Intership report

Context

As a third year of engineering school, students have to perform a three months internship abroad to get involved in a foreign culture and to use the skills they learnt at school.

The company choosen for that intership, Shadow Robot Company LTD, is a research company which is working on a robotic human hand, the Shadow Hand. The theme given by the company was closely related to what had been learnt at school, since it is pure software engineering.

Objectives

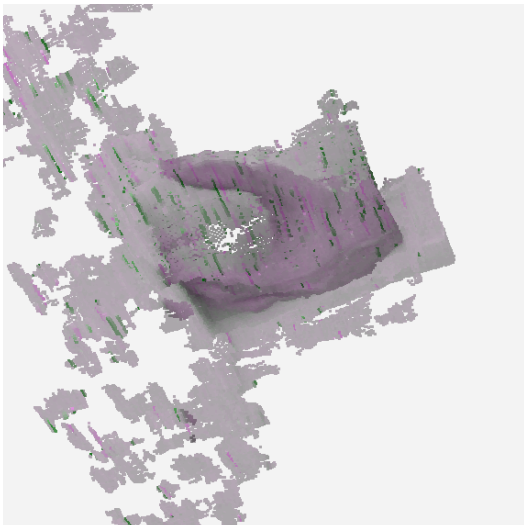
The first objective of the internship was to get a 3D vision of a human hand. Another one was to be able to identify the fingers on the 3D image stream, to send information to the robotic hand and make it follow the user's hand.

The evolution of the company, and the requirements of some clients put these objectives a bit to the background of the internship. They needed a new control interface for the hand, which should be done by a software engineer.

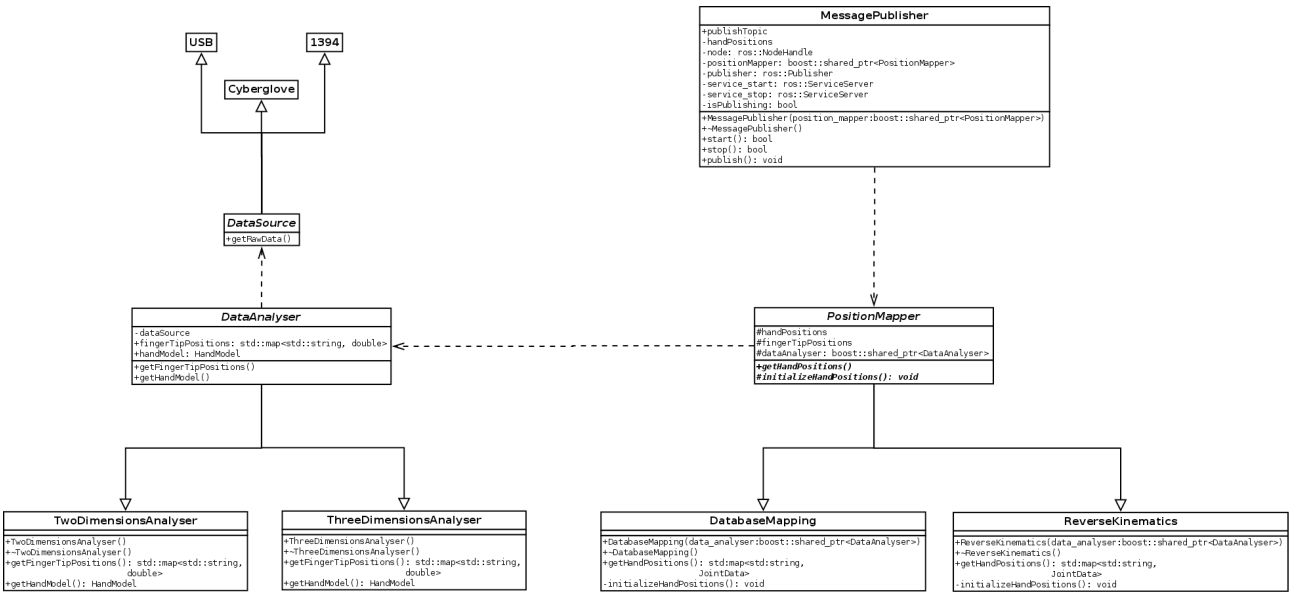
What has been done

For the most important part of the internship time, the focus has been on the graphic control interface. Some screenshots are enclosed to this report, as the documentation that has been written for it, which describes the functions and explains how to use the interface.

