electric | fuerte | groovy | hydro | indigo | jade Documentation Status

***vision_visp (/vision_visp?distro=jade)**: visp_auto_tracker (/visp_auto_tracker?distro=jade) | visp_bridge (/visp_bridge?distro=jade) | visp_camera_calibration (/visp_camera_calibration?distro=jade) | visp_hand2eye_calibration (/visp_hand2eye_calibration?distro=jade) | visp_tracker (/visp_tracker?distro=jade)*

Package Links

- **Code API (http://docs.ros.org/jade/api/visp_hand2eye_calibration/html)**
- **Msg/Srv API (http://docs.ros.org/jade/api/visp_hand2eye_calibration/html/index-msg.html)**
- FAQ (http://answers.ros.org/questions/scope:all/sort:activity-desc/tags:visp_hand2eye_calibration/page:1/)
- Changelog (http://docs.ros.org/jade/changelogs/visp_hand2eye_calibration/changelog.html)
- Change List (/vision_visp/ChangeList)
- Reviews (/visp_hand2eye_calibration/Reviews)

Dependencies (10)
Used by (1)
Jenkins jobs (9)

Package Summary

✔ **Released** ✔ **Continuous integration** ✔ **Documented**

visp_hand2eye_calibration estimates the camera position with respect to its effector using the ViSP library.

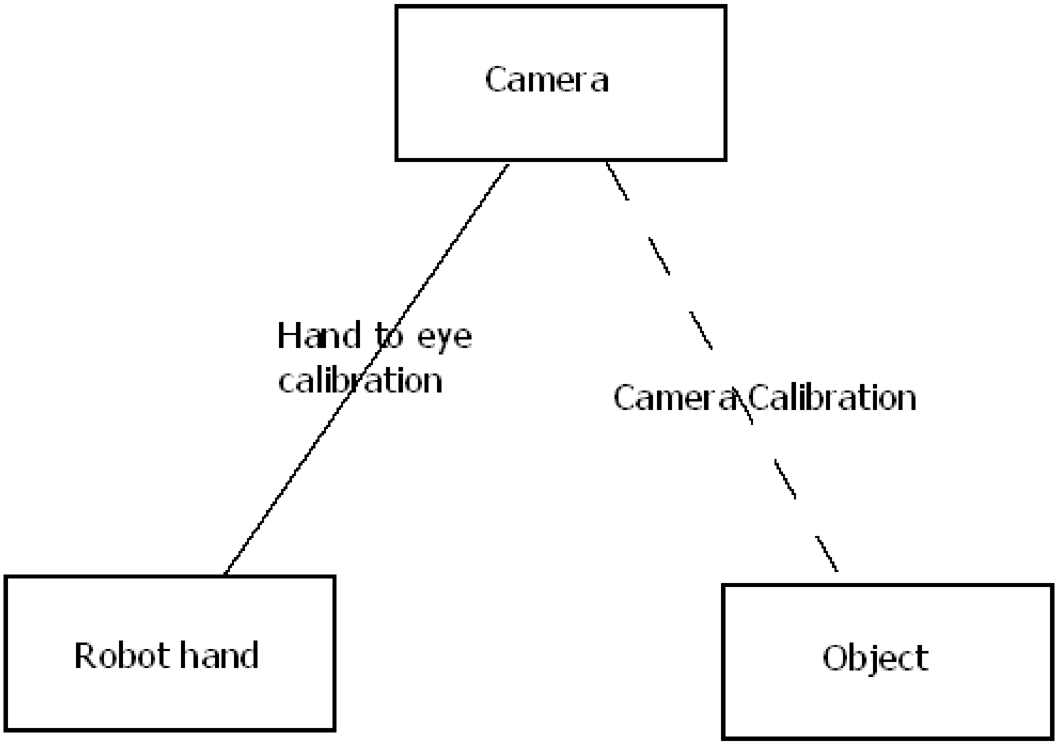
- Maintainer status: maintained
- Maintainer: Fabien Spindler <Fabien.Spindler AT inria DOT fr>
- Author: Filip Novotny
- License: GPLv2
- Source: git https://github.com/lagadic/vision_visp.git (https://github.com/lagadic/vision_visp) (branch: jade)

Conteúdo

1. Overview

visp_hand2eye_calibration is a ROS package that computes extrinsic camera parameters : the constant transformation from the hand to the camera coordinates.

