

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 触覚認識モジュール

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. ロボット、ハンド、変位センサの準備	8
4. 2. ロボット単体での動作チェック	8
4. 3. 対象物体のモデル形状データ、指先位置データの準備	8
4. 4. ロボットの動作シナリオを記述	10
5. 実行	11
5. 1. ネームサーバの起動	11
5. 2. RTSystemEditorの起動	11
5. 3. ロボットでのデータ計測、またはファイルからのデータと読み込みと触覚認識	12
5. 3. 1. ロボットの起動	12
5. 3. 2. RTCの起動	12
5. 3. 3. 接続	12
5. 3. 4. 活性化	12
5. 3. 5. RTCの終了手順	17
6. 特記事項	18

1. はじめに

1. 1. 本書の適用範囲

本書は、指先位置モジュールと触覚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。指先位置モジュールは、ロボットを動かしながら、変位センサで物体（缶、スプーンなど）の部分的な形状計測し、指先位置を出力するものである。

触覚認識モジュールは、複数の指先位置データを予め登録した物体形状データと照合し、物体の重心位置・姿勢を算出するものである。本書は、これら二つのモジュールと、データ作成のためのロボット動作について記述したものである。

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	触覚認識モジュール 機能仕様書	触覚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,sp2 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/ 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
C:\ProgramFiles			
└ OpenC V	C++	OpenC V 関連ファイル	
└ OpenRTM	C++	RTM 本体	
└ 1.0.0		Ver1.0.0	
└ examples			
└ C++			
└ TactileRecognition	C++	触覚認識モジュール関連ファイル	
└ FingerPosition	C++	指先位置モジュール関連ファイル	

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、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。

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

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>
- rtspfile
 - <http://www.openrtm.org/pub/OpenRTM-aist/tools/1.0.0/rtspfile-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

本知能モジュールを動作させるためには、直動アームと、ロボットハンド、および変位センサが必要である。詳細については、機能仕様書を参照されたい。図 1 に本RTCのハードウェア構成図を示す。図 1 は、ハードウェア構成の一例であり、これ以外の構成もあり得る。ここでは、直動ステージを用いているが、多関節アームでもかまわない。ハードウェアについては、ユーザの方で適宜構成し、それ専用のシナリオコントローラを用意して、触覚情報を獲得するための動作を行わせるものとする。

