

次世代ロボット知能化技術開発プロジェクト

作業対象物認識モジュール 仕様書・取扱説明書

2011 年 3 月 3 日

国立大学法人 東北大学
株式会社 パイケーク

更新履歴

改定日付

2009/12/14	新規作成
2010/06/14	GTK, SDL に関する記述を追加。 インストール手順の簡素化に対応
2010/06/24	インストール時に必要な Ruby に関する 記述を追加
2011/03/03	OpenRTM 1.0 対応

目次

更新履歴.....	2
目次.....	3
1. はじめに	4
2. 作業対象物認識モジュール	4
2. 1. データ入出力ポート	4
2. 2. コンフィギュレーションインターフェース	4
3. 作業対象物認識モジュールと周辺モジュール	5
3. 1. 画像キャプチャモジュール	7
3. 2. 指定画像送信モジュール	8
3. 2. 処理結果画像表示モジュール	9
3. 3. 処理結果表示モジュール	10
4. 作業対象物認識モジュールのインストール	11
4. 1. 必要環境	11
4. 2. コンパイル手順	12
5. 作業対象物認識モジュールの実行手順	13
5. 1. 学習ファイルの生成手順	13
5. 1. 1. 学習用プログラムの準備	13
5. 1. 2. 対象物の写真を撮る	13
5. 1. 3. 学習ファイルの生成	14
5. 1. 4. 学習ファイルの配置	14
5. 2. 作業対象物認識モジュールの実行手順	15
5. 2. 1. モジュールの起動	15
5. 2. 3. モジュールの接続	15
5. 2. 3. rtc.conf の設定について	21

1. はじめに

入力された画像に対して BagOfKeypoints 法を使用し、作業対象物を認識し、認識した対象物と確度を出力する RT コンポーネントです。

図 1 は作業対象物認識モジュールの概要です。



図 1 モジュール概要

2. 作業対象物認識モジュール

2. 1. データ入出力ポート

名称	データ型	入出力	概要
imageIn	TimedImage	入力	処理対象とする画像を入力するポート
bokOut	BokResult	出力	認識した対象物の名称と、認識した際の確度を示す値を出力
imageOut	TimedImage	出力	入力された画像を処理した結果を出力するポート。デバッグ用

2. 2. コンフィギュレーションインターフェース

下記の初期値の値を変更することで、動的に学習ファイルの変更が可能です

コンフィグレーション名称	データ型	概要	初期値
m_vocabularyFile	string	vocabulary ファイル名を指定	vocabulary
m_svm_xml	string	svm ファイル名を指定	svm.xml
m_class_label	string	classlabel ファイル名を指定	classlabel

