
scikit-surgerynditracker Documentation

Stephen Thompson

Apr 19, 2023

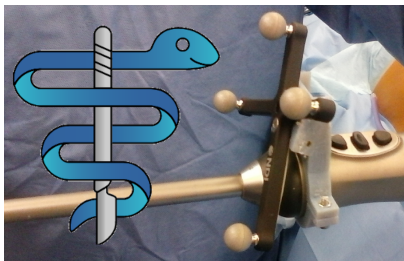
Contents

1	scikit-surgerynditracker	3
1.1	Installing	4
1.2	Using	4
1.3	Developing	4
1.4	Licensing and copyright	5
1.5	Acknowledgements	5
2	NDI Tracking	7
	Index	9

Source [code](#) is available on GitHub.

CHAPTER 1

scikit-surgerynditracker



scikit-surgerynditracker is a python interface for Northern Digital (NDI) trackers. It should work with Polaris Viera, Spectra, and Vega optical trackers and Aurora electromagnetic trackers. Tracking data is output as NumPy arrays.

Author: Stephen Thompson

scikit-surgerynditracker is part of the [SciKit-Surgery](#) software project, developed at the [Wellcome EPSRC Centre for Interventional and Surgical Sciences](#), part of [University College London \(UCL\)](#).

1.1 Installing

```
pip install scikit-surgerynditracker
```

1.2 Using

Configuration is done using Python libraries at instantiation. Invalid configuration should raise exceptions. Tracking data is returned in a set of lists, containing the port handles, timestamps, framenumbers, the tracking data and a tracking quality metric. By default tracking data is returned as a 4x4 NumPy array, though can be returned as a quaternion by changing the configuration.

```
from sksurgerynditracker.nditracker import NDITracker
SETTINGS = {
    "tracker type": "polaris",
    "romfiles" : ["../data/8700339.rom"]
}
TRACKER = NDITracker(SETTINGS)

TRACKER.start_tracking()
port_handles, timestamps, framenumbers, tracking, quality = TRACKER.get_frame()
for t in tracking:
    print (t)
TRACKER.stop_tracking()
TRACKER.close()
```

See `demo.py` for a full example

1.3 Developing

1.3.1 Cloning

You can clone the repository using the following command:

```
git clone https://github.com/SciKit-Surgery/scikit-surgerynditracker
```

1.3.2 Running the tests

You can run the unit tests by installing and running tox:

```
pip install tox
tox
```

1.3.3 Contributing

Please see the [contributing guidelines](#).

1.3.4 Useful links

- [Source code repository](#)
- [Documentation](#)

1.4 Licensing and copyright

Copyright 2018 University College London. scikit-surgerynditracker is released under the BSD-3 license. Please see the [license file](#) for details.

1.5 Acknowledgements

Supported by [Wellcome](#) and [EPSRC](#).

Class implementing communication with NDI (Northern Digital) trackers

class sksurgerynditracker.nditracker.**NDITracker** (*configuration*)

Bases: sksurgerycore.baseclasses.tracker.SKSBASETracker

Class for communication with NDI trackers. Should support Polaris, Aurora, and Vega. Currently only tested with wireless tools on Vega

close ()

Closes the connection to the NDI Tracker and deletes the tracker device.

Raises Exception – ValueError

get_frame ()

Gets a frame of tracking data from the NDI device.

Returns

port_numbers : list of port handles, one per tool

time_stamps : list of timestamps (cpu clock), one per tool

frame_numbers : list of framenumbers (tracker clock) one per tool

tracking : list of 4x4 tracking matrices, rotation and position, or if use_quaternions is true, a list of tracking quaternions, column 0-2 is x,y,z column 3-6 is the rotation as a quaternion.

tracking_quality : list the tracking quality, one per tool.

Note: The time stamp is based on the host computer clock. Read the following extract from NDI's API Guide for advice on what to use: "Use the frame number, and not the host computer clock, to identify when data was collected. The frame number is incremented by 1 at a constant rate of 60 Hz. Associating a time from the host computer clock to replies from the system assumes that the duration of time between raw data collection and when the reply is received by the host computer is constant. This is not necessarily the case."

get_tool_descriptions ()

Returns the port handles and tool descriptions

start_tracking()

Tells the NDI devices to start tracking.

Raises Exception – ValueError

stop_tracking()

Tells the NDI devices to stop tracking.

Raises Exception – ValueError

`skisurgerynditracker.nditracker.int2byte()`

`S.pack(v1, v2, ...) -> bytes`

Return a bytes object containing values v1, v2, ... packed according to the format string S.format. See `help(struct)` for more on format strings.

C

`close()` (*skurgerynditracker.nditracker.NDITracker*
method), 7

G

`get_frame()` (*skurgeryndi-*
tracker.nditracker.NDITracker method), 7

`get_tool_descriptions()` (*skurgeryndi-*
tracker.nditracker.NDITracker method), 7

I

`int2byte()` (*in module skurgeryndi-*
tracker.nditracker), 8

N

`NDITracker` (*class in skurgerynditracker.nditracker*),
7

S

`skurgerynditracker.nditracker` (*module*), 7

`start_tracking()` (*skurgeryndi-*
tracker.nditracker.NDITracker method), 7

`stop_tracking()` (*skurgeryndi-*
tracker.nditracker.NDITracker method), 8