Data Analysis

- Read the data files generated by the 5 - Depth Caldulation OFD notebook
- combine the datapoints from the different files with the same rotational and translational speeds
- remove the rotational optical flow
- do descriptive statistics
- calculate distances

Use only the frames around camera angle of 0 degrees

In [78]:

```
user = 'marcvanzyl'
```

In [116…

```
import numpy as np
import cv2, os
from cv2 import aruco
import matplotlib.pyplot as plt
import matplotlib as mpl
import pandas as pd
%matplotlib inline
```

Now we can check

Camera features:

- sensor size = 3.68 x 2.76 mm
- sensor resolution = 3280 × 2464
- focal length = 3.04 mm

$$d_{mm} = \frac{pix \times 3.68}{3280}$$

The depth can now be found

$$Z = \frac{T \times f}{d_{mm}}$$

In [117…

```
import math
def deg_to_rad(deg):
    return deg/90*math.pi()

focal_length = 3.04

pix = math.sin(math.radians(1))*focal_length/3.68*3280
pix
```

Out[117…

```
47.288433442239494
```

In [118…

```
# 5 deg/s
pix*5/21
```

Out[118…

```
11.259150819580832
```

In [119…

```
math.radians(180)
```

Out[119…

```
3.141592653589793
```

Load the list of files

In [202…

```
datadir = "/Users/{}/Google Drive/ScienceFair2021/DataCapture/smooth/".format


data_files = np.array([f for f in os.listdir(datadir) if f.endswith("res3.p")

# just sorts the files according to the number so we match picture 1_A with 1_
#orderR = np.argsort([int((p.split('_')[-1]).split('.')[0]) for p in video_fi
#video_files = video_files[orderR]

data_files.sort()
```

In [203…

```
data_files
```

Out[203…

```
array(['Smooth_15_-2_0_4623HF_res3.p', 'Smooth_15_-2_1_QV4WNZ_res3.p',
       'Smooth_15_-2_2_JB90U4_res3.p', 'Smooth_15_-2_3_MCDNNA_res3.p',
       'Smooth_15_-2_4_VIWO2O_res3.p', 'Smooth_15_-4_0_ESS22N_res3.p',
       'Smooth_15_-4_1_8GD6QO_res3.p', 'Smooth_15_-4_2_FMBNMX_res3.p',
       'Smooth_15_-4_3_EEUONU_res3.p', 'Smooth_15_-4_4_64CXH3_res3.p',
       'Smooth_15_-6_0_2HDNNZ_res3.p', 'Smooth_15_-6_1_D23FV7_res3.p',
       'Smooth_15_-6_2_U8BJY4_res3.p', 'Smooth_15_-6_3_GK86VH_res3.p',
       'Smooth_15_-6_4_JU8DFH_res3.p', 'Smooth_15_0_0_D26LA1_res3.p',
       'Smooth_15_0_1_OOBKBZ_res3.p', 'Smooth_15_0_2_MNBSLT_res3.p',
       'Smooth_15_0_3_VY12DF_res3.p', 'Smooth_15_0_4_46CIDO_res3.p',
       'Smooth_30_-2_0_07HKRJ_res3.p', 'Smooth_30_-2_1_UA678T_res3.p',
       'Smooth_30_-2_2_Z2CEPT_res3.p', 'Smooth_30_-2_3_DPNF63_res3.p',
       'Smooth_30_-2_4_O2G6SZ_res3.p', 'Smooth_30_-4_0_9WKFQF_res3.p',
       'Smooth_30_-4_1_B6QVWT_res3.p', 'Smooth_30_-4_2_9YG7KZ_res3.p',
       'Smooth_30_-4_3_ZZ3YEH_res3.p', 'Smooth_30_-4_4_ARXESZ_res3.p',
       'Smooth_30_-6_0_HOA2A3_res3.p', 'Smooth_30_-6_1_2F304V_res3.p',
       'Smooth_30_-6_2_R5OVIS_res3.p', 'Smooth_30_-6_3_7A1JE7_res3.p',
       'Smooth_30_-6_4_YKIR3Y_res3.p', 'Smooth_30_0_0_TOWEUT_res3.p',
       'Smooth_30_0_1_RB5B21_res3.p', 'Smooth_30_0_2_AAU13M_res3.p',
       'Smooth_30_0_3_STATJY_res3.p', 'Smooth_30_0_4_YKCRBC_res3.p',
       'Smooth_45_-2_0_3P4T66_res3.p', 'Smooth_45_-2_1_JFT8WZ_res3.p',
       'Smooth_45_-2_2_XM1ETY_res3.p', 'Smooth_45_-2_3_BR9Z93_res3.p',
       'Smooth_45_-2_4_OZMRTY_res3.p', 'Smooth_45_-4_0_GNGX25_res3.p',
       'Smooth_45_-4_1_DPV5FD_res3.p', 'Smooth_45_-4_2_BZ37KH_res3.p',
       'Smooth_45_-4_3_GRNIXW_res3.p', 'Smooth_45_-4_4_810HRV_res3.p',
       'Smooth_45_-6_0_R378Z4_res3.p', 'Smooth_45_-6_1_0NECPK_res3.p',
       'Smooth_45_-6_2_U2KIG0_res3.p', 'Smooth_45_-6_3_9UF74H_res3.p',
       'Smooth_45_-6_4_WAKIAQ_res3.p', 'Smooth_45_0_0_V4SN9S_res3.p',
       'Smooth_45_0_1_IGTHOY_res3.p', 'Smooth_45_0_2_B8A482_res3.p',
       'Smooth_45_0_3_9GBKOP_res3.p', 'Smooth_45_0_4_SLPA8L_res3.p',
       'Smooth_60_-2_0_EENTSE_res3.p', 'Smooth_60_-2_1_TOLJ0V_res3.p',
       'Smooth_60_-2_2_QJP3WV_res3.p', 'Smooth_60_-2_3_C9X5WP_res3.p',
       'Smooth_60_-2_4_UX8ITL_res3.p', 'Smooth_60_-4_0_EVQ115_res3.p',
       'Smooth_60_-4_1_9CNJ4F_res3.p', 'Smooth_60_-4_2_XJTK3X_res3.p',
       'Smooth_60_-4_3_DDB2LK_res3.p', 'Smooth_60_-4_4_PPJSIF_res3.p',
       'Smooth_60_-6_0_EO5Q0F_res3.p', 'Smooth_60_-6_1_ITFYF8_res3.p',
       'Smooth_60_-6_2_P77OMH_res3.p', 'Smooth_60_-6_3_HY8MRS_res3.p',
       'Smooth_60_-6_4_ONJJEZ_res3.p', 'Smooth_60_0_0_6KASLJ_res3.p',
       'Smooth_60_0_1_H2CBPV_res3.p', 'Smooth_60_0_2_HI1FZ4_res3.p',
       'Smooth_60_0_3_0IMBF0_res3.p', 'Smooth_60_0_4_XV9Y0W_res3.p'],
      dtype='<U28')
```