About the 3D vision, what has been done is more like a proof of concept. As it can been seen on the screenshots, a 3D stream is displayed on the screen. But the quality of the cameras used (which are two low-cost firewire webcams) is too bad to get a really good and usable stream. That stream is obtained using a block-matching algorithm provided by Open-CV, which needs a good pair of pictures with an adjusted colors. Actually, performing stereo vision is often done with pre-calibrated cameras, which are much more expansive than simple webcams. That's why even if the 3D vision is done and easy to feel with human eyes / brain, it is not usable yet for a computer processing.

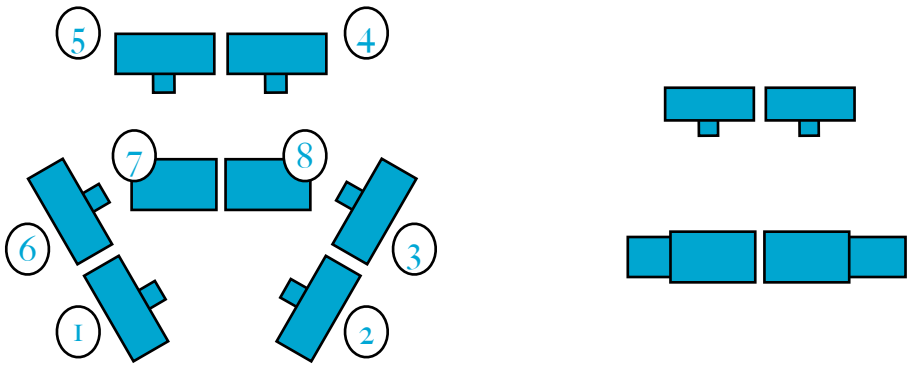


Even if the first goal was to design an optical dataglove, a more generic architecture has been put in place. Only the skeleton of the code is developped yet, but the functions are ready to implement, and the classes are communicating between them. For example, the dataglove could be implemented using some USB webcams, IEE1394 ones, of even a Cyberglove. The same generic idea is used for the mapping, which could be done with reverse kinematics of with a very smart database. See the enclosed UML diagram for further details :



What remains to be done

Regarding the 3D vision, the exploitation of the stream is not started at all. Even the acquisition is not over yet. Two cameras are not enough to get a good 3D points cloud, and the goal would be to use 4 pairs of cameras (views from top and front) :



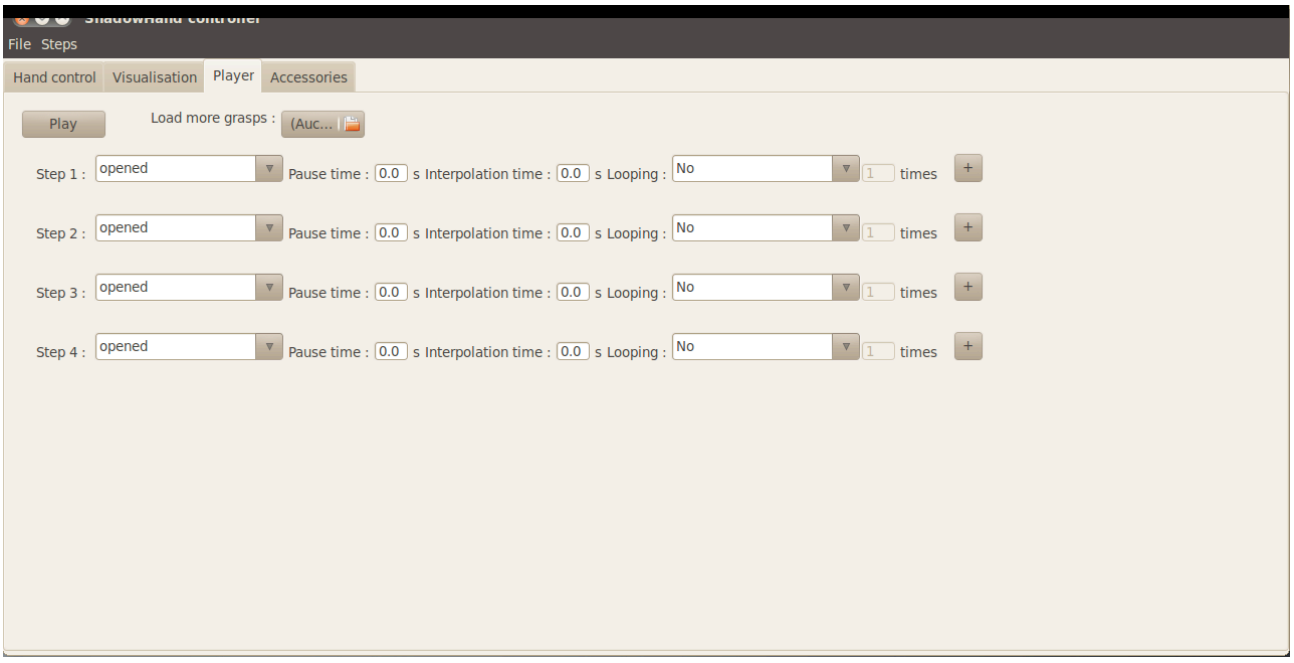
The good way to do that would be :

