

Dismiss

Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

Sign up

Tools for using the Kinect One (Kinect v2) in ROS

243 commits

2 branches

0 releases

24 contributors

Apache-2.0

Branch: master

New pull request

Find file

Clone or download

bbferka Merge pull request #467 from dipendras/master ... Latest commit 0e2c5f6 on 3 Jul 2018

iai_kinect2	renamed depth_registration package to kinect2_registration.	4 years ago
kinect2_bridge	Change help()	2 years ago
kinect2_calibration	Merge pull request #467 from dipendras/master	7 months ago
kinect2_registration	enable exceptions	a year ago
kinect2_viewer	removed obsolete isnan and check the integer depth value instead of t...	3 years ago
LICENSE	Initial commit	5 years ago
README.md	Merge pull request #381 from samarth-robo/samarth-robo-patch-readme	11 months ago

README.md

IAI Kinect2

Maintainer

- Thiemo Wiedemeyer <wiedemeyer@cs.uni-bremen.de>, Institute for Artificial Intelligence, University of Bremen

Read this first

Please read this README and the ones of the individual components thoroughly before asking questions. We get a lot of repeated questions, so when you have a problem, we urge everyone to check the [github issues \(including closed ones\)](#). Your issue is very likely discussed there already.

The goal of this project is to give you a driver and the tools needed to receive data from the Kinect-2 sensor, in a way useful for robotics. You will still need to know how to use ROS to make use of it. Please follow the [ROS tutorials](#). You will also need to learn how to work with point-clouds, or depth-clouds, or images (computer vision) to do useful things with the data.

*Note: **Please use the GitHub issues** for questions and problems regarding the iai_kinect2 package and its components. **Do not write emails.***

Table of contents

- Description
- FAQ
- Dependencies
- Install
- GPU acceleration
 - OpenCL with AMD

- [OpenCL/CUDA with Nvidia](#)
- [OpenCL with Intel](#)
- [Citation](#)
- [Screenshots](#)

Description

This is a collection of tools and libraries for a ROS Interface to the Kinect One (Kinect v2).

It contains:

- [a calibration tool](#) for calibrating the IR sensor of the Kinect One to the RGB sensor and the depth measurements
- [a library](#) for depth registration with OpenCL support
- [the bridge](#) between [libfreenect2](#) and [ROS](#)
- [a viewer](#) for the images / point clouds

FAQ

If I have any question or something is not working, what should I do first?

First you should look at this FAQ and the [FAQ from libfreenect2](#). Secondly, look at [issue page from libfreenect2](#) and the [issue page of iai_kinect2](#) for similar issues and solutions.

Point clouds are not being published?

Point clouds are only published when the launch file is used. Make sure to start `kinect2_bridge` with `roslaunch kinect2_bridge kinect2_bridge.launch`.

Will it work with OpenCV 3.0

Short answer: No.

Long answer: Yes, it is possible to compile this package with OpenCV 3.0, but it will not work. This is because `cv_bridge` is used, which itself is compiled with OpenCV 2.4.x in ROS Indigo/Jade and linking against both OpenCV versions is not possible.

Working support for OpenCV 3.0 might come with a future ROS release.

kinect2_bridge is not working / crashing, what is wrong?

There are many reasons why `kinect2_bridge` might not be working. The first thing to find out whether the problem is related to `kinect2_bridge` or `libfreenect2`. A good tool for testing is `Protonect`, it is a binary located in `libfreenect2/build/bin/Protonect`. It uses `libfreenect2` directly with a minimal dependency on other libraries, so it is a good tool for the first tests.

Execute:

- `./Protonect gl` to test OpenGL support.
- `./Protonect cl` to test OpenCL support.
- `./Protonect cpu` to test CPU support.

Before running `kinect2_bridge` please make sure `Protonect` is working and showing color, depth and ir images. If some of them are black, then there is a problem not related to `kinect2_bridge` and you should look at the issues from the `libfreenect2` GitHub page for help.

If one of them works, try out the one that worked with `kinect2_bridge`: `roslaunch kinect2_bridge kinect2_bridge _depth_method:=<opengl|opencl|cpu>`. You can also change the registration method with `_reg_method:=<cpu|opencl>`.

Protonect works fine, but kinect2_bridge is still not working / crashing.

If that is the case, you have to make sure that `Protonect` uses the same version of `libfreenect2` as `kinect2_bridge` does. To do so, run `make` and `sudo make install` in the build folder again. And try out `kinect2_bridge` again.

```
cd libfreenect2/build
make & sudo make install
```

Also make sure that you are not using OpenCV 3.0.

If it is still crashing, compile it in debug and run it with gdb:

```
cd <catkin_ws>
catkin_make -DCMAKE_BUILD_TYPE="Debug"
cd devel/lib/kinect2_bridge
gdb kinect2_bridge
# inside gdb: run until it crashes and do a backtrace
run
bt
quit
```

Open an issue and post the problem description and the output from the backtrace (`bt`).

kinect2_bridge hangs and prints "waiting for clients to connect"

This is the normal behavior. 'kinect2_bridge' will only process data when clients are connected (ROS nodes listening to at least one of the topics). This saves CPU and GPU resources. As soon as you start the `kinect_viewer` or `rostopic hz` on one of the topics, processing should start.

rosdep: Cannot locate rosdep definition for [kinect2_bridge] or [kinect2_registration]

`rosdep` will output errors on not being able to locate `[kinect2_bridge]` and `[kinect2_registration]`. That is fine because they are all part of the `iai_kinect2` package and `rosdep` does not know these packages.