In [204…

```python
import pickle
```

In [205…

```python
file = data_files[0]
df = pickle.load(open('{}{}'.format(datadir,file), 'rb'))
```

In [206…

```
cons_df = pd.DataFrame(columns=['File','FrameNum', 'LinVel', 'RotVel','Run','
                                     'OFC0', 'OFC1', 'OFC2', 'OFC3', 'OFC4',
                                     'OFZ0','OFZ1','OFZ2','OFZ3','OFZ4','OFZ5
for file in data_files:

    # get the features from the filename
    linear_vel = float(file.split('_')[1])
    rotational_vel = float(file.split('_')[2])
    run = int(file.split('_')[3])

    # old file format
    #temp = file.split('_')[2].split('-')
    #if temp[0] == '': # negative rotation
    #    rotational_vel = -1*float(temp[1])
    #    run = int(temp[2])
    #else:
    #    rotational_vel = float(temp[0])
    #    run = int(temp[1])

    print('{}: lin: {} rot: {} run:{}'.format(file, linear_vel, rotational_ve

    df = pickle.load(open('{}{}'.format(datadir,file), 'rb'))

    df['File'] = file
    df['LinVel'] = linear_vel
    df['RotVel'] = rotational_vel
    df['Run'] = run
    df['FrameNum'] = df.index

    cons_df = cons_df.append(df, ignore_index=True)
```

```
Smooth_15_-2_0_4623HF_res3.p: lin: 15.0 rot: -2.0 run:0
Smooth_15_-2_1_QV4WNZ_res3.p: lin: 15.0 rot: -2.0 run:1
Smooth_15_-2_2_JB90U4_res3.p: lin: 15.0 rot: -2.0 run:2
Smooth_15_-2_3_MCDNNA_res3.p: lin: 15.0 rot: -2.0 run:3
Smooth_15_-2_4_VIWO2O_res3.p: lin: 15.0 rot: -2.0 run:4
Smooth_15_-4_0_ESS22N_res3.p: lin: 15.0 rot: -4.0 run:0
Smooth_15_-4_1_8GD6QO_res3.p: lin: 15.0 rot: -4.0 run:1
Smooth_15_-4_2_FMBNMX_res3.p: lin: 15.0 rot: -4.0 run:2
Smooth_15_-4_3_EEUONU_res3.p: lin: 15.0 rot: -4.0 run:3
Smooth_15_-4_4_64CXH3_res3.p: lin: 15.0 rot: -4.0 run:4
Smooth_15_-6_0_2HDNNZ_res3.p: lin: 15.0 rot: -6.0 run:0
Smooth_15_-6_1_D23FV7_res3.p: lin: 15.0 rot: -6.0 run:1
Smooth_15_-6_2_U8BJY4_res3.p: lin: 15.0 rot: -6.0 run:2
Smooth_15_-6_3_GK86VH_res3.p: lin: 15.0 rot: -6.0 run:3
Smooth_15_-6_4_JU8DFH_res3.p: lin: 15.0 rot: -6.0 run:4
Smooth_15_0_0_D26LA1_res3.p: lin: 15.0 rot: 0.0 run:0
Smooth_15_0_1_OOBKBZ_res3.p: lin: 15.0 rot: 0.0 run:1
Smooth_15_0_2_MNBSLT_res3.p: lin: 15.0 rot: 0.0 run:2
Smooth_15_0_3_VY12DF_res3.p: lin: 15.0 rot: 0.0 run:3
Smooth_15_0_4_46CIDO_res3.p: lin: 15.0 rot: 0.0 run:4
Smooth_30_-2_0_07HKRJ_res3.p: lin: 30.0 rot: -2.0 run:0
Smooth_30_-2_1_UA678T_res3.p: lin: 30.0 rot: -2.0 run:1
Smooth_30_-2_2_Z2CEPT_res3.p: lin: 30.0 rot: -2.0 run:2
Smooth_30_-2_3_DPNF63_res3.p: lin: 30.0 rot: -2.0 run:3
Smooth_30_-2_4_O2G6SZ_res3.p: lin: 30.0 rot: -2.0 run:4
Smooth_30_-4_0_9WKFQF_res3.p: lin: 30.0 rot: -4.0 run:0
Smooth_30_-4_1_B6QVWT_res3.p: lin: 30.0 rot: -4.0 run:1
```

```
Smooth_30_-4_2_9YG7KZ_res3.p: lin: 30.0 rot: -4.0 run:2
Smooth_30_-4_3_ZZ3YEH_res3.p: lin: 30.0 rot: -4.0 run:3