3. 作業対象物認識モジュールと周辺モジュール

図 2 は作業対象物認識モジュールを利用する際の、作業対象物認識モジュールと周辺モジュールとの接続と、データの流れを表した図です

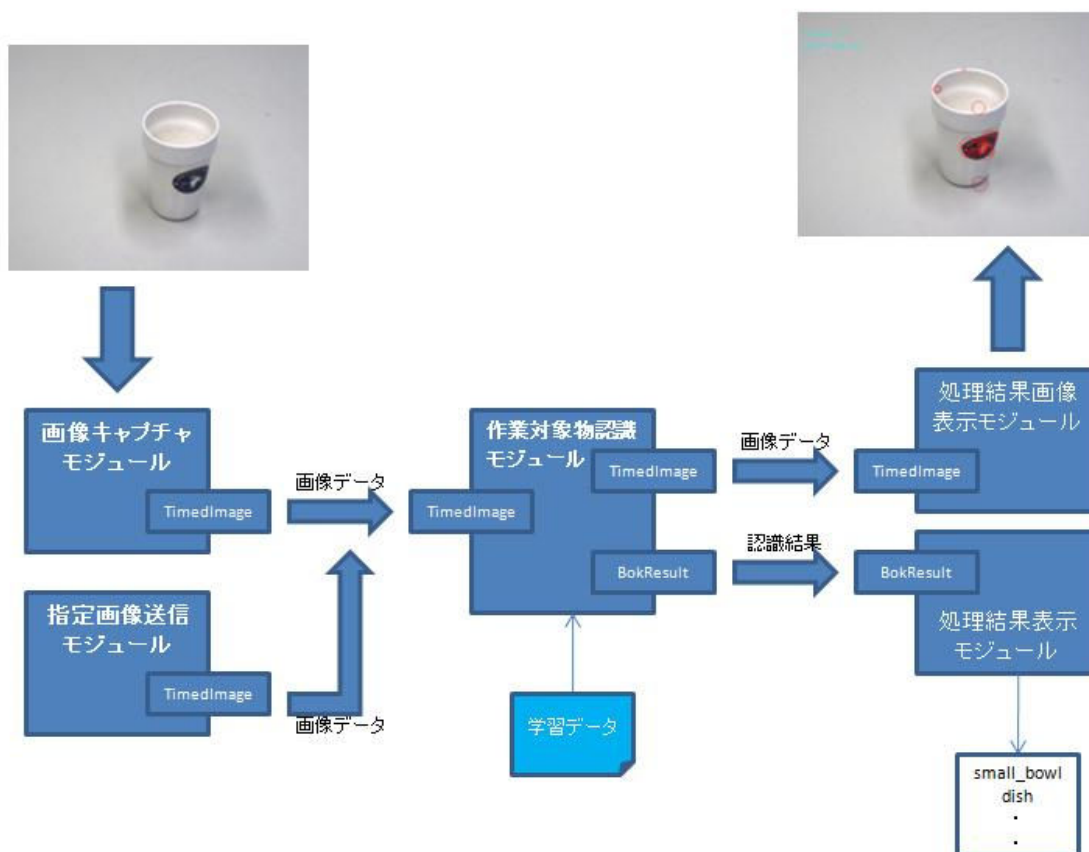


図 2 作業対象物認識モジュールと周辺モジュールについて

USB カメラからキャプチャした画像を認識対象とする場合は画像キャプチャモジュール、静止画を認識対象とする場合は指定画像送信モジュールの出力を作業対象物認識モジュールの `imageIn` ポートへ接続します。

対象物認識モジュールは、入力された画像に対して、`BagOfKeypoints` 法を使用し対象物を認識します。

処理結果画像表示モジュールは、**imageOut** ポートから出力される作業対象物認識モジュールが処理した画像の出力を表示しています。作業対象物認識モジュールは検出した特徴点を赤い丸で図示した画像を処理結果として **imageOut** ポートから出力します。

作業対象物認識の **bokOut** ポートからはモジュールが認識した対象物の名称と、認識した際の対象物が出力されます。処理結果表示モジュールはその出力をコンソールに出力します。

