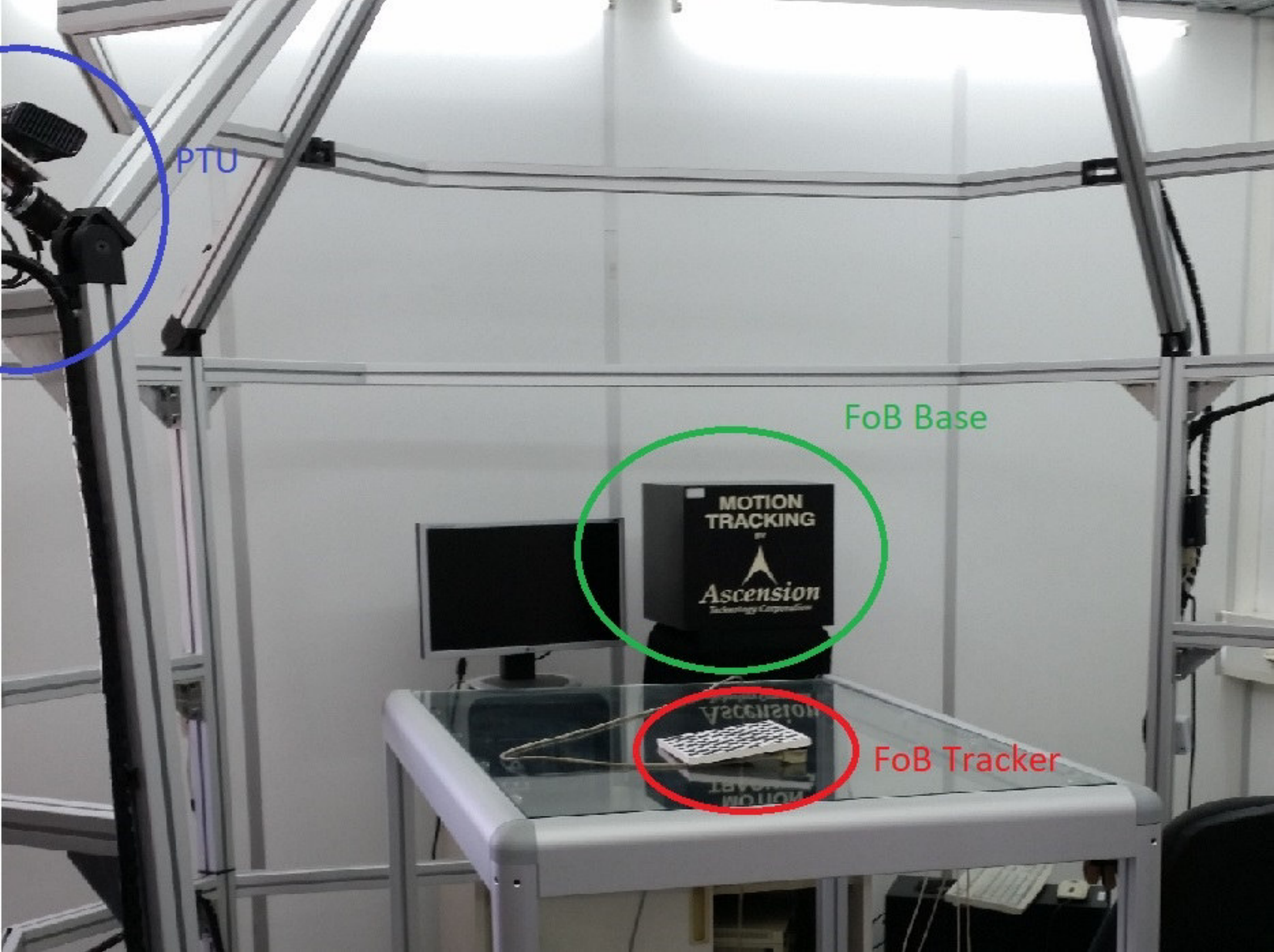Picture of Pdb dome

PTU

FoB Base

FoB Tracker

Kinematic chain of Pbd-dome with FoB and PTU in rviz

PTU