Smooth_30_-4_4_ARXESZ_res3.p: lin: 30.0 rot: -4.0 run:4
Smooth_30_-6_0_HOA2A3_res3.p: lin: 30.0 rot: -6.0 run:0
Smooth_30_-6_1_2F304V_res3.p: lin: 30.0 rot: -6.0 run:1
Smooth_30_-6_2_R5OVIS_res3.p: lin: 30.0 rot: -6.0 run:2
Smooth_30_-6_3_7A1JE7_res3.p: lin: 30.0 rot: -6.0 run:3
Smooth_30_-6_4_YKIR3Y_res3.p: lin: 30.0 rot: -6.0 run:4
Smooth_30_0_0_TOWEUT_res3.p: lin: 30.0 rot: 0.0 run:0
Smooth_30_0_1_RB5B21_res3.p: lin: 30.0 rot: 0.0 run:1
Smooth_30_0_2_AAU13M_res3.p: lin: 30.0 rot: 0.0 run:2
Smooth_30_0_3_STATJY_res3.p: lin: 30.0 rot: 0.0 run:3
Smooth_30_0_4_YKCRBC_res3.p: lin: 30.0 rot: 0.0 run:4
Smooth_45_-2_0_3P4T66_res3.p: lin: 45.0 rot: -2.0 run:0
Smooth_45_-2_1_JFT8WZ_res3.p: lin: 45.0 rot: -2.0 run:1
Smooth_45_-2_2_XM1ETY_res3.p: lin: 45.0 rot: -2.0 run:2
Smooth_45_-2_3_BR9Z93_res3.p: lin: 45.0 rot: -2.0 run:3
Smooth_45_-2_4_OZMRTY_res3.p: lin: 45.0 rot: -2.0 run:4
Smooth_45_-4_0_GNGX25_res3.p: lin: 45.0 rot: -4.0 run:0
Smooth_45_-4_1_DPV5FD_res3.p: lin: 45.0 rot: -4.0 run:1
Smooth_45_-4_2_BZ37KH_res3.p: lin: 45.0 rot: -4.0 run:2
Smooth_45_-4_3_GRNIXW_res3.p: lin: 45.0 rot: -4.0 run:3
Smooth_45_-4_4_810HRV_res3.p: lin: 45.0 rot: -4.0 run:4
Smooth_45_-6_0_R378Z4_res3.p: lin: 45.0 rot: -6.0 run:0
Smooth_45_-6_1_0NECPK_res3.p: lin: 45.0 rot: -6.0 run:1
Smooth_45_-6_2_U2KIG0_res3.p: lin: 45.0 rot: -6.0 run:2
Smooth_45_-6_3_9UF74H_res3.p: lin: 45.0 rot: -6.0 run:3
Smooth_45_-6_4_WAKIAQ_res3.p: lin: 45.0 rot: -6.0 run:4
Smooth_45_0_0_V4SN9S_res3.p: lin: 45.0 rot: 0.0 run:0
Smooth_45_0_1_IGTHOY_res3.p: lin: 45.0 rot: 0.0 run:1
Smooth_45_0_2_B8A482_res3.p: lin: 45.0 rot: 0.0 run:2
Smooth_45_0_3_9GBKOP_res3.p: lin: 45.0 rot: 0.0 run:3
Smooth_45_0_4_SLPA8L_res3.p: lin: 45.0 rot: 0.0 run:4
Smooth_60_-2_0_EENTSE_res3.p: lin: 60.0 rot: -2.0 run:0
Smooth_60_-2_1_TOLJ0V_res3.p: lin: 60.0 rot: -2.0 run:1
Smooth_60_-2_2_QJP3WV_res3.p: lin: 60.0 rot: -2.0 run:2
Smooth_60_-2_3_C9X5WP_res3.p: lin: 60.0 rot: -2.0 run:3
Smooth_60_-2_4_UX8ITL_res3.p: lin: 60.0 rot: -2.0 run:4
Smooth_60_-4_0_EVQ115_res3.p: lin: 60.0 rot: -4.0 run:0
Smooth_60_-4_1_9CNJ4F_res3.p: lin: 60.0 rot: -4.0 run:1
Smooth_60_-4_2_XJTK3X_res3.p: lin: 60.0 rot: -4.0 run:2
Smooth_60_-4_3_DDB2LK_res3.p: lin: 60.0 rot: -4.0 run:3
Smooth_60_-4_4_PPJSIF_res3.p: lin: 60.0 rot: -4.0 run:4
Smooth_60_-6_0_EO5Q0F_res3.p: lin: 60.0 rot: -6.0 run:0
Smooth_60_-6_1_ITFYF8_res3.p: lin: 60.0 rot: -6.0 run:1
Smooth_60_-6_2_P77OMH_res3.p: lin: 60.0 rot: -6.0 run:2
Smooth_60_-6_3_HY8MRS_res3.p: lin: 60.0 rot: -6.0 run:3
Smooth_60_-6_4_ONJJEZ_res3.p: lin: 60.0 rot: -6.0 run:4
Smooth_60_0_0_6KASLJ_res3.p: lin: 60.0 rot: 0.0 run:0
Smooth_60_0_1_H2CBPV_res3.p: lin: 60.0 rot: 0.0 run:1
Smooth_60_0_2_HI1FZ4_res3.p: lin: 60.0 rot: 0.0 run:2
Smooth_60_0_3_0IMBF0_res3.p: lin: 60.0 rot: 0.0 run:3
Smooth_60_0_4_XV9Y0W_res3.p: lin: 60.0 rot: 0.0 run:4
```