図 3 は、実際に RT System Editor 上で画像キャプチャモジュール、対象物認識モジュール、処理結果画像表示モジュールを接続した例です。

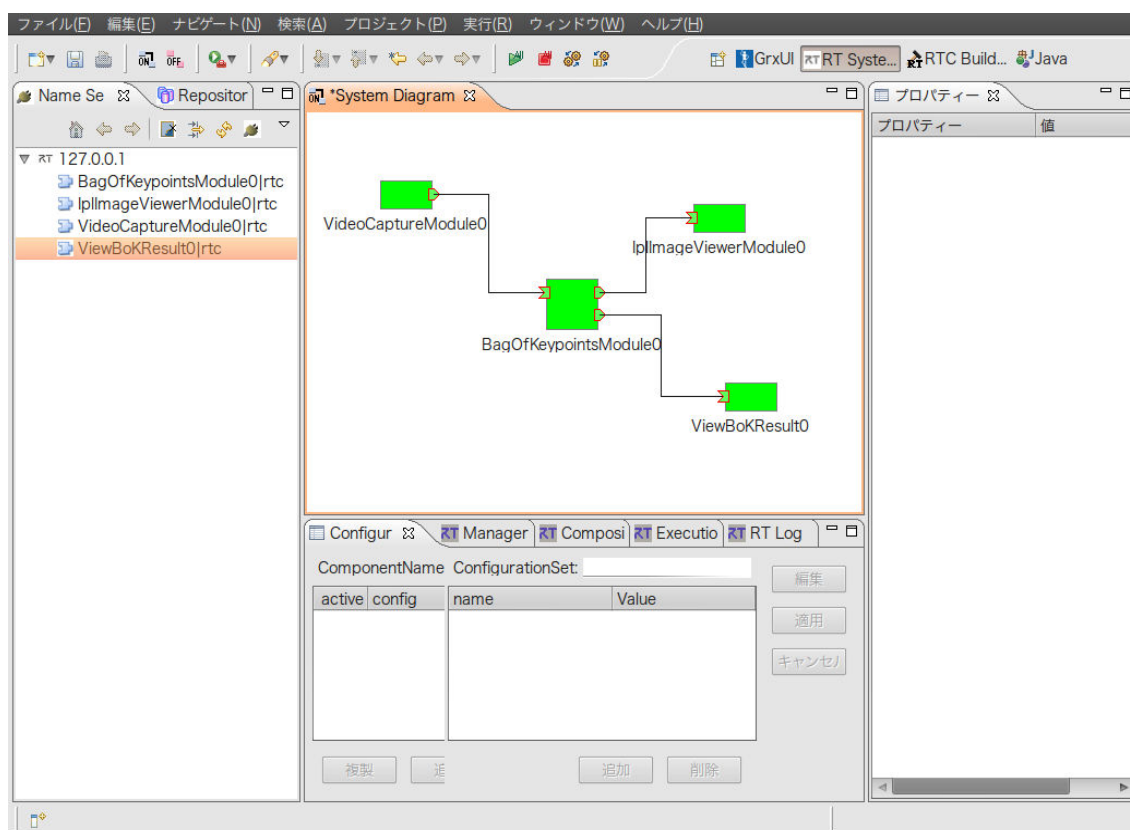


図 3 RT System Editor の接続例

3. 1. 画像キャプチャモジュール

USB カメラなどからキャプチャした画像を、作業対象物認識モジュールに送信するモジュールです。

画像キャプチャモジュールを実行するには、あらかじめ PC に USB カメラを接続している必要があります。

入出力ポートについては以下の通りです。

名称	データ型	入出力	概要
imageOut	TimedImage	出力	USB カメラなどからキャプチャ画像を出力します。

3. 2. 指定画像送信モジュール

指定された画像を作業対象物認識モジュールに送信するモジュールです。アクティブになると、図 4 のようなファイルダイアログが表示されます。



図 4 指定画像送信モジュールのファイルダイアログ

ここで画像ファイルを指定すると、指定画像送信モジュールはその画像を imageOut ポートから出力し続けます。

入出力ポートについては以下の通りです。

名称	データ型	入出力	概要
imageOut	TimedImage	出力	USB カメラなどからキャプチャ画像を出力します。

3. 2. 処理結果画像表示モジュール

作業対象物認識モジュールで処理された画像を表示するためのモジュールです。作業対象物認識モジュールから送信された画像を表示します。

起動すると図 5 のような画像を表示するためのウインドウが 1 つ表示されます。ウインドウはリサイズ可能です。



図 5 処理結果画像表示モジュール

入出力ポートについては以下の通りです。

名称	データ型	入出力	概要
imageIn	TimedImage	入力	ポートに入力された画像をウインドウに表示します。

3. 3. 処理結果表示モジュール

作業対象物認識モジュールが認識した対象物の名称と確度を標準出力に出力するモジュールです。

入出力ポートについては以下の通りです。

名称	データ型	入出力	概要
bokIn	BokResult	入力	作業対象物認識モジュールが認識した対象物の名称と確度を標準出力に出力します。

4. 作業対象物認識モジュールのインストール

4. 1. 必要環境

各モジュールは下記の環境で動作確認を行っています。

- Ubuntu 10.04 Desktop Edition
- OpenRTM-aist-1.0.0-RELEASE
 - 以下の URL を参考にインストールしてください。
 - <http://www.openrtm.org/openrtm/ja/node/849>
- OpenRTM Eclipse tools 1.0-RELEASE
 - <http://www.openrtm.org/openrtm/ja/node/941> から Linux 用全部入りバージョンをダウンロードし、インストールしてください。
- OpenCV
 - OpenCV 2.0 を使用しています
 - `sudo apt-get install libcv4 libcv-dev libhighgui4 libhighgui-dev libcvaux4 libcvaux-dev` のコマンドを実行してインストールしてください。
- GTK
 - `sudo apt-get install libgtk2.0-0 libgtk2.0-dev libglib2.0-0 libglib2.0-dev` のコマンドを実行して、必要なライブラリをインストールしてください。
 - 本コンポーネントでは、ファイルダイアログの表示に使用しています。
- SDL (Simple DirectMedia Layer)