FoB Base
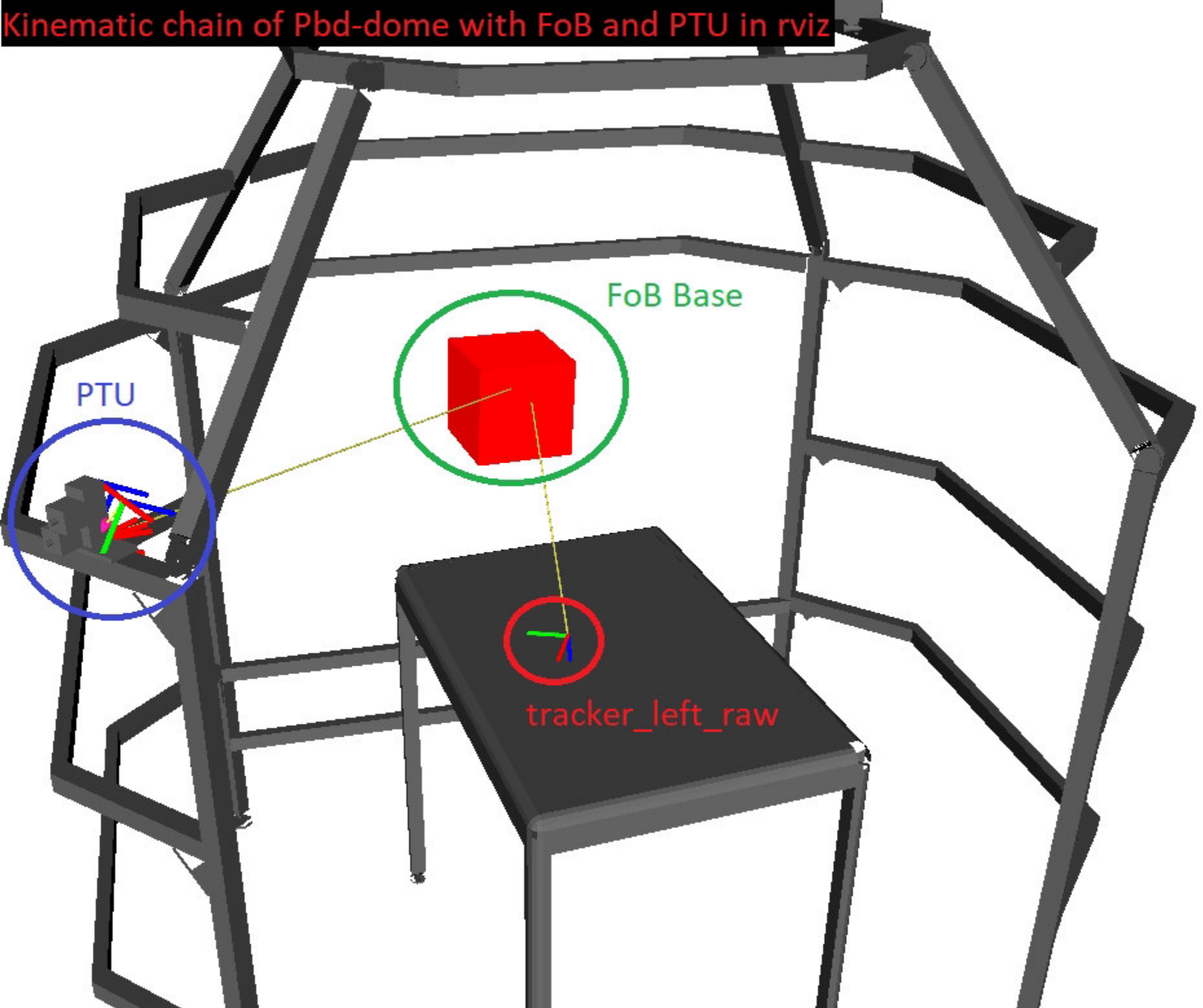
tracker_left_raw

Inaccurate transformation due to incorrect kinematic chain -> calibration is needed

tracker_right_raw
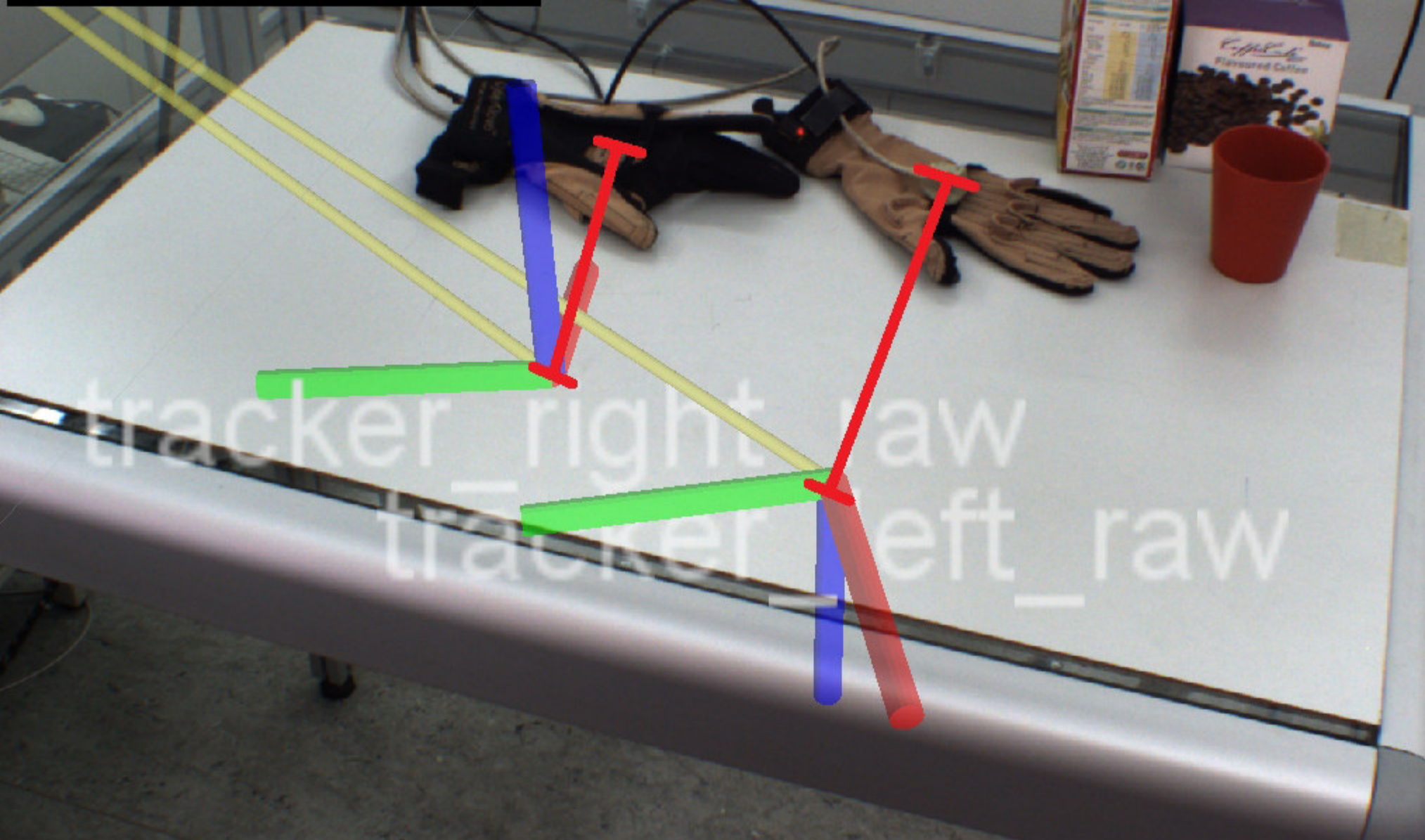
tracker_left_raw

1. Transformation of FoB tracker into camera frame