Protonect or kinect2_bridge outputs [TransferPool::submit] failed to submit transfer

This indicates problems with the USB connection.

I still have an issue, what should I do?

First of all, check the issue pages on GitHub for similar issues, as they might contain solutions for them. By default you will only see the open issues, but if you click on `closed` you will see the ones solved. There is also a search field which helps to find similar issues.

If you found no solution in the issues, feel free to open a new issue for your problem. Please describe your problem in detail and provide error messages and log output.

Dependencies

- ROS Hydro/Indigo
- OpenCV (2.4.x, using the one from the official Ubuntu repositories is recommended)
- PCL (1.7.x, using the one from the official Ubuntu repositories is recommended)
- Eigen (optional, but recommended)
- OpenCL (optional, but recommended)
- [libfreenect2](#) (>= v0.2.0, for stability checkout the latest stable release)

Install

1. Install the ROS. [Instructions for Ubuntu 14.04](#)
2. [Setup your ROS environment](#)
3. Install [libfreenect2](#):

Follow [the instructions](#) and enable C++11 by using `cmake .. -DENABLE_CXX11=ON` instead of `cmake ..`. If you are compiling `libfreenect2` with CUDA, use `cmake .. -DENABLE_CXX11=ON -DCUDA_PROPAGATE_HOST_FLAGS=off`.

If something is not working, check out the latest stable release, for example `git checkout v0.2.0`.

4. Clone this repository into your catkin workspace, install the dependencies and build it:

```
cd ~/catkin_ws/src/
git clone https://github.com/code-iai/iai_kinect2.git
```

```
cd iai_kinect2
rosdep install -r --from-paths .
cd ~/catkin_ws
catkin_make -DCMAKE_BUILD_TYPE="Release"
```

Note: `rosdep` will output errors on not being able to locate `[kinect2_bridge]` and `[depth_registration]`. That is fine because they are all part of the `iai_kinect2` package and `rosdep` does not know these packages.

Note: If you installed `libfreenect2` somewhere else than in `$HOME/freenect2` or a standard location like `/usr/local` you have to specify the path to it by adding `-Dfreenect2_DIR=path_to_freenect2/lib/cmake/freenect2` to `catkin_make`.

5. Connect your sensor and run `kinect2_bridge`:

```
roslaunch kinect2_bridge kinect2_bridge.launch
```

6. Calibrate your sensor using the `kinect2_calibration`. [Further details](#)

7. Add the calibration files to the `kinect2_bridge/data/<serialnumber>` folder. [Further details](#)

8. Restart `kinect2_bridge` and view the results using `roslaunch kinect2_viewer kinect2_viewer kinect2 sd cloud`.

GPU acceleration

OpenCL with AMD

Install the latest version of the AMD Catalyst drivers from <https://support.amd.com> and follow the instructions. Also install `opengl-headers`.

```
sudo apt-get install opengl-headers
```

OpenCL/CUDA with Nvidia

Go to developer.nvidia.com/cuda-downloads and select `linux`, `x86_64`, `Ubuntu`, `14.04`, `deb(network)`. Download the file and follow the instructions. Also install `nvidia-modprobe` and `opengl-headers`.

```
sudo apt-get install nvidia-modprobe opengl-headers
```

You also need to add CUDA paths to the system environment, add these lines to your `~/.bashrc`:

```
export LD_LIBRARY_PATH="/usr/local/cuda/lib64:${LD_LIBRARY_PATH}"
export PATH="/usr/local/cuda/bin:${PATH}"
```

A system-wide configuration of the library path can be created with the following commands:

```
echo "/usr/local/cuda/lib64" | sudo tee /etc/ld.so.conf.d/cuda.conf
sudo ldconfig
```

OpenCL with Intel

You can either install a binary package from a PPA like [ppa:floe/beignet](https://ppa.launchpad.net/floe/beignet/ubuntu-14.04-ppa), or build `beignet` yourself. It's recommended to use the binary from the PPA.

```
sudo add-apt-repository ppa:floe/beignet && sudo apt-get update
sudo apt-get install beignet beignet-dev opengl-headers
```

Citation

If you used `iai_kinect2` for your work, please cite it.

```
@misc{iai_kinect2,  
  author = {Wiedemeyer, Thiemo},  
  title = {{IAI Kinect2}},  
  organization = {Institute for Artificial Intelligence},  
  address = {University Bremen},  
  year = {2014 -- 2015},  
  howpublished = {\url{https://github.com/code-iai/iai_kinect2}},  
  note = {Accessed June 12, 2015}  
}
```

The result should look something similar to this (may depend on the bibliography style used):

T. Wiedemeyer, "IAI Kinect2," https://github.com/code-iai/iai_kinect2,
Institute for Artificial Intelligence, University Bremen, 2014 – 2015,
accessed June 12, 2015.

Screenshots

Here are some screenshots from our toolkit:



