User Manual - DetectAndLocalize

This document provides all the required information to estimate the pose of an object, given a model of the object, using RGB-D Sensors (such as Microsoft Kinect, Orbec Astra Pro and Intel Euclid RealSense).

1 Download

The code to estimate the pose of an object can be found here.

2 Prerequisites

The following prerequisites are required to build and run the code.

- OpenCV 2.4.8 or higher
- PCL 1.7.1 or higher
- ROS Indigo or higher
- CMAKE 2.8.9 or higher

3 CMake and Build

To build the code execute the following steps :

1. Create a build directory in *DetectAndLocalize* folder.
   ```
   $ cd Object-Pose-Estimation-master/DetectAndLocalize
   $ mkdir build
   $ cd build
   ```

2. CMAKE and Build
   ```
   $ cmake ..
   $ make
   ```

4 Command Line Interface

Usage : `./DetectAndLocalize [-parameters=value]`

Parameter	Default Value	Description
h help	false	Prints the help
o object	brick	Object to get pose [brick(default); drill ; yellow]
l limits	-0.5,0.5,-0.5,0.3,0.6,1.7	Xmin,Xmax,Ymin,Ymax,Zmin,Zmax for field of processing of sensor (in m) No Spaces!
sc scenario	0	Scenario table - 0 ; nottable - 1
p path	empty	Path to new object model (pointcloud (.pcd, .ply) format)
s sensor	astra	Sensor to choose [kinect, astra, euclid]

5 Working and Example

1. In order to estimate the pose of an object whose model is already available in the models provided along with code (brick, drill and yellow cylinder), provide the necessary value to the '-o' object parameter using command line. And to estimate the pose of a new object, then provide the path of the new object's model using '-p' parameter in the command line.
   ```
   $ ./DetectAndLocalize -p=path_to_new_model
   ```

2. For example, to estimate the pose of a brick present on a plane which is in the X, Y and Z distance range, in sensor frame of reference as (-0.3,0.3,-0.7,0.3,0.5,1.5) with a kinect sensor. Then we need to run :
   ```
   $ ./DetectAndLocalize -l=-0.3,0.3,-0.7,0.3,0.5,1.5 -s=kinect
   ```