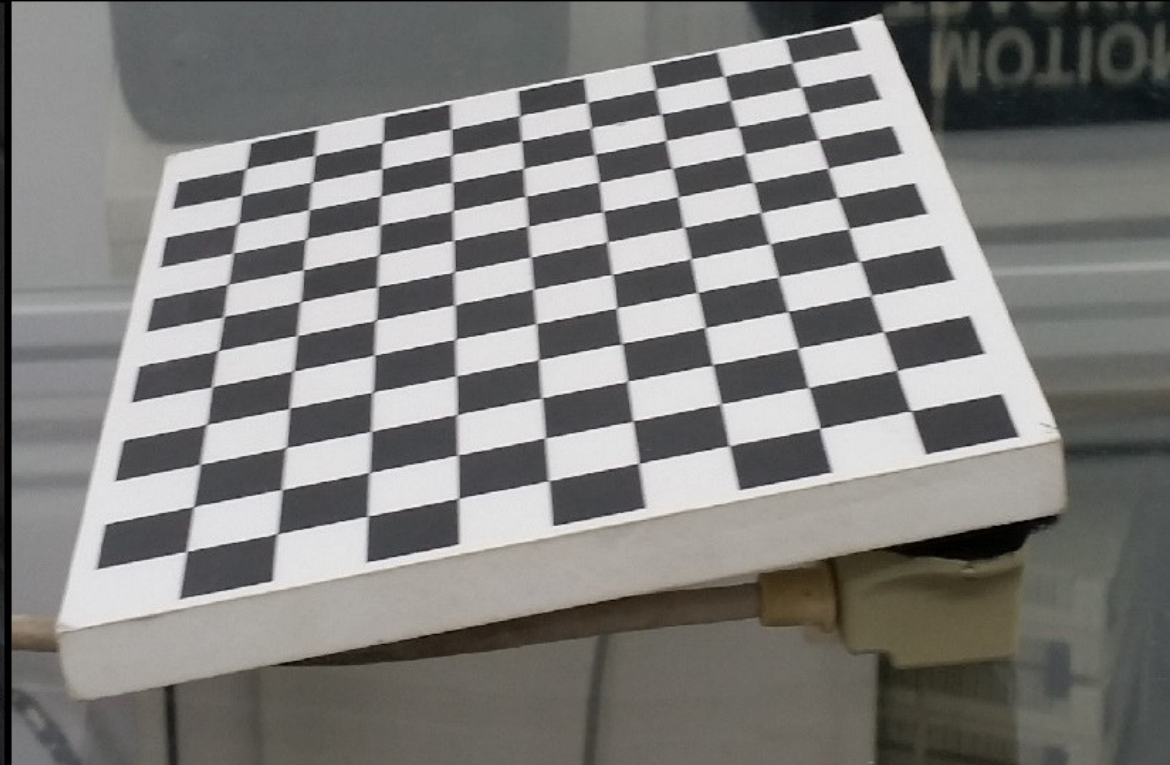
2. Checkerboard for detection in cameraframe ("/camera_left_frame") and FoB tracker in "/tracker_left_raw"

