

## Visual Odometry using Intensity assisted Iterative Closest Point

This ros package provides the implementation for the visual odometry method described in [1].

- The main.cpp handles the subscribed and published topics.
- The camera\_tracker class handles the camera pose integration, keyframe selection etc.
- The iaicp class handles the pose estimation problem between two frames.

If you use our code for your research purpose, please cite our paper [1].

[1] **Fast Visual Odometry Using Intensity-Assisted Iterative Closest Point**, Shile Li and Dongheui Lee, in IEEE Robotics and Automation Letters Volume 1 Issue 2, 2016

### Requirements:

- Ubuntu 14.04
- Ros-Indigo
- Point Cloud Library

### Installation:

- Download the source code from website: <http://www.hri.ei.tum.de/en/download/>
- Unzip the downloaded file and put it under the /src folder in your catkin workspace:  
~/catkin\_ws/src/
- In terminal:  
cd ~/catkin\_ws  
catkin\_make -DCMAKE\_BUILD\_TYPE=Release

### Usage with registered point cloud:

- In param.yaml
  - set /sub/points/topic to your subscribed ros topic name
  - set /usepoints to true
  - set camera parameters /fx /fy /cx /cy /width /height
- start the OpenNI camera driver or play the recorded rosbag
- roscd iicp; rosload param.yaml; rosrn iicp iicp;

### Usage with rgb/depth image:

- In param.yaml
  - set /sub/rgb/topic and /sub/depth/topic to your subscribed ros topic names
  - set /usepoints to false
  - set camera parameters /fx /fy /cx /cy /width /height
- start the OpenNI camera driver or play the recorded rosbag
- roscd iicp; rosload param.yaml; rosrn iicp iicp;

### Visualization:

- start rviz
- load the configuration file 'iicp.rviz' included in the /iicp folder