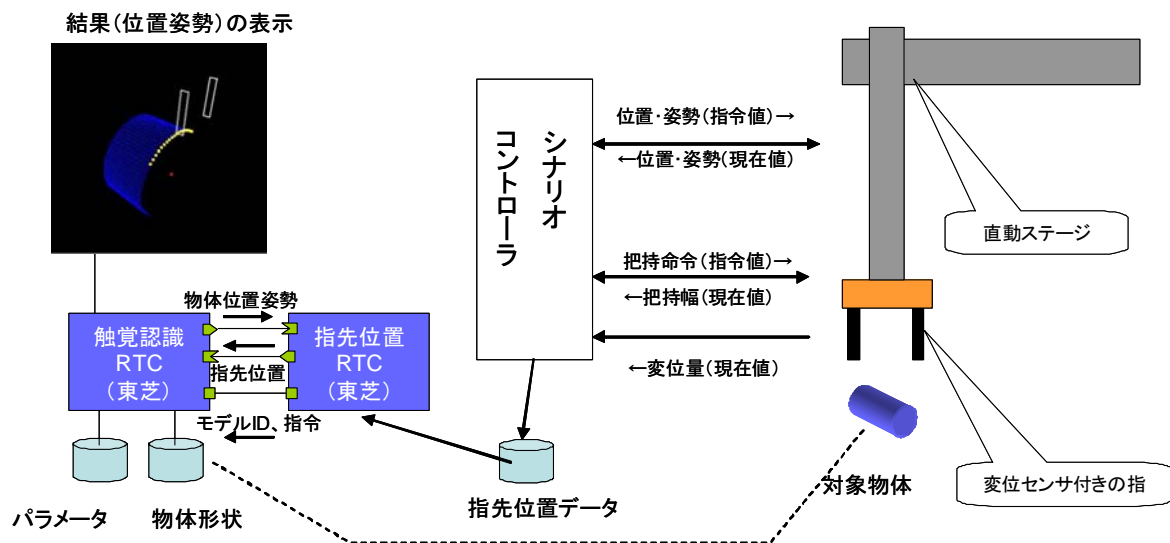


図 1 本RTCのハードウェア構成図

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. ロボット、ハンド、変位センサの準備

詳細は、機能仕様を参照のこと。

4. 2. ロボット単体での動作チェック

ロボット単体の動作をチェックする。操作手順は、ロボットの R T C モジュールの操作手順書などを参照する。

4. 3. 対象物体のモデル形状データ、指先位置データの準備

触覚認識 RTC (TactileRecognitionComp) のための物体形状データを作成する。書式は、「触点番号」、「X座標」、「Y座標」、「Z座標」、「改行」を複数行まとめたもので最後の行に E O F の印として、「-1」を入れる。ファイルは、「tactile_model○.txt」という名称にする。○の部分には、モデル番号を半角数字で入れる。このファイルは、TactileRecognitionComp が読み込むものである。以下に形状データ例を示す。単位は、[mm]である。

```
0,90.000000,100.000000,0.000000
1,91.000000,99.000000,0.000000
2,92.000000,99.000000,0.000000
3,93.000000,99.000000,0.000000
...
-1
```

次に、指先位置 RTC (FingerPositionComp) が扱う指先位置データについて説明する。指先データは、ロボットの手先を動かしながら、そのエンコード値に基づく手先位置と、変化センサの変位量を足し合わせたものである。これについては、動作シナリオなどを用い、ロボット側でデータを加工する。

ロボットを使わない場合は、指先の軌跡をファイルに書いておき、FingerPositionComp に読ませることができる。書式は、「触点番号」、「左指 X 座標」、「左指 Y 座標」、「左指 Z 座標」、「右指 X 座標」、「右指 Y 座標」、「右指 Z 座標」、「改行」を一行とし、複数の触点のデータを複数行ならべ、最後に E O F として「-1」を入れる。ファイルは、「tactile_data○.txt」という名称にする。○の部分にはデータ番号を半角数字で入れる。以下に指先位置データの例を示す。

```
0, 448.110000, -21.775000, 373.0000, 448.110000, 21.775000, 373.0000
1, 446.100000, -22.515000, 373.0000, 446.100000, 22.515000, 373.0000
2, 444.110000, -23.135000, 373.0000, 444.110000, 23.135000, 373.0000
```


3, 442.100000, -23.800000, 373.0000, 442.100000, 23.800000, 373.0000
 ...
 -1

また、「tactile_parameter○.txt」というファイルも用意する。○の部分には、モデル番号を半角数字で入れる。これは、実際に計測した物体形状とモデルデータの縮尺を調整したり、計算機内で、指先軌跡とモデル形状をマッチングさせるときの、モデル初期位置、初期姿勢などを設定したりするものである。以下にその設定例を示す。

```
x_distanse,100.000000
data_rollangle,180.0
data_pitchangle,0.000000
data_yawangle,0.000000
data_scale,1.000000
model_rollangle,0.000000
model_pitchangle,0.000000
model_yawangle,0.000000
model_scale,0.960000
RightF,1
LeftF,1
```

これらは以下の意味を持つ

x_distanse: 実測の指先軌跡の重心と、モデル重心の x 軸上の距離
 data_rollangle: 実測の指先軌跡の、計算機内でのロール角 (degree)
 data_pitchangle: 実測の指先軌跡の、計算機内でのピッチ角 (degree)
 data_yawangle: 実測の指先軌跡の、計算機内でのヨー角 (degree)
 data_scale: 実測の指先軌跡の、計算機内での倍率
 model_rollangle: 形状モデルデータの、計算機内での初期ロール角 (degree)
 model_pitchangle: 形状モデルデータの、計算機内での初期ピッチ角 (degree)
 model_yawangle: 形状モデルデータの、計算機内での初期ヨー角 (degree)
 model_scale: 形状モデルデータの、計算機内での倍率
 RightF: 右指位置データを使用する場合は 1、しない場合は 0
 LeftF: 左指位置データを使用する場合は 1、しない場合は 0

