Here we introduce how to calibrate cameras using several commonly used models and covert to svo format.

Pinhole projection + Radial-tangential

This is the distortion model used in opencv and ROS, also known as `plumb_bob`. We can calibrate it using the tool provided by ROS:

```
sudo apt-get install ros-indigo-camera-calibration
rosrun camera_calibration cameracalibrator.py <specify topics/calibration target>
```

Make sure to adapt size for the checkerboard actually used. What you get is in the format:

```
camera matrix
fx 0 cx
0 fy cy
0 0 1

distortion
d0 d1 d2 d3 0

.........
```

For use with svo, copy the values to the following template (values with $ prefix have to be filled in):

```
cameras:
- camera:
    distortion:
      parameters:
        cols: 1
        rows: 4
        data: [$d0, $d1 , $d2 , $d3]
      type: radial-tangential
    image_height: $image_height
    image_width: $image_width
    intrinsics:
      cols: 1
      rows: 4
      data: [$fx, $fy, $cx, $cy]
    label: cam0
    line-delay-nanoseconds: 0
    type: pinhole
  T_B_C:
    cols: 4
    rows: 4
    data: [ 1., 0., 0., 0.,
            0., 1., 0., 0.,
            0., 0., 1., 0.,
            0., 0., 0., 1.]
  serial_no: 0
  calib_date: 0
  description: '$camera_name'
label: $camera_name
```

`T_B_C` is the pose of the camera frame in the IMU frame. This is used when SVO is set to use the IMU.

Pinhole projection + Equidistant

This is a generic distortion model that can model very different field of views (paper), therefore we can use it for pinhole as well as fisheye cameras. OpenCV (from 3.0) also supports this model . To calibrate a camera using a equidistant camera model, we can use Kalibr . For details of Kalibr calibration, please refer to this official manual .

Afterwards, you can use the script `kalibr_to_svo.py` under `svo_ros/scripts` to convert the output to svo format:

```
./kalibr_to_svo --kalibr <output_of_kalibr>
```

Omnidirectional

This is a special model that combines projection and distortion together. It works for fisheye as well as catadioptric cameras. To use this camera model, you need to calibrate the camera using this Matlab Toolbox . Please refer to the page of the toolbox for details.

Afterwards, you can use the script `omni_matlab_to_rpg.py` under `svo_ros/scripts` to convert the output to svo format.