



Assignment 1

VISION BASED NAVIGATION

EXERCISE 1

a) What is a ROS node?

A node is a process that performs computation. Nodes are combined together into a graph and communicate with one another using streaming topics, RPC services, and the Parameter Server. A robot control system will usually comprise many nodes and each node can execute a separate task independently. E.g. A node can subscribe to a command (e.g. information about the next position of joints of robotic arm) and then to perform robotic movement accordingly.

A ROS node is written with the use of a ROS client library, such as roscpp or rospy.

b) What is the difference between ROS messages, services, and parameters?

Messages: Nodes communicate with each other by passing messages in a publisher/subscriber fashion. A message is simply a data structure, comprising typed fields. Standard primitive types (integer, floating point, Boolean, etc.) are supported, as are arrays of primitive types. Messages can also include arbitrarily nested structures and arrays.

Services: Services establish a Request / response system, which is defined by a pair of messages: one for the request and one for the response. A providing ROS node offers a service under a string name, and a client calls the service by sending the request message and awaiting the reply. Client libraries usually present this interaction to the programmer as if it were a remote procedure call.

Parameters: Parameters are globally visible data. They are stored in a shared dictionary called the Parameter server. The Parameter Server allows data to be stored by key in a central location. It helps to store the configuration

c) What does TF stand for, and what is a TF tree? What do you need to implement in your node to get relative poses between two frames at a specific time through TF?

tf is a transformation package. It helps user keep track of multiple coordinate frames over time. A robotic system may typically have many coordinate frames, such as the base, gripper, shoulder etc. and tf helps in maintaining their relationships. It maintains the relationship between coordinate frames in a tree structure buffered in time, and lets the user transform points, vectors, etc between any two coordinate frames at any desired point in time.

tf::TransformListener can do help getting the relative poses between two frames by calling **lookupTransform** with identical target- and source-frame, but with different target- and source-times.

```

Tf::TransformListener listener;
ros::Rate rate(5.0);
while(node.ok())
{
    tf::StampedTransform transform;
    try{
        listener.lookupTransform(<Frame1> , <Frame2> , ros::Time(0),transform);
    }
    catch(tf::TransformException e)
    {
        ROS_ERROR( "%s " , ex.what ( ) );
    }
}

```

d) What are the possibilities to visualize point clouds using RViz? Give at least two.

Point-Cloud2: to visualize point clouds.

Depth Cloud: to visualize points clouds with depth information.

EXERCISE 2

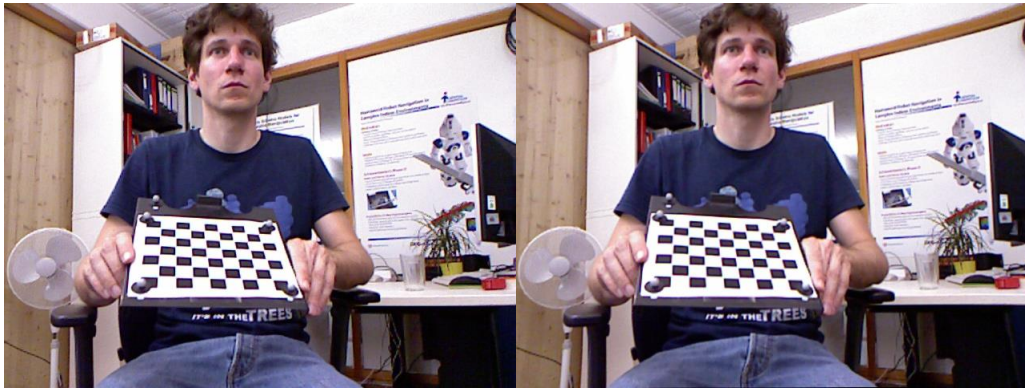
a) Use ROS camera calibration to obtain the intrinsic parameters of the RGB camera on both datasets. Compare your results with the calibration results given on this website.

The main difference in our calibration is the distortion parameter d_4 . It is 0, while the parameter given by the website is 1.163314 for Freiburg1 and 0.917205 for Freiburg2. That means that our radial distortion only depends on r^2 and r^4 , but not on r^6 . The website probably fitted a polynomial of sixth order, while we only used a polynomial of fourth order.

The translational distortion (d_2 and d_3) are only slightly different as well as the camera intrinsics.

b) What are your observations of the effect of the differences between both parameter sets?

The main difference can be seen at the borders of the images, e.g. the wooden strip in the top right corner is curved differently depending on the parameters. This is probably due to missing calibration points, because the checkerboard is mostly moved through the image center.



Left: Input image , Right: After distrotrion correction



Distortion correction (Website output)