- Calibrate 2-3, 4-5, 6-1, 7-8 together as pairs and save calibration files as backups.
- Calibrate 1-2, 3-4, 5-6 together as pairs.
- Write a node to acquire images and publish them through ROS as raw images.
- Write another node to remap the raw images using the first calibration files.

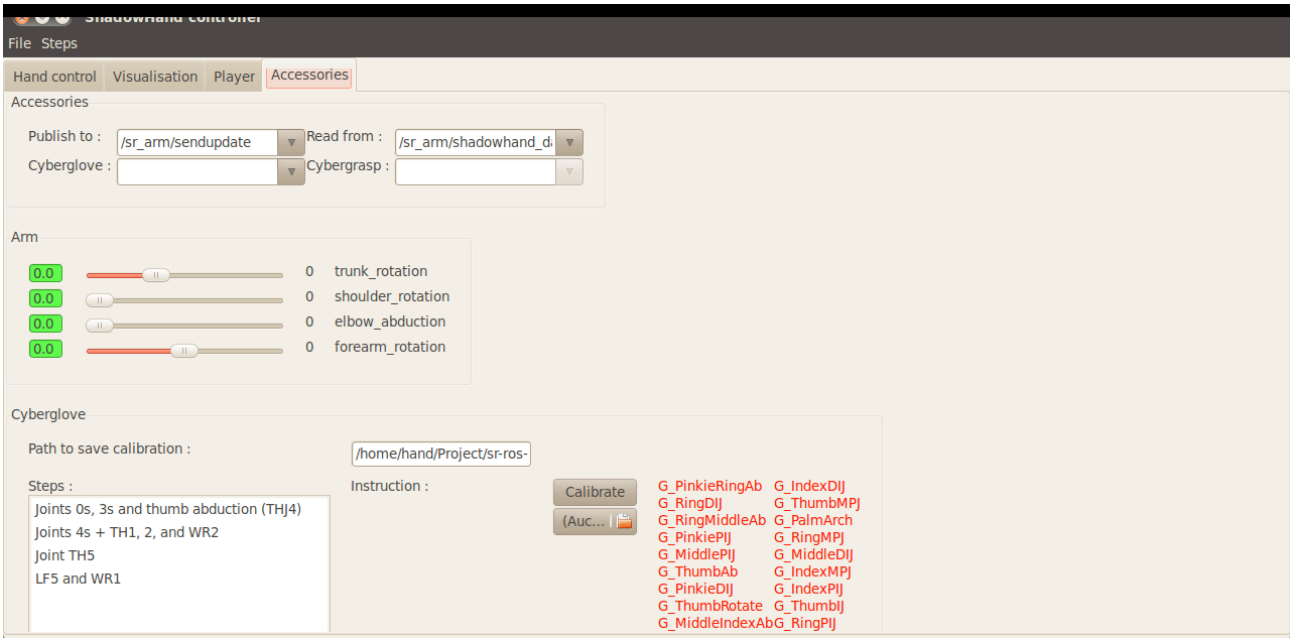
- Write a third node to remap the raw images using the second calibration files.
- Write a node to publish the tf_transformations to make the cloud points superposing at images intersections.
- Write a node to merge the three corrected points cloud to a new one.
- Finally, use PCL to work with the points cloud to identify the fingertips.

Screenshots

The main control interface : it is possible to move each joint separately, many joints together, go from a grasp to another (using a linear interpolation) and save grasps.



The step player interface : it is possible to play a sequence of saved grasps, specifying the time to spend on each grasp (interpolation time and pause time after interpolation) with a looping feature.



The accessories interface : it is possible to change the listening and publishing topics, and accessories are added dynamically, depending on their presence. At this time, the cyberglove is supported, with a calibration feature, and with the possibility of switching the calibration file without restarting the program. The Shadow Arm is also supported, with sliders to move the joints.

Extra : a (small) overview of Blender

During the internship period, a client asked for an old version of the Shadow Hand, which was using air muscles instead of motors. That version was not set up to be used in Rviz, which is the ROS viewer for many things, especially the hand simulator. The part that was breaking the compatibility was that there were no mesh adapted to its dimensions. That problem was solved using Blender :