 - `sudo apt-get install libsdl-gfx1.2-4 libsdl-gfx1.2-dev` のコマンドを実行して、必要なライブラリをインストールしてください。
 - 本コンポーネントでは、ウインドウの表示で使用しています
 - <http://www.libsdl.org/>
- Ruby
 - `sudo apt-get install ruby` のコマンドを実行して、必要な実行ファイル・ライブラリをインストールしてください。
 - ソースコードをコンパイルする際に使用しています。

4. 2. コンパイル手順

基本的に各モジュールは*.tar.gz ファイルを取得・展開した後、ディレクトリ内で `make extract & make` を実行すると自動的に各モジュールが展開され、コンパイルされます。

```
$ tar xvfz bag_of_keypoints_module-???????.tar.gz
$ cd bag_of_keypoints_module-???????
$ ls -al
BagOfKeypoint-???????.tar.gz
BagOfKeypointsModule-???????.tar.gz
IplImageOutModule-???????.tar.gz
IplImageViewerModule-???????.tar.gz
Makefile*
README.txt
VideoCaptureModule-???????.tar.gz
ViewBokResult-???????.tar.gz
build.rb*
image-???????.tar.gz

$ make extract
$ make
```

展開・コンパイルされるモジュールは次のとおりです。

```
BagOfKeypointディレクトリ
  学習ファイル生成プログラム

BagOfKeypointsModuleディレクトリ
  作業対象物認識モジュール

VideoCaptureModuleディレクトリ
  画像キャプチャモジュール

IplImageOutModuleディレクトリ
  指定画像送信モジュール

IplImageViewerModuleディレクトリ
  処理結果画像表示モジュール

ViewBokResultディレクトリ
  処理結果表示モジュール

imageディレクトリ
  サンプルイメージ
```

5. 作業対象物認識モジュールの実行手順

5. 1. 学習ファイルの生成手順

本モジュールは、画像から対象物を認識する際、学習ファイルが必要になります。学習ファイルの作り方を説明します

5. 1. 1. 学習用プログラムの準備

BagOfKeypoint-????????.tar.gz をコンパイルすると「trainingBOK」という実行形式ファイルが生成されます。trainingBOK は作業対象物認識モジュールが使用する、対象物の学習データを作成するためのプログラムです。

この trainingBOK に作業対象物とする物の写真を複数枚入力すると、特徴点を抽出し、BagOfKeypoint 法を使用して学習した結果を、classlabel, svm.xml, vocabulary の 3 つのファイルに出力します。

具体的な trainingBOK の使い方については、次に説明します。

5. 1. 2. 対象物の写真を撮る

- 作業対象物の撮影
 - デジタルカメラ等で作業対象物とする物の写真を撮影します。様々な角度から沢山の撮影すると認識する確率が向上します。
- 写真を一つのディレクトリにまとめて配置
 - 撮影した写真をまとめて一つのディレクトリに配置します。ディレクトリの名前は、撮影した物が分かるように命名しておきます。
- 作業対象物を増やす場合
 - 同様に写真を撮って、適切なディレクトリ名にします。ディレクトリの階層は「dish」ディレクトリと同じにしてください。