In [207…

```
cons_df.columns
```

Out[207...

```
Index(['File', 'FrameNum', 'LinVel', 'RotVel', 'Run', 'AM0', 'AM1', 'AM2',
       'AM3', 'AM4', 'AM5', 'AM6', 'AM7', 'AM8', 'AM9', 'AM10', 'AM11', 'AM12'
,
       'AM13', 'AM14', 'OFC0', 'OFC1', 'OFC2', 'OFC3', 'OFC4', 'OFC5', 'OFZ0',
       'OFZ1', 'OFZ2', 'OFZ3', 'OFZ4', 'OFZ5', 'OFZ0x', 'OFZ1x', 'OFZ2x',
       'OFZ3x', 'OFZ4x', 'OFZ5x', 'OFZLinear0', 'OFZLinear1', 'OFZLinear2',
       'OFZLinear3', 'OFZLinear4', 'OFZLinear5'],
      dtype='object')
```

In [208...

```
lin_vels = cons_df['LinVel'].unique()
rot_vels = cons_df['RotVel'].unique()
rot_vels
```

Out[208...

```
array([-2., -4., -6.,  0.])
```

In [209...

```
# number of pixels per degree/s of rotation

rotational_coeff = 2.1  #2.10661689

camera_fps = 21
```

In [210...

```
# Actual distances
actual_dist = pd.Series([1550,1560,2020,2030,2575,2580], index=['OFZ0','OFZ1'
actual_dist_linear = pd.Series([1550,1560,2020,2030,2575,2580], index=['OFZLi
```

In [211...

```
actual_df = pd.DataFrame(actual_dist)
actual_df.index.name = 'Zone'
actual_df
```

Out[211...