Let's consider a camera attached to a robotic hand, as shown in the following diagram:



The interest of this package is to perform the hand to eye calibration part.

To do so, two sets of transformations are fed to the `calibrator` node.

- a list of camera-to-object transformations
- a list of world to hand transformations

Once the node has enough transformations, it can estimate the camera-to-hand transformation which is the goal of the calibration.

The package consists of a `calibrator` node and an experimental client doing a sample calibration from a few poses.

The experimental client is merely there for a quick demonstration of the package abilities.

2. Usage

To run the `calibrator` node:

```
rosrun visp_hand2eye_calibration visp_hand2eye_calibration_calibrator
```

3. Example

An example client is shipped with this package. It feeds different transformations (camera->object and world->hand) to the calibrator which computes the relative transformation between the the camera and the hand.

To try this example, run roscore first.

```
roscore
```

Then, run the client:

```
rosrun visp_hand2eye_calibration visp_hand2eye_calibration_client
```

Finally, run the calibrator:

```
rosrun visp_hand2eye_calibration visp_hand2eye_calibration_calibrator
```

The output displayed in the terminal running your client should look like this:

```
[ INFO] [1329226083.184531090]: Waiting for topics...
[ INFO] [1329226084.186233704]: 1) GROUND TRUTH:
[ INFO] [1329226084.186327570]: hand to eye transformation:
translation:
  x: 0.1
  y: 0.2
  z: 0.3
rotation:
  x: 0.96875
  y: 0.0863555
  z: -0.0863555
  w: 0.215889


[ INFO] [1329226085.186682976]: 2) QUICK SERVICE:
[ INFO] [1329226085.188853282]: hand_camera:
translation:
  x: 0.1
  y: 0.2
  z: 0.3
rotation:
  x: 0.96875
  y: 0.0863555
  z: -0.0863555
  w: 0.215889

[ INFO] [1329226085.188915366]: 3) TOPIC STREAM:
[ INFO] [1329226085.190303537]: hand_camera:
translation:
  x: 0.1
  y: 0.2
  z: 0.3
rotation:
  x: 0.96875
  y: 0.0863555
  z: -0.0863555
  w: 0.215889
```

4. Nodes

4.1 calibrator

4.1.1 Subscribed Topics

`world_effector` (visp_hand2eye_calibration/TransformArray (http://docs.ros.org/api/visp_hand2eye_calibration/html/msg/TransformArray.html))

transformation between the world and the hand frame. The node expects to receive several of those transformations.

`camera_object` (visp_hand2eye_calibration/TransformArray (http://docs.ros.org/api/visp_hand2eye_calibration/html/msg/TransformArray.html))

transformation between the camera and the object frame. The node expects to receive several of those transformations.

4.1.2 Services Called

`compute_effector_camera` (visp_hand2eye_calibration/compute_effector_camera (http://docs.ros.org/api/visp_hand2eye_calibration/html/srv/compute_effector_camera.html))

Returns the hand to camera transformation result after calibration based on the data published on the subscribed topics.

`compute_effector_camera_quick` (visp_hand2eye_calibration/compute_effector_camera_quick (http://docs.ros.org/api/visp_hand2eye_calibration/html/srv/compute_effector_camera_quick.html))

Returns the hand to camera transformation result after calibration. This service is not based on the data published on the subscribed topics. Instead, the transformation are passed as service parameters.

`reset` (visp_hand2eye_calibration/reset (http://docs.ros.org/api/visp_hand2eye_calibration/html/srv/reset.html))

Reset all the data published on the subscribed topics. Only new publications (transformations) will be taken into account.

Wiki: visp_hand2eye_calibration/Electric (última edição 2012-02-22 13:19:45 efectuada por ThomasMoulard (/ThomasMoulard))