●
図 6 に以下にディレクトリ構成の例を示します。

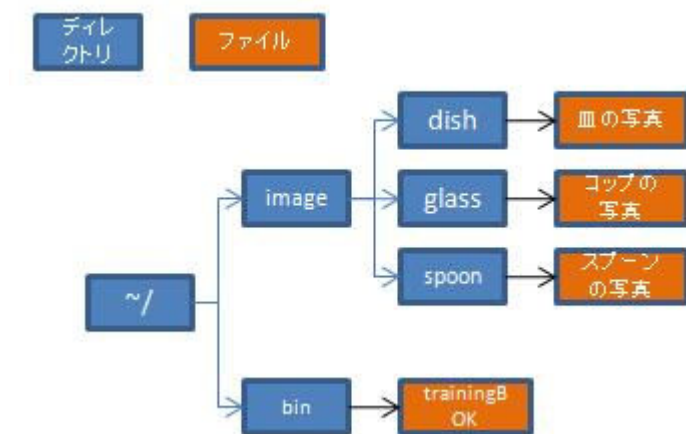


図 6 ディレクトリ構成

5. 1. 3. 学習ファイルの生成

5. 1. 2. のディレクトリ構成の状態、カレントディレクトリが **bin** ディレクトリの場合、次の要領で **trainingBOK** コマンドを実行してください。

```
$ ./trainingBOK ../image
```

trainingBOK が正常に終了すると、カレントディレクトリに **classlabel**, **svm.xml**, **vocabulary** の 3 つのファイルが生成されます。

5. 1. 4. 学習ファイルの配置

5. 1. 3. で生成した **classlabel**, **svm.xml**, **vocabulary** の 3 つのファイルを作業対象物認識モジュールと同じディレクトリに配置してください。

5. 2. 作業対象物認識モジュールの実行手順

以下の4つのモジュールを用いて、手順を説明します

- 作業対象物認識モジュール (BagOfKeypointsModule)
- 画像キャプチャモジュール (VideoCaptureModule)
- 処理結果画像表示モジュール (IplImageViewerModule)
- 処理結果表示モジュール (ViewBoKResult)

5. 2. 1. モジュールの起動

モジュールを展開&コンパイルしたディレクトリ内で、make run コマンドで各モジュールを起動します。

```
$ make run
```

5. 2. 3. モジュールの接続

OpenRTM Eclipse tools を起動して、各モジュールのポートを接続します。

RT System Editor のパースペクティブが"RT System Editor"に設定されていない場合は、メニューから「ウインドウ」->「パースペクティブを開く」->「その他」を選び、表示されているダイアログの中から" RT System Editor "を選択して、パースペクティブを" RT System Editor "に設定してください。

" RT System Editor"パースペクティブの状態で、左側の NameServerView に図 7 のように 4 つのモジュールが表示されることを、確認して下さい