	Actual Distance [mm]
Zone	
OFZ0	1550
OFZ1	1560
OFZ2	2020
OFZ3	2030
OFZ4	2575
OFZ5	2580

Camera features:

- sensor size = 3.68 x 2.76 mm
- sensor resolution = 3280 × 2464
- focal length = 3.04 mm

$$d_{mm} = \frac{pix \times 3.68}{3280}$$

The depth can now be found

$$Z = \frac{T \times f}{d_{mm}}$$

In [230…

```
def calc_camera_translation(v, frame_rate):
    return v/frame_rate

def calc_depth(pix, step):

    sensor_x = 3.68
    f = 3.1 #3.04
    sensor_x_res = 3280

    d_mm = pix*sensor_x/sensor_x_res

    return step*f/d_mm
```

In [231…

```
calc_depth(5., calc_camera_translation(60, 21))
```

Out[231…

```
1578.8819875776396
```

In [232…

```
calc_depth(3., calc_camera_translation(60, 21))
```

Out[232…

```
2631.469979296066
```

Create all the data tables

result_dict = {} for lin_vel in lin_vels: for rot_vel in rot_vels: arrays = [['n','Raw Flow','Linear Flow', 'Distance', 'Actual Dist', 'Frame Error', 'Frame StdDev','Rolling Window Error', 'Rolling Window Error %','Rolling Window Error StdDev'], ['','[pix/frame]','[pix/frame]', '[mm]', '[mm]', '[mm]', '[mm]', '[mm]', '[%]','[mm]']] full_result_df = pd.DataFrame(columns=pd.MultiIndex.from_arrays(arrays, names=('', 'Zone'))) temp_df = cons_df[(cons_df['LinVel']== lin_vel) & (cons_df['RotVel']==rot_vel)] of_lin = temp_df[['OFZLinear0','OFZLinear1', 'OFZLinear2', 'OFZLinear3', 'OFZLinear4', 'OFZLinear5']].copy() of_raw = temp_df[['OFZ0','OFZ1', 'OFZ2', 'OFZ3', 'OFZ4', 'OFZ5']].copy() col_names =['OFZ0', 'OFZ1', 'OFZ2', 'OFZ3', 'OFZ4', 'OFZ5'] # change the column names of_lin.columns = col_names of_raw.columns = col_names means = of_lin.mean() stds = of_lin.std() num_std_dev =1.0 for col in of_lin.columns: of_lin.loc[:,col] = of_lin[col].apply(lambda x: x if ((x>means[col]-num_std_dev*stds[col]) and (xmeans[col]-num_std_dev*stds[col]) and (x 600] = np.nan #rolling_error[rolling_error < -600] = np.nan summay_stats = summay_stats.append(rolling_error, ignore_index=True) full_result_df[('Rolling Window Error','[mm]')] =

summay_stats.mean().round(2) full_result_df[('Rolling Window Error %','[%]')] = (full_result_df[('Rolling Window Error','[mm]')]/actual_dist*100).round(2) full_result_df[('Rolling Window Error StdDev','[mm]')] = summay_stats.std().round(2) result_dict['{}_{}'.format(lin_vel,rot_vel)] = full_result_df display(full_result_df)