4. 4. ロボットの動作シナリオを記述

指先を動かして、触点データを集めるための動作シナリオを作成する。これは物体の形状を把握するための手先動作である。指の開閉ストロークや変位センサのストロークを考慮し、事前に動作範囲を決め、動作方向、動作ステップなどを決める。これらについては、ロボットのリンク構成に依存するので、ユーザ側で適宜、作製して頂きたい。

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  
00000800000000100000000545441010000001c000000010000000100010001000000010001050901  
010001000000009010100035454410800000059699d4d01000c2c
```

```
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. ロボットでのデータ計測、またはファイルからのデータと読み込みと触覚認識

5. 3. 1. ロボットの起動

ロボットのサーボをONする。

5. 3. 2. RTC の起動

FingerPositionComp、TactileRecognitionComp 等を起動。

5. 3. 3. 接続

RTSystemEditor を用いて各 RTC を接続する。図 2 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、参照されたい。

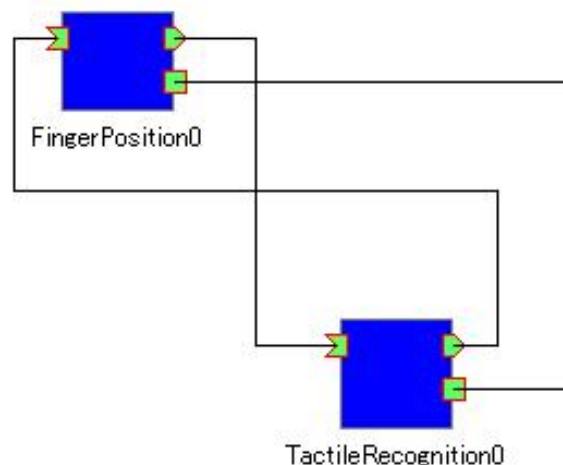


図 2 R T C 接続図

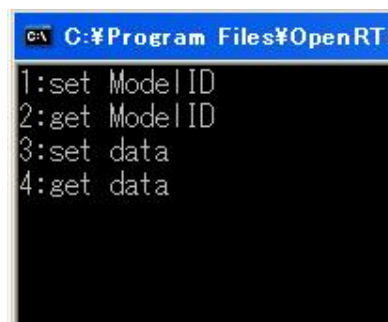
5. 3. 4. 活性化

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。

活性化後、図 3 のような表示が、FingerPositionComp 側のコンソールに現れる。この 1 番と 2 番は、TactileRecognitionComp 側で選ぶ物体形状モデル番号を指定するためのもの

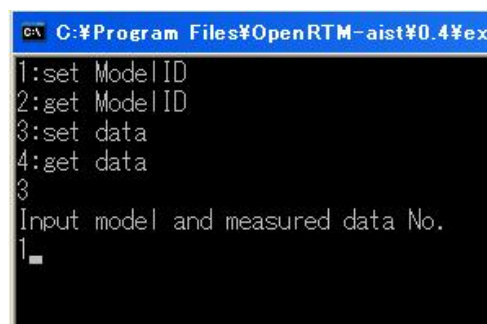
である。これは、今後、モデルが増えたときに使うが、今は特に機能しない。

今回は、簡単のために、リアルタイムでロボットから獲得したデータではなく、予め測定しておいた指先位置データをファイル「tactile_data○.txt」から読み込む場合について説明する。コンソールで3番のset dataを入力すると、図4のようにInput model and measured data Noを問い合わせてくる。これは「tactile_data○.txt」や「tactile_model○.txt」の○の部分、つまり指先位置データや形状モデルの番号を入れるためのものである。現状では、この番号を指先位置データと形状モデルで合わせており、「1」を入力すると「スプーン」のデータ、「2」を入力すると「飲料缶」のデータを読み込むようになっている。これらのデータは、各コンポーネントに同梱してある。



```
C:\Program Files\OpenRTM
1:set ModelID
2:get ModelID
3:set data
4:get data
```

図3 FingerPositionComp の初期表示



```
C:\Program Files\OpenRTM-aist\0.4\example
1:set ModelID
2:get ModelID
3:set data
4:get data
3
Input model and measured data No.
1
```

図4 FingerPositionComp のデータ番号入力

一方、TactileRecognitionComp 側では、図5の表示が現れる。1番から5番までは、前章で説明したパラメータを変更するためのものである。また、6番は、パラメータを保存するためのものである。そして、7番を入力するとマッチング（触覚認識）が行われる。図6～図9が「スプーン」の例で、図6は、マッチング前、図7～9がマッチング後である。黄色