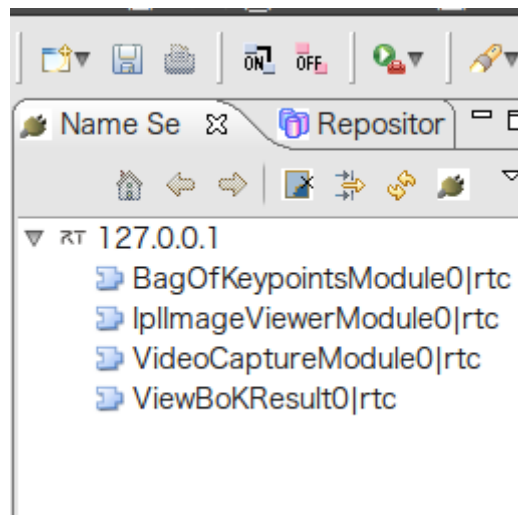


図 7 NameServerView の表示

図 8 の"Open New System Editor"のボタンを押して、"System Diagram"のウインドウを開きます。

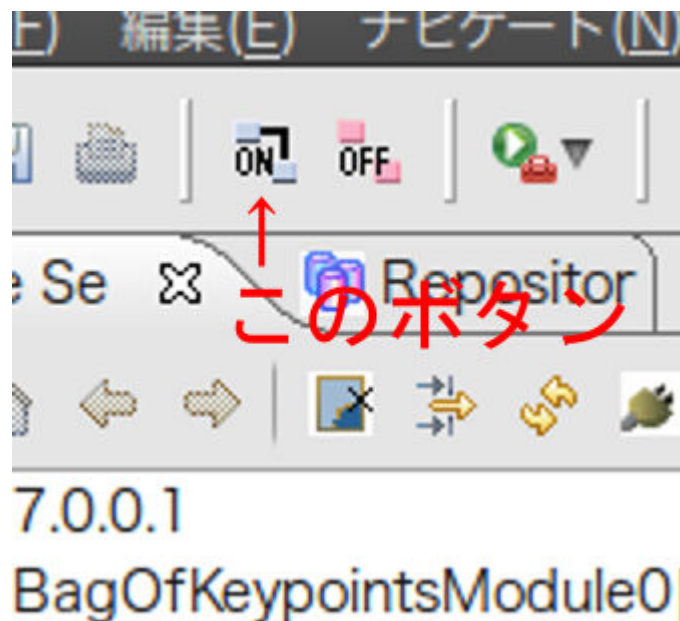


図 8 Open New System Editor ボタン

NameServiceView から 4 つのモジュールを 1 つずつドラッグして、"System Diagram" ウィンドウへモジュールをドロップします。"System Diagram"へ配置するモジュールは、図 9 のように BagOfKeypointsModule0、VideoCaptureModule0、IplImageViewerModule0、ViewBoKResult0 の 4 つです。

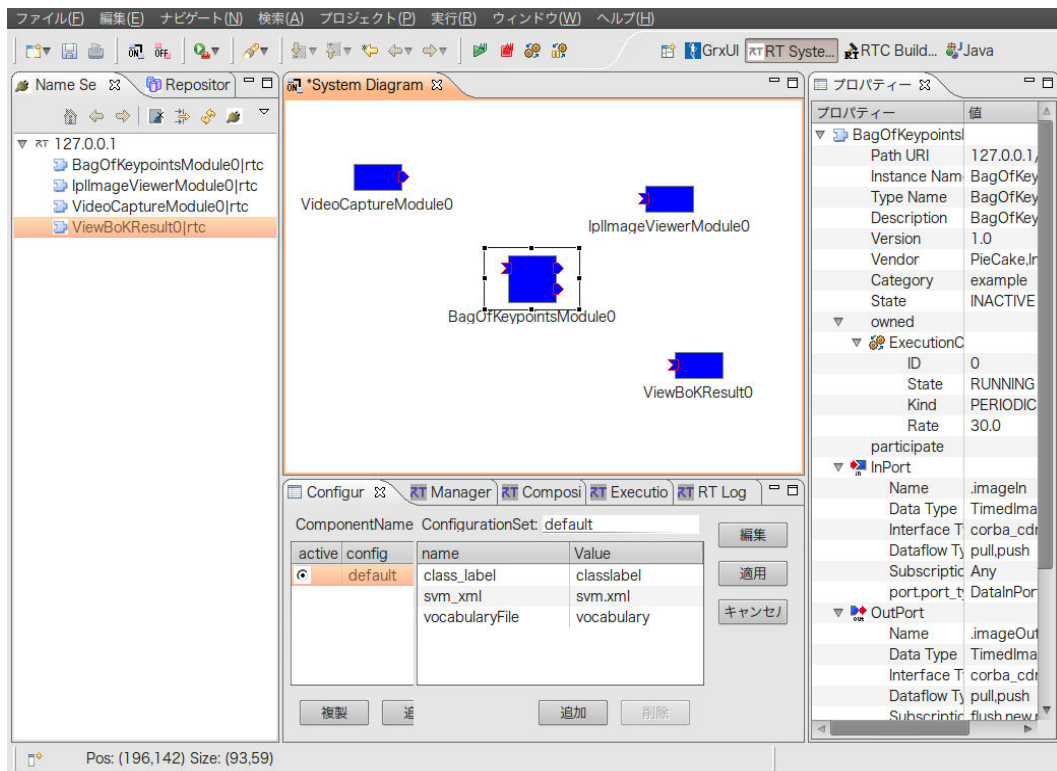


図 9 System Diagram へ配置したモジュール

モジュールの配置が完了したら、各モジュールの入出力ピンを接続します。接続するピンは次の通りです。

モジュール名	出力ピン	モジュール名	入力ピン	概要
VideoCaptureModule	imageOut	BagOfKeypointsModule	imageIn	USB カメラから入力された画像
BagOfKeypointsModule	imageOut	IplImageViewerModuleComp	imageIn	作業対象物認識モジュールで処理した結果の画像
BagOfKeypointsModule	bokOut	ViewBoKResult	bokIn	作業対象物認識モジュールが認識した対象物名と確率