New version that uses only frames when around the camera facing sideways

In [233…

```
frame_x = 3264
mid_frame_x = 3264/2
```

In [234…

```
result_dict = {}

for lin_vel in lin_vels:
    for rot_vel in rot_vels:
        arrays = [['n','Raw Flow','Linear Flow', 'Distance', 'Actual Dist', '
        full_result_df = pd.DataFrame(columns=pd.MultiIndex.from_arrays(array

        temp_df = cons_df[(cons_df['LinVel']== lin_vel) & (cons_df['RotVel']=


        of_lin = temp_df[['OFZLinear0','OFZLinear1', 'OFZLinear2', 'OFZLinear
        of_raw = temp_df[['OFZ0','OFZ1', 'OFZ2', 'OFZ3', 'OFZ4', 'OFZ5']].cop

        col_names =['OFZ0', 'OFZ1', 'OFZ2', 'OFZ3', 'OFZ4', 'OFZ5']

        # change the column names
        of_lin.columns = col_names
        of_raw.columns = col_names


        means = of_lin.mean()
        stds = of_lin.std()

        num_std_dev =1.0

        for col in of_lin.columns:
            of_lin.loc[:,col] = of_lin[col].apply(lambda x: x if ((x>means[co


        depth_res = calc_depth(of_lin, calc_camera_translation(lin_vel, camer


        full_result_df[('n','')] = (of_lin.count())
        full_result_df['Raw Flow'] = of_raw.mean().round(3)
        full_result_df['Linear Flow'] = of_lin.mean().round(3)
        # do the data clean up here on Linear Flow


        full_result_df[('Distance','[mm]')] = calc_depth(of_lin.mean(), calc_
        full_result_df[('Frame StdDev','[mm]')] = (full_result_df[('Distance'
        full_result_df[('Actual Dist','[mm]')] = actual_dist
        full_result_df[('Frame Error','[mm]')] = full_result_df[('Distance','
        #full_result_df[('Error %','[%]')] = (full_result_df[('Error','[mm]')
        print()
        print('Linear velocity: {}mm/s  Rotational velocity: {}deg/s'.format(
        # calc the averaging

        files = temp_df['File'].unique()

        summay_stats = pd.DataFrame(columns=['OFZ0', 'OFZ1', 'OFZ2', 'OFZ3',



        for file in files:
            # grab the file
            full_file_df = temp_df[temp_df['File']==file].copy()

            num_frames = full_file_df.shape[0]
            if rot_vel != 0.0:

                # extract the x pos of Aruco marker 2 to determine the rotati
                full_file_df['AM2x'] = full_file_df['AM2'].apply(lambda x: x[


                mid_frame = full_file_df[full_file_df['AM2x']>0]['FrameNum'].
                max_frame = full_file_df['FrameNum'].max()
                side_frames = max_frame - mid_frame
            else:
                min_frame = full_file_df['FrameNum'].min()
                max_frame = full_file_df['FrameNum'].max()
                mid_frame = int((min_frame+max_frame)/2)
                side_frames = max_frame-mid_frame

            #print('mid_frame: {} side_frames {}'.format(mid_frame, side_fram

            #extract the mid frames
            full_file_df = full_file_df[(full_file_df['FrameNum']>(mid_frame-
                                       (full_file_df['FrameNum']<(mid_frame+s

            file_df = full_file_df[['OFZLinear0','OFZLinear1', 'OFZLinear2',
            file_df.columns = col_names

            # clean the outliers:

            means = file_df.mean()
            stds = file_df.std()

            num_std_dev = 2.0

            for col in file_df.columns:
                file_df.loc[:,col] = file_df[col].apply(lambda x:
                                                        x if ((x>means[col]-n
                                                              and (x<means[co
                                                        else np.nan).copy()

            rolling_measure = file_df.rolling(21, min_periods=10).mean()



            rolling_dist = calc_depth((rolling_measure), calc_camera_translat
                    # calc distance and clean up noise
                    rolling_error = rolling_dist.dropna()-actual_dist

                    #rolling_error[rolling_error > 600] = np.nan
                    #rolling_error[rolling_error < -600] = np.nan


                    summay_stats = summay_stats.append(rolling_error, ignore_index=Tr

                full_result_df[('Rolling Window Error','[mm]')] = summay_stats.mean()
                full_result_df[('Rolling Window Error %','[%]')] = (full_result_df[('
                full_result_df[('Rolling Window Error StdDev','[mm]')] = summay_stats

                result_dict['{}_{}'.format(lin_vel,rot_vel)] = full_result_df

                display(full_result_df)

        pickle.dump(result_dict, open('{}full_analysis_results.p'.format(datadir), 'w
```

```
Linear velocity: 15.0mm/s  Rotational velocity: -2.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Ro Win E Stc
Zone		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[
OFZ0	1345	5.861	1.346	1466.7	1550	-83.3	394.0	-6.46	-0.42	9
OFZ1	1400	5.569	1.302	1515.3	1560	-44.7	407.2	-28.37	-1.82	10
OFZ2	1373	5.650	1.002	1970.2	2020	-49.8	733.4	47.15	2.33	17
OFZ3	1389	5.458	0.977	2019.3	2030	-10.7	744.8	15.50	0.76	17
OFZ4	1361	5.581	0.797	2475.0	2575	-100.0	1173.3	26.61	1.03	27
OFZ5	1382	5.378	0.790	2499.3	2580	-80.7	1163.7	-6.62	-0.26	27

