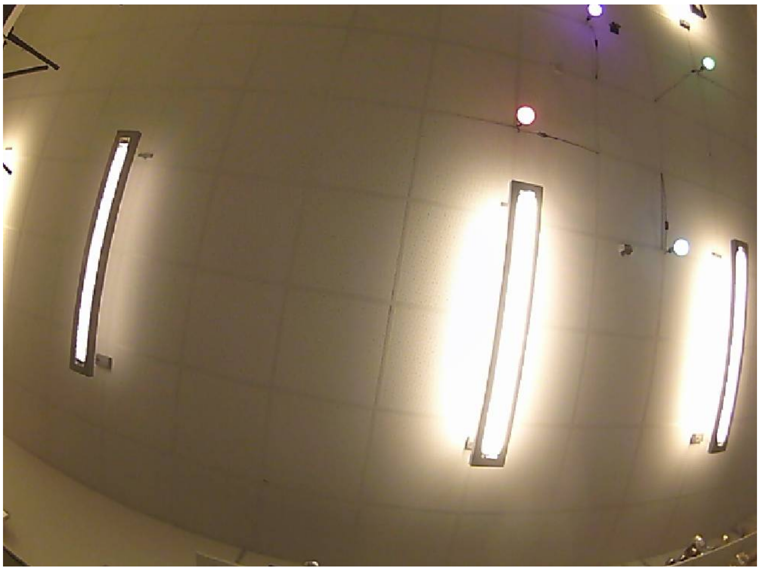
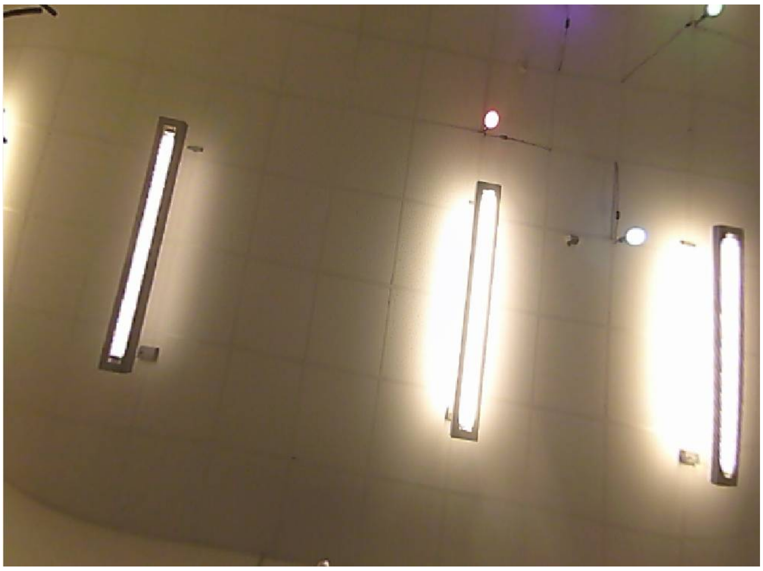
All Code can be found at: https://github.com/chbrock/robotik_ws1718

Aufgabe 1

distorted:



undistorded:



Aufgabe 2

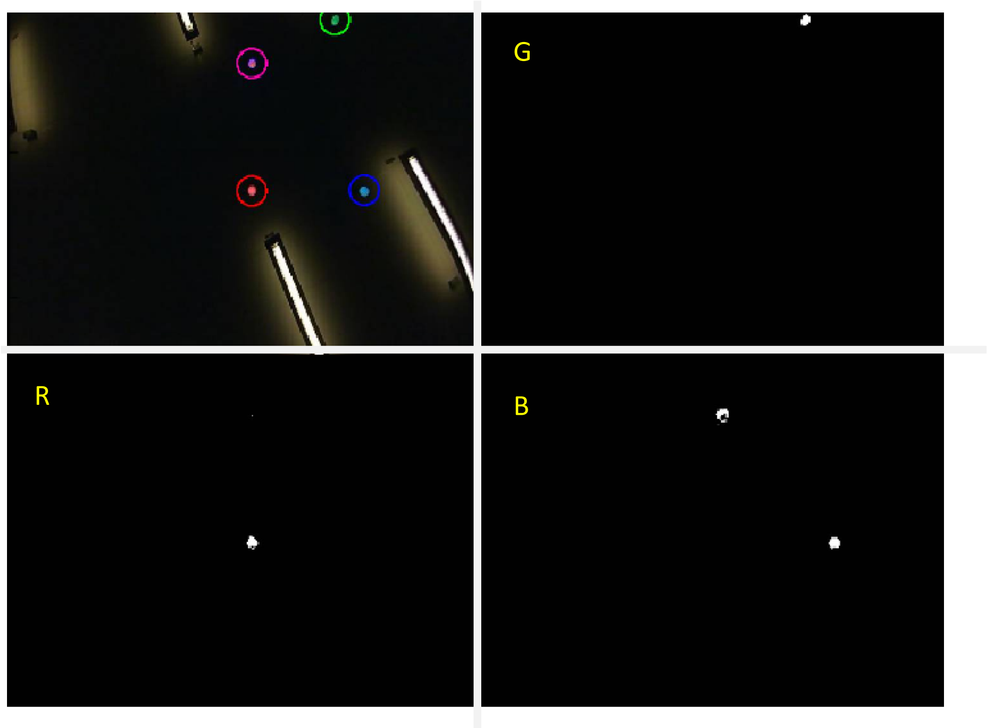
We used cv2.inRange(img, lower, upper). assembling the parametersets for the three colors as follows. Since blue and purple were very similar in the provided bag-file, these were extracted from the same separation.

```
r = 190; r_s = 20
b = 130; b_s = 40
g = 60;  g_s = 20

s_low = 150; s_high = 255
v_low = 30; v_high = 255

lower_r = np.array([r-r_s,s_low,v_low])
upper_r = np.array([r+r_s,s_high,v_high])

lower_b = np.array([b-b_s,s_low,v_low])
upper_b = np.array([b+b_s,s_high,v_high])

lower_g = np.array([g-g_s,s_low,v_low])
upper_g = np.array([g+g_s,s_high,v_high])
```