い点線が、指先位置の軌跡を表し、青い点がモデル形状データを表す。なお、白い直方体のワイヤフレームは指先のイメージで、変化はしない。

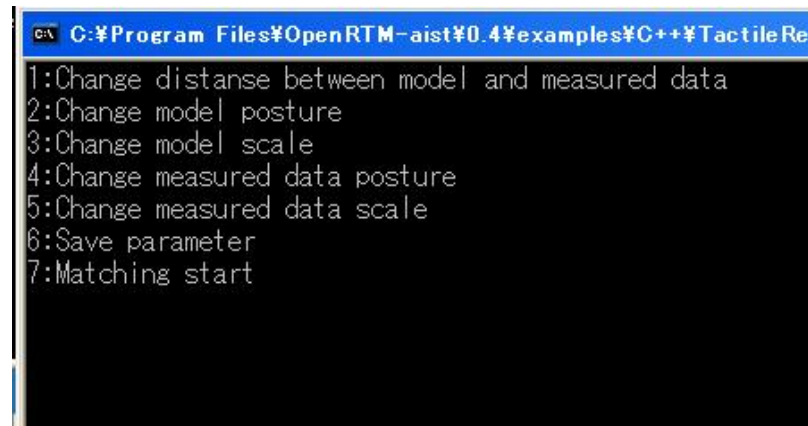


図 5 TactileRecognitionComp 側のコンソール初期画面

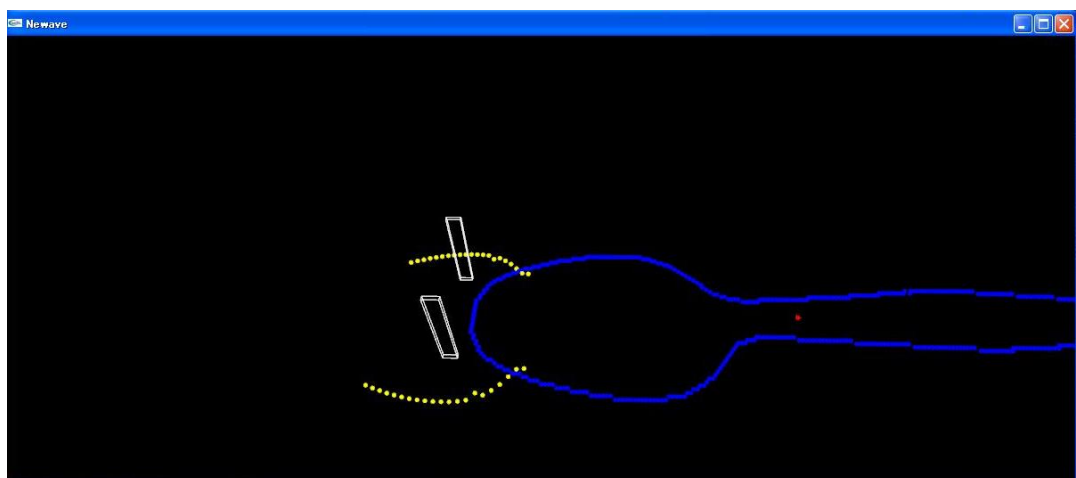


図 6 マッチング前の指先軌跡点（黄）、とモデルデータ形状（青）。スプーンの例。