```
Linear velocity: 15.0mm/s  Rotational velocity: -4.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Winc Er Stdl
Zone		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[n
OFZ0	375	10.170	1.407	1402.8	1550	-147.2	424.9	32.23	2.08	105
OFZ1	366	9.694	1.363	1447.9	1560	-112.1	428.7	-4.67	-0.30	125
OFZ2	368	10.134	1.084	1821.3	2020	-198.7	726.0	124.77	6.18	194
OFZ3	367	9.807	1.043	1892.4	2030	-137.6	765.6	49.24	2.43	19
OFZ4	365	10.209	0.892	2212.0	2575	-363.0	1100.4	117.69	4.57	31
OFZ5	369	9.845	0.876	2252.1	2580	-327.9	1105.9	40.60	1.57	31

```
Linear velocity: 15.0mm/s  Rotational velocity: -6.0deg/s
```

Zone	n	Raw Flow [pix/frame]	Linear Flow [pix/frame]	Distance [mm]	Actual Dist [mm]	Frame Error [mm]	Frame StdDev [mm]	Rolling Window Error [mm]	Rolling Window Error % [%]	Roll Wind Er Std [m
OFZ0	211	14.758	1.539	1282.7	1550	-267.3	424.3	-140.87	-9.09	119
OFZ1	219	13.873	1.495	1320.1	1560	-239.9	440.0	-199.62	-12.80	115
OFZ2	222	14.847	1.205	1638.2	2020	-381.8	736.4	-133.81	-6.62	263
OFZ3	225	14.211	1.177	1677.4	2030	-352.6	757.3	-251.28	-12.38	235
OFZ4	223	15.058	1.044	1889.7	2575	-685.3	1015.5	-228.29	-8.87	397
OFZ5	220	14.392	1.024	1927.3	2580	-652.7	990.4	-342.88	-13.29	366

```
Linear velocity: 15.0mm/s  Rotational velocity: 0.0deg/s
```

Zone	n	Raw Flow [pix/frame]	Linear Flow [pix/frame]	Distance [mm]	Actual Dist [mm]	Frame Error [mm]	Frame StdDev [mm]	Rolling Window Error [mm]	Rolling Window Error % [%]	Ro Win E Std [
OFZ0	1937	1.292	1.294	1525.6	1550	-24.4	32.5	-25.79	-1.66	
OFZ1	1973	1.276	1.277	1545.9	1560	-14.1	32.5	-14.67	-0.94	
OFZ2	1876	0.968	0.970	2034.3	2020	14.3	42.5	13.25	0.66	1
OFZ3	1861	0.962	0.963	2048.5	2030	18.5	46.0	17.52	0.86	1
OFZ4	1934	0.779	0.779	2535.1	2575	-39.9	52.9	-41.00	-1.59	1
OFZ5	1956	0.773	0.773	2551.8	2580	-28.2	44.8	-28.15	-1.09	1

```
Linear velocity: 30.0mm/s  Rotational velocity: -2.0deg/s
```

Zone	n	Raw Flow [pix/frame]	Linear Flow [pix/frame]	Distance [mm]	Actual Dist [mm]	Frame Error [mm]	Frame StdDev [mm]	Rolling Window Error [mm]	Rolling Window Error % [%]	Roll Wind Er Std [m
OFZ0	611	7.145	2.687	1469.0	1550	-81.0	135.3	-5.32	-0.34	46
OFZ1	578	6.772	2.584	1527.7	1560	-32.3	136.9	-7.74	-0.50	55
OFZ2	583	6.585	2.014	1959.4	2020	-60.6	239.4	61.81	3.06	77
OFZ3	581	6.341	1.948	2026.7	2030	-3.3	256.5	47.97	2.36	87
OFZ4	575	6.316	1.612	2448.4	2575	-126.6	379.8	30.28	1.18	127
OFZ5	581	6.078	1.583	2494.1	2580	-85.9	389.9	29.47	1.14	133

```
Linear velocity: 30.0mm/s  Rotational velocity: -4.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Winc Er Std
Zone		**[pix/frame]**	**[pix/frame]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[%]**	**[n**
OFZ0	415	11.656	2.789	1415.3	1550	-134.7	230.0	-25.99	-1.68	84
OFZ1	386	11.036	2.669	1478.8	1560	-81.2	229.6	-36.06	-2.31	59
OFZ2	388	11.239	2.076	1901.6	2020	-118.4	410.6	20.45	1.01	122
OFZ3	386	10.818	2.044	1930.6	2030	-99.4	411.3	-9.74	-0.48	97
OFZ4	377	11.106	1.696	2328.0	2575	-247.0	624.6	-33.64	-1.31	194
OFZ5	377	10.670	1.661	2376.2	2580	-203.8	628.2	-54.40	-2.11	152

