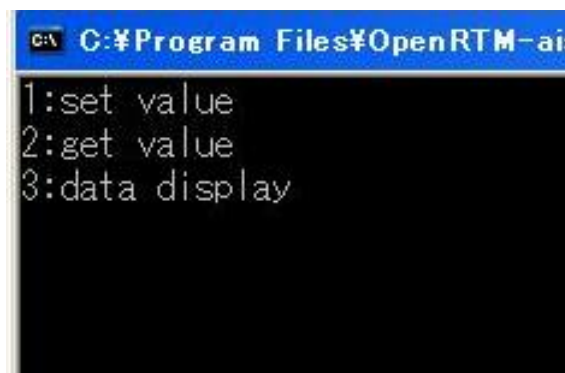
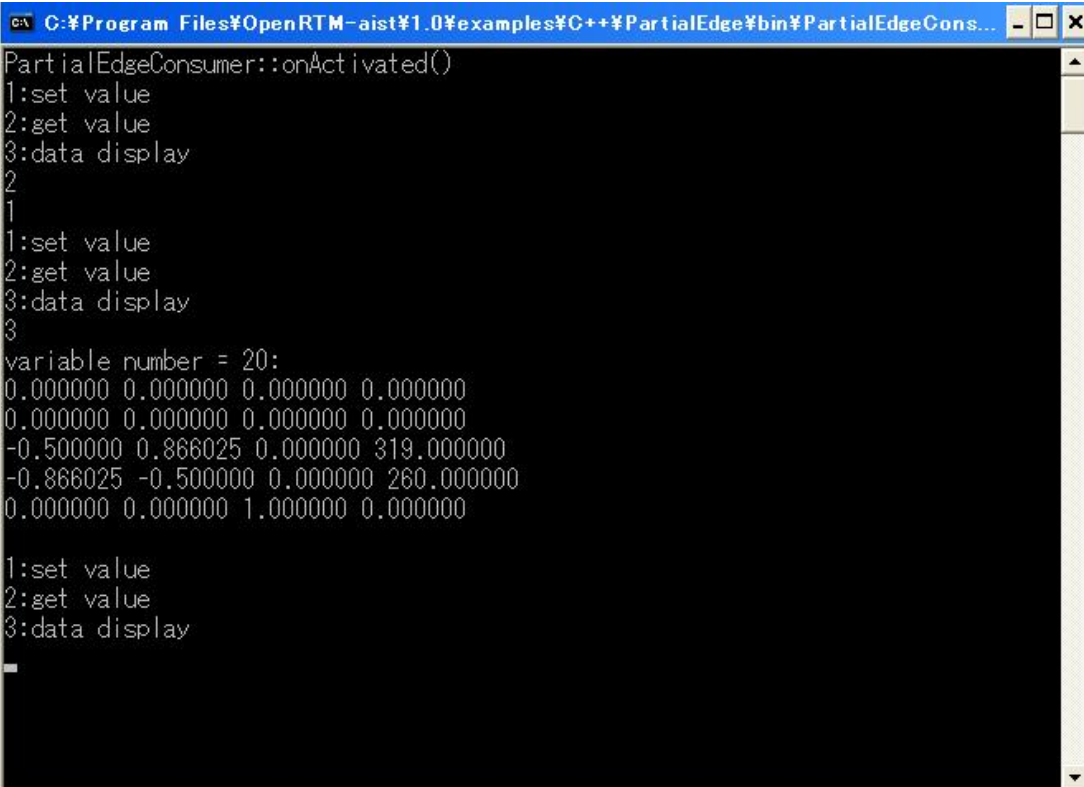


図 8 PartialEdgeConsumer の表示



```

C:\Program Files\OpenRTM-aist\1.0\examples\C++\PartialEdge\bin\PartialEdgeCons...
PartialEdgeConsumer::onActivated()
1:set value
2:get value
3:data display
2
1
1:set value
2:get value
3:data display
3
variable number = 20:
0.000000 0.000000 0.000000 0.000000
0.000000 0.000000 0.000000 0.000000
-0.500000 0.866025 0.000000 319.000000
-0.866025 -0.500000 0.000000 260.000000
0.000000 0.000000 1.000000 0.000000
1:set value
2:get value
3:data display

```

図 9 PartialEdgeConsumerComp での結果出力

5. 3. 6. 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

本モジュールをご利用される場合は、以下に示す記載事項・条件にご同意いただけたものとします。

本モジュールのライセンスは **Eclipse Public License(EPL)**に従います。利用条件の詳細については、下記サイトを参照ください、なお、本モジュールは利用条件に同意した場合にのみ利用可能となっており、本モジュールを利用した時点でライセンス条項に同意したものとみなします。

Eclipse Public License <http://www.eclipse.org/legal/epl-v10.html>