マッチング状態を表すこれらの画面は、画面をマウスで左ドラッグすることにより、視点を変えることができ、図 7～9 は、視点をそれぞれ変えている。

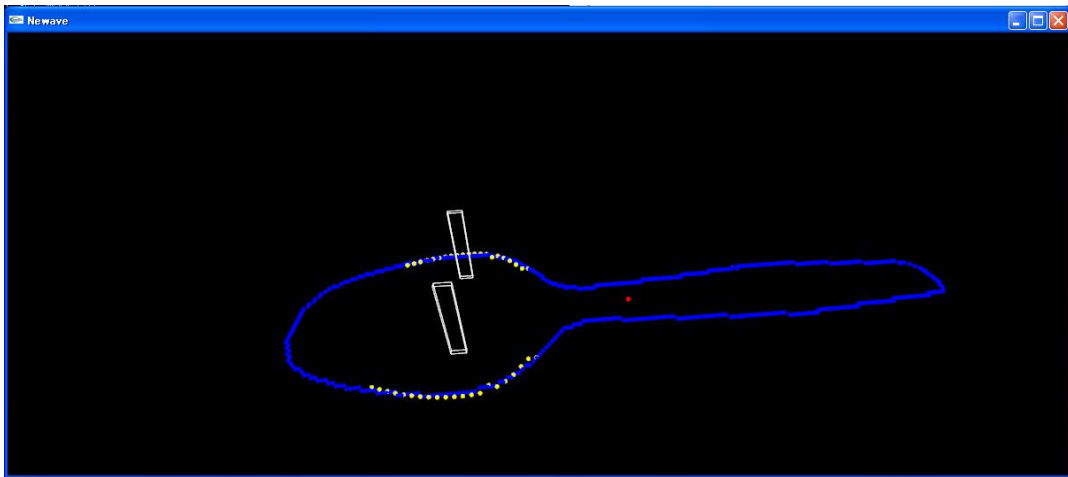


図 7 マッチング後の指先軌跡（黄）とモデル形状（青）

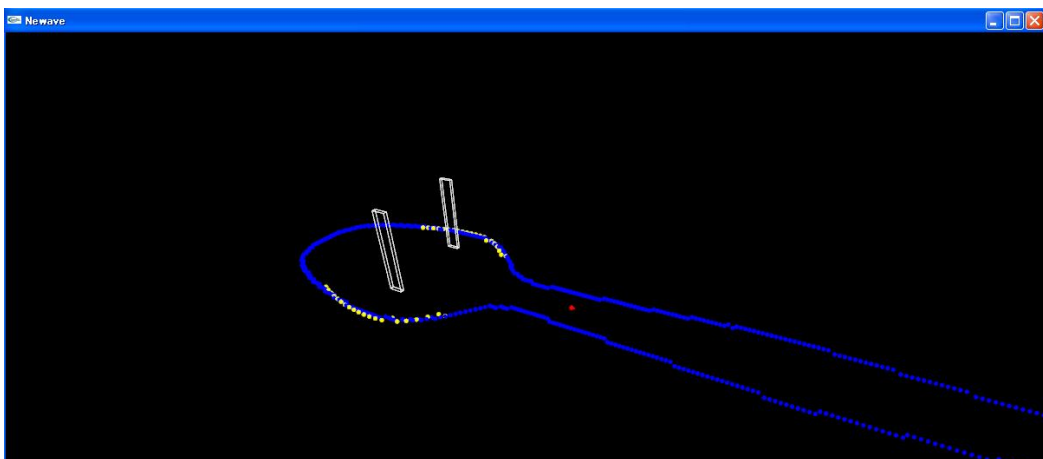


図 8 図 7 の視点を変えたもの

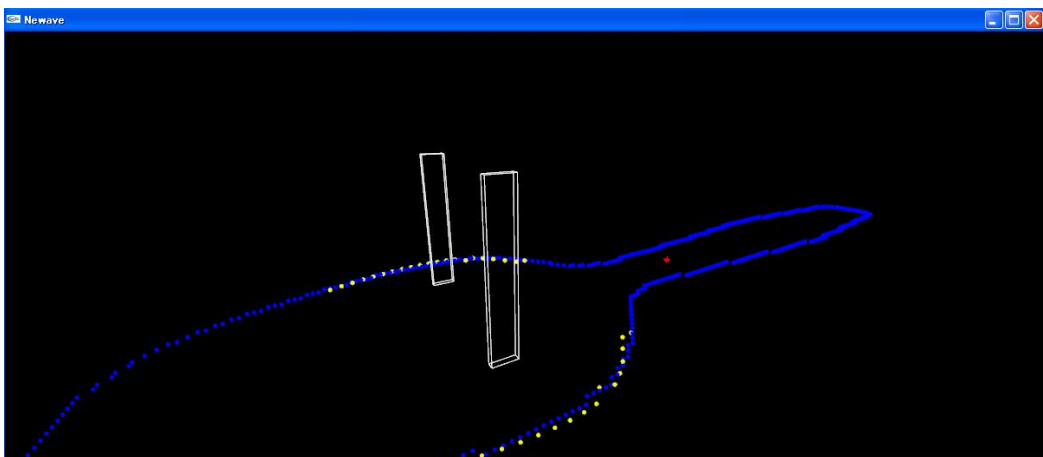


図 9 図 7 の視点を変えたもの

図 10,図 11 は、飲料缶の事例である。