3. Calibration Setup

Checkerboard Detector

PTU Controller

Checkerboard
with FoB Tracker

Calibration data
Recorder

4. Record data (>100 sets) in area of calibration (e. g. calibration tool dome)

area of working = area of calibration

```xml
<!--Frames.xml-->
<Data>
<Frames>
<!--Base of kinematic chain-->
<Frame name="fob_base" base="" a="0" b="0" g="0" x="0" y="0" z="0" />

<!--helper frame for calibration. This frame is identical as root, but has margin for calibration-->
<Frame name="ptu_base_offset" base="fob_base" a="0" b="0" g="0" x="0" y="0" z="0" min_x="-1000" max_x="1000" min_y="-1000" max_y="1000" min_z="-1000" max_z="1000"/>

<!--real TF from FoBBase (with calibration offset) to PTU-->
<Frame name="ptu_base" base="ptu_base_offset" a="0" b="0" g="0" x="923.776+683" y="332+873" z="158.5-549.5-95.0"/>
<Frame name="ptu" base="ptu_base" a="0" b="0" g="-45" y="0" x=" 0" z="0 " />

<!--helper frame for PTU pan input (Variable and data=true)-->
<Frame name="ptu_pan" base="ptu" a="0" b="0" g="-([1])" x="0" y="0" z="0" data="true" />

<!--helper frame for PTU tilt input (Variable and data=true)-->
<Frame name="ptu_tilt" base="ptu_pan" a="([2])" b="0" g="0" x="0" y="0" z="0" data="true"/>

<!--TF from PTU to camera-->
<Frame name="camera_left_frame" base="ptu_tilt" a="90" b="0" g="0" x="-70" y="-37" z="-128+95" />

<!--Datapoint of checkerboard in camera frame (Variables and data=true)-->
<Frame name="checkerboard_camera" base="camera_left_frame" x="([10])*1000" y="([11])*1000" z="([12])*1000" data="true" />

<!--Datapoint of fob tracker in fob_base frame (Variables and data=true)-->
<Frame name="tracker_left_raw" base="fob_base" x="([3])*1000" y="([4])*1000" z="([5])*1000" qw="([6])" qx="([7])" qy="([8])" qz="([9])" data="true" />

<!--Offset from tracker to checkerboard point-->
<Frame name="checkerboard_tracker" base="tracker_left_raw" a="0" b="0" g="0" x="0" y="0" z="20" />
</Frames>

<Goal>
<!--Match both checkerboard points (they are the same points in real)-->
<Position first="checkerboard_camera" second="checkerboard_tracker"/>
<Orientation first="checkerboard_camera" second="checkerboard_tracker"/>
</Goal>
</Data>
```

Optimization Result:

```
vatue: 0.686734 X: 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124
Value: 0.686734 X: 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124
Value: 0.686734 X: 0.331139 0.497854 0.671096 0.498129 0.766122 0.499124
Value: 0.686734 X: 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124
Value: 0.686734 X: 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124
Value: 0.686734 X: 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124
Value: 0.686734 X: 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124
Best: 0.686734
Result:
0.331140
0.497854
0.671095
0.498129
0.766122
0.499124
0.686734 0.331140 0.497854 0.671095 0.498129 0.766122 0.499124 0.686734
Final result:
Result: <Frame name="ptu_base_offset" base="fob_base" a="-158.165" b="88.3969" g="-157.948" x="-0.101316" y="-0.00128745" z="0.102657" />
Position (xyz): -0.101316 -0.001287 0.102657 Quaternion (wxyz): 0.697861 -0.005036 0.716212 -0.002358 EulerAngles(ZXZ): 90.209261 91.488106 -90.  ..9/
```

⬅ new optimal Frame

```
<-- result -->
```
This can be used for any static_transformation_publisher or URDF

```
Showing final result:
Value: 0.686734
Values: 0.33114 0.497854 0.671095 0.498129 0.766122 0.499124
Name: ptu_base_offset Position (xyz): -0.101316 -0.001287 0.102657 Quaternion (wxyz): 0.697861 -0.005036 0.716212 -0.002358
[kinematic chain optimizer-1] process has finished cleanly
```

```
<!--after_calibration.launch -->
<launch>
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_0" args="-0.101316 -0.00128745 0.102657 -0.005036 0.716212 -0.002358 -0.697861 fob_base ptu_base_offset 100"/>
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_2" args="1.606780 1205 -0.486000 0.000000 0.000000 0.000000 1.000000 ptu_base_offset ptu_base 100"/>
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_3" args="0.000000 0.000000 0.000000 0.000000 0.000000 0.923880 -0.382683 ptu_base ptu 100"/>
# input: PTUPan
# input: PTUTilt
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_7" args="-70.000000 -37.000000 -33.000000 0.707107 0.000000 0.000000 0.707107 CameraRot Camera 100"/>
</launch>




<!--before_calibration.launch -->
<launch>
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_0" args="0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 1.000000 fob_base ptu_base_offset 100"/>
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_2" args="1.606780 1205 -0.486000 0.000000 0.000000 0.000000 1.000000 ptu_base_offset ptu_base 100"/>
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_3" args="0.000000 0.000000 0.000000 0.000000 0.000000 0.923880 -0.382683 ptu_base ptu 100"/>
# input: PTUPan
# input: PTUTilt
<node pkg="tf" type="static_transform_publisher" name="calibration_publisher_7" args="-70.000000 -37.000000 -33.000000 0.707107 0.000000 0.000000 0.707107 CameraRot Camera 100"/>
</launch>
```