Aufgabe 3

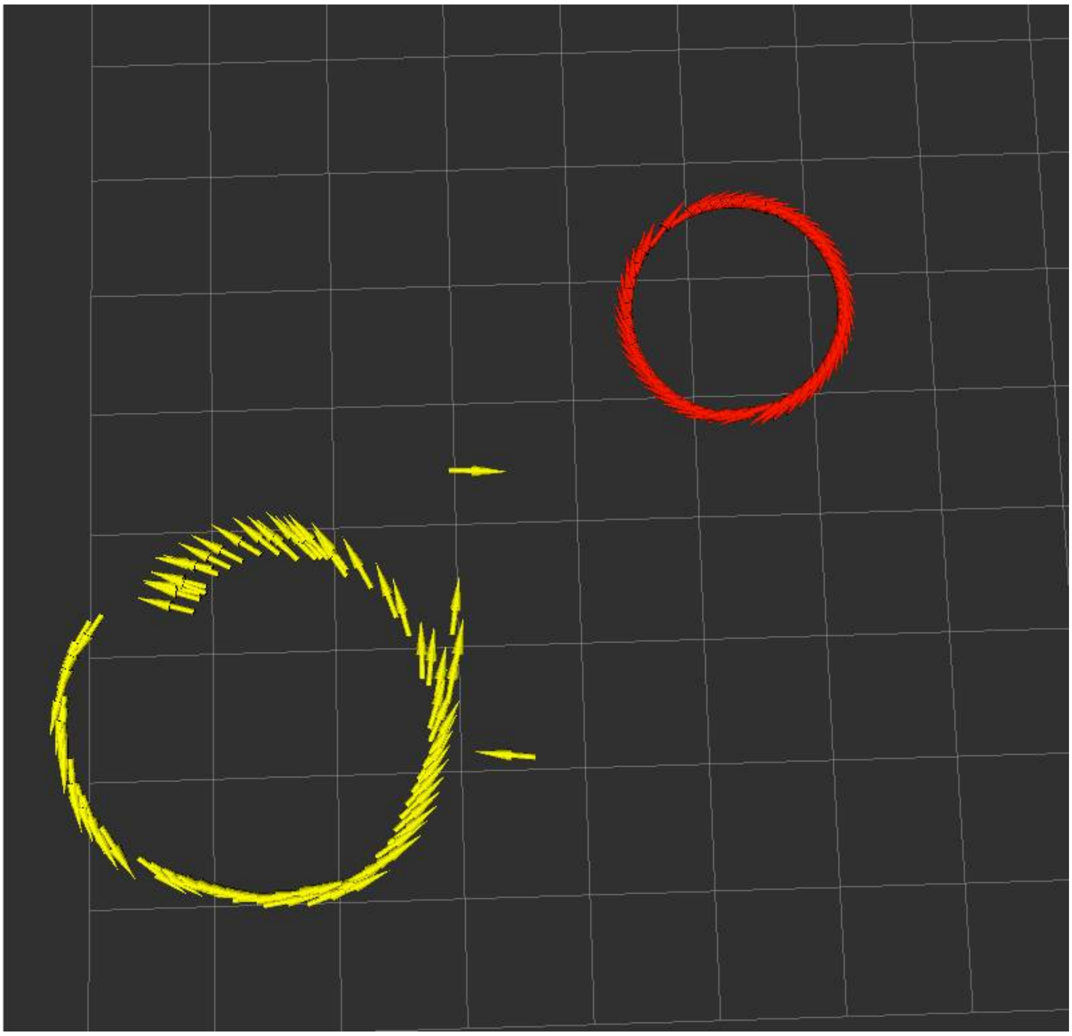
For this task we used all 4 lightbulbs and all available angles. To determine the position a precalculated (XxYx6)-Matrix at a grid of 5cm of all 6 angles in world coordinates was used. For each frame the positions of the bulbs were fitted in pixel coordinates and a vector of camera to bulb angles coordinated. The position was then determined by finding the smallest Euclidean distance of this vector to any of the X,Y positions of the matrix.

Yaw was calculated from the direction of the red to green vector.



```
red:     [ 257.08928571  332.66071429]
green:   [  24.89473684  445.55263158]
blue:    [ 256.93333333  485.24444444]
purple:  [  83.16455696  332.94936709]
pos:     [ 3.65   4.   ]
yaw:     -3.00613310058
```

During the run, we lost the green bulb from view for a couple of frames and did not report coordinates for these.



Positions visualized in RViz looked like this: odom: red odom_gps: yellow. While the yellow coordinates clearly match the left turn and the approximate scale from the video, odom clearly uses a different coordinate system.