図 10 はポートを接続した後の System Diagram の表示です。

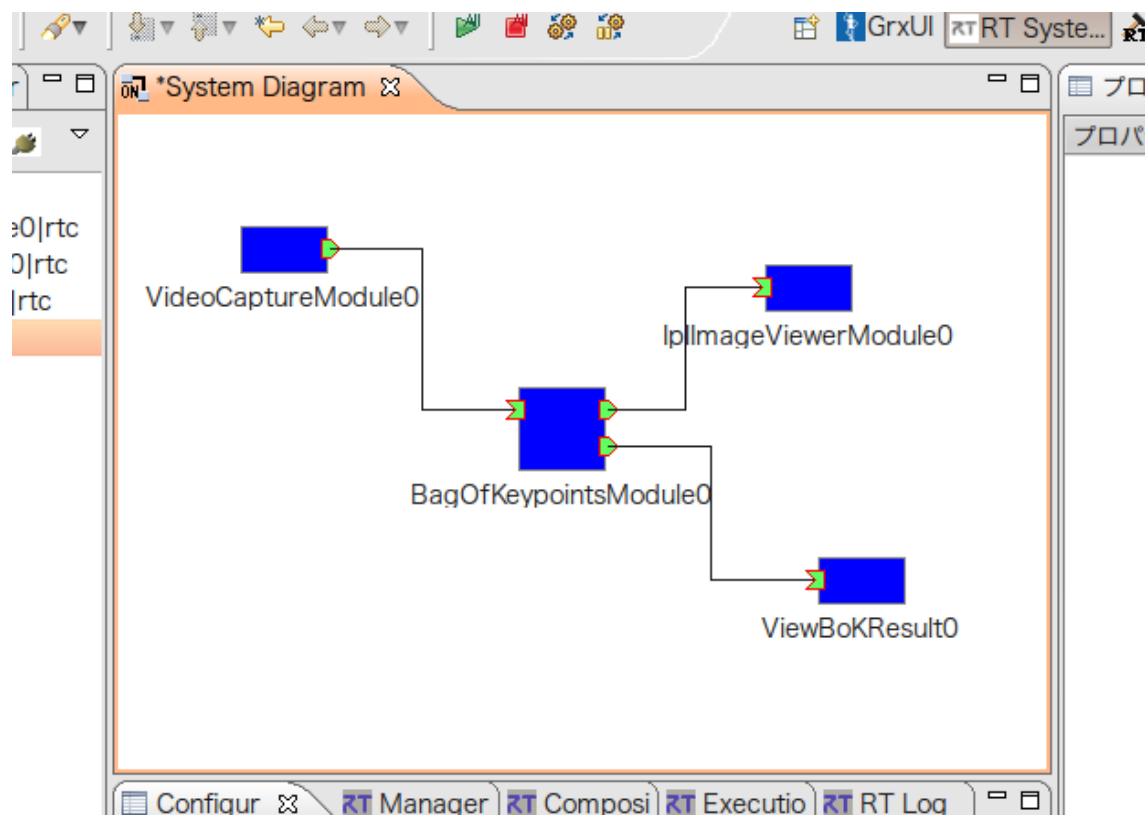


図 10 接続後の様子

接続が完了したら、全てのモジュールを **Activate** 状態にします。モジュールの上で右クリックしてポップアップメニューを表示し、メニューから"**Activate**"を選択します。すべてのモジュールを **Activate** 状態に設定すると、図 11 のようにすべてのモジュールが緑色に変化して、**Activate** 状態であることを表します。