なお、マッチング後のモデル物体の重心位置、姿勢は、統一 I / F 仕様で FingerPositionComp 側に表示される。

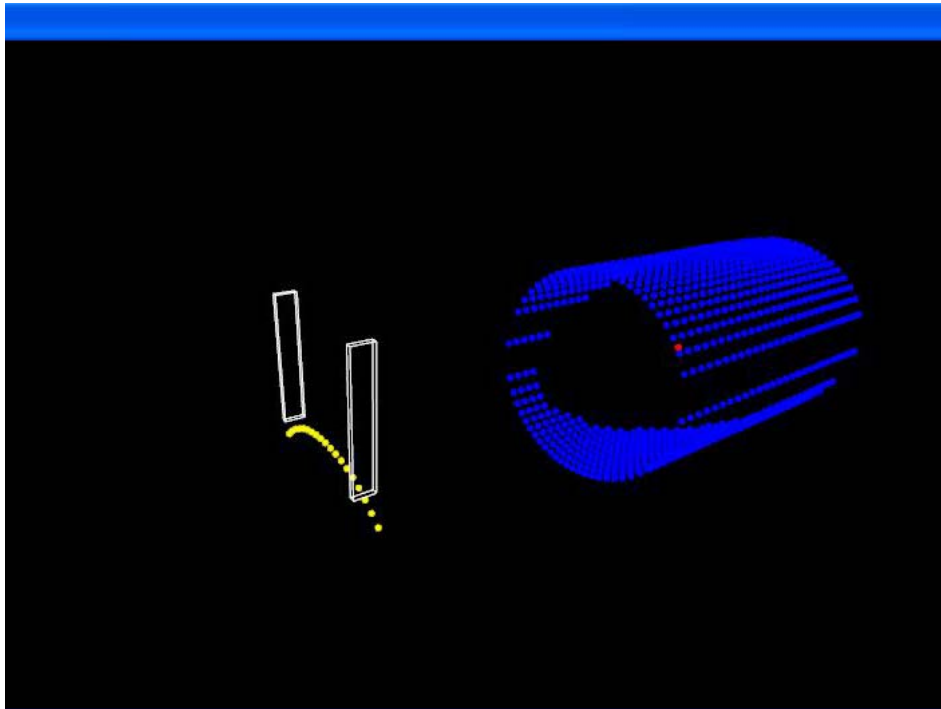


図 10 マッチング前の指先軌跡（黄）とモデル形状（青）。飲料缶の例。

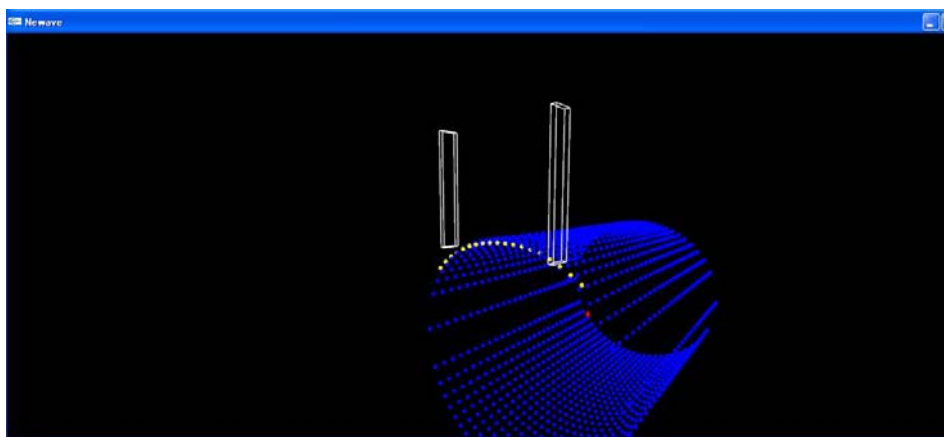


図 11 マッチング後の指先軌跡（黄）とモデル形状（青）。飲料缶の例。

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>