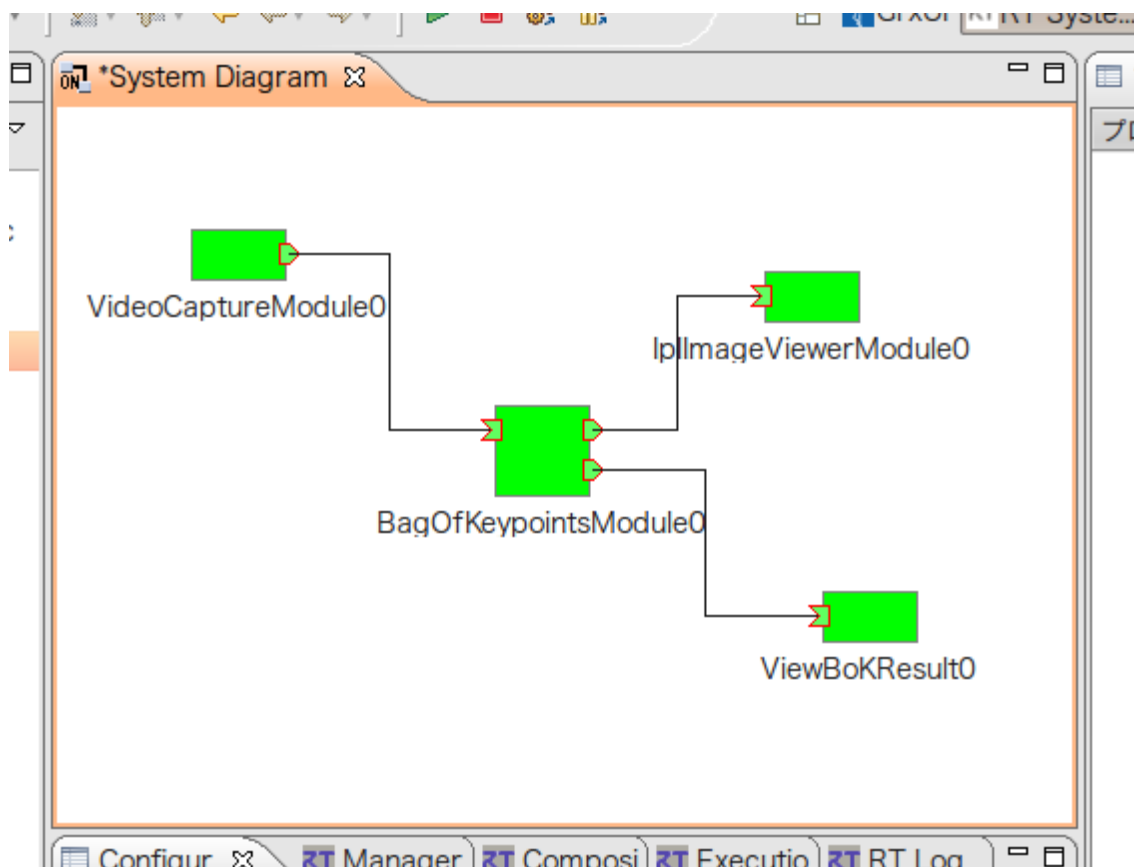


図 11 Activate 状態

この状態で USB カメラを対象物に向けると、作業対象物認識モジュールが動作し、認識可能な対象物であれば認識した対象物の名称と確度を出力します。図 12 は処理結果画像表示モジュールのウィンドウに表示される、作業対象物認識モジュールの処理結果画像の表示例です。



図 12 処理結果画像の表示

5. 2. 3. rtc.conf の設定について

各モジュールがあるディレクトリごとに、rtc.conf というモジュールの実行に必要な設定ファイルが用意されています。

標準で添付している rtc.conf の内容は次の通りです。

```
corba.nameservers: localhost:2809
corba.args: -ORBgiopMaxMsgSize: 3145728
nameing.formats: %n.rtc
logger.log_level: TRACE
exec_cxt.periodic.rate: 30
manager.modules.load_path: .
manager.modules.abs_path_allowed: yes
```

次に、上記の rtc.conf で OpenRTM の標準の設定とは異なる設定を行っている部分について説明します。

OpenRTM 標準の設定では、一度の通信で 2MByte までのデータ送信しか対応していません。画像データなどの大きなデータをポートで送受信する際には、送受信可能なデータの最大サイズを指定する必要があります。作業対象物認識モジュールを利用する際には、2 行目の一度の通信で送ることができるデータの最大サイズを指定する必要があります。

また、通常の OpenRTM では動作周期が 1 秒間に 1000 回に設定されていますが、USB カメラからキャプチャした画像を処理対象とする場合は 30fps 程度で処理することができれば十分だと考えられます。そのため、5 行目の設定で動作周波数を 1 秒間に 30 回に設定しています。

rtc.conf の設定の詳細については、OpenRTM-Aist の Web サイトを参照して下さい。