```
Linear velocity: 30.0mm/s  Rotational velocity: -6.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Roll Winc Er Std
Zone		**[pix/frame]**	**[pix/frame]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[%]**	**[n**
OFZ0	245	16.019	2.738	1441.4	1550	-108.6	295.5	11.24	0.73	42
OFZ1	227	15.071	2.609	1513.1	1560	-46.9	303.4	-8.38	-0.54	47
OFZ2	238	15.761	2.088	1890.9	2020	-129.1	526.8	83.68	4.14	78
OFZ3	240	15.088	1.991	1982.9	2030	-47.1	579.7	24.12	1.19	76
OFZ4	236	15.777	1.675	2357.0	2575	-218.0	843.5	30.97	1.20	126
OFZ5	236	15.086	1.669	2364.7	2580	-215.3	818.6	-8.27	-0.32	12

```
Linear velocity: 30.0mm/s  Rotational velocity: 0.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Winc Er Std
Zone		**[pix/frame]**	**[pix/frame]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[%]**	**[n**
OFZ0	800	2.646	2.648	1490.8	1550	-59.2	14.3	-60.21	-3.88	9
OFZ1	794	2.544	2.543	1552.0	1560	-8.0	14.8	-8.58	-0.55	
OFZ2	740	1.969	1.968	2006.0	2020	-14.0	15.8	-15.66	-0.78	12
OFZ3	721	1.917	1.915	2060.9	2030	30.9	19.3	28.63	1.41	10
OFZ4	775	1.590	1.591	2481.1	2575	-93.9	28.9	-94.32	-3.66	17
OFZ5	726	1.552	1.553	2542.1	2580	-37.9	30.2	-37.27	-1.44	15

```
Linear velocity: 45.0mm/s  Rotational velocity: -2.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Winc Er Std
Zone		**[pix/frame]**	**[pix/frame]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[%]**	**[m**
OFZ0	374	8.419	4.013	1475.3	1550	-74.7	99.2	-30.24	-1.95	64
OFZ1	360	7.988	3.830	1546.0	1560	-14.0	99.4	-24.73	-1.59	52
OFZ2	360	7.521	3.002	1972.4	2020	-47.6	180.4	11.84	0.59	112
OFZ3	340	7.252	2.893	2046.4	2030	16.4	177.8	9.76	0.48	101
OFZ4	361	7.060	2.424	2442.1	2575	-132.9	282.7	-49.18	-1.91	180
OFZ5	343	6.806	2.363	2505.5	2580	-74.5	272.8	-39.80	-1.54	164

Linear velocity: 45.0mm/s Rotational velocity: -4.0deg/s

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Winc Er Std
Zone		**[pix/frame]**	**[pix/frame]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[%]**	**[m**
OFZ0	384	13.059	4.076	1452.5	1550	-97.5	173.0	-27.88	-1.80	48
OFZ1	364	12.305	3.859	1534.5	1560	-25.5	170.1	-35.07	-2.25	49
OFZ2	364	12.260	3.007	1969.3	2020	-50.7	304.0	6.87	0.34	86
OFZ3	358	11.757	2.942	2012.3	2030	-17.7	297.0	-10.29	-0.51	77
OFZ4	354	11.917	2.433	2433.5	2575	-141.5	466.6	-58.80	-2.28	139
OFZ5	353	11.424	2.386	2481.3	2580	-98.7	455.8	-65.50	-2.54	119

Linear velocity: 45.0mm/s Rotational velocity: -6.0deg/s

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Roll Winc Er Std
Zone		**[pix/frame]**	**[pix/frame]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[mm]**	**[%]**	**[m**
OFZ0	248	17.439	4.115	1438.7	1550	-111.3	204.7	-0.02	-0.00	35
OFZ1	234	16.428	3.917	1511.5	1560	-48.5	212.2	0.05	0.00	46
OFZ2	245	16.814	3.062	1933.4	2020	-86.6	392.3	83.07	4.11	64
OFZ3	230	16.112	3.017	1962.5	2030	-67.5	373.2	54.64	2.69	76
OFZ4	236	16.621	2.490	2377.4	2575	-197.6	583.9	54.02	2.10	100
OFZ5	233	15.913	2.450	2417.1	2580	-162.9	583.3	43.36	1.68	116

Linear velocity: 45.0mm/s Rotational velocity: 0.0deg/s

Zone	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Wind Er Std
		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[n
OFZ0	429	3.967	3.966	1492.8	1550	-57.2	9.8	-59.64	-3.85	
OFZ1	445	3.814	3.813	1552.8	1560	-7.2	8.5	-8.33	-0.53	4
OFZ2	450	2.955	2.953	2004.8	2020	-15.2	13.6	-17.58	-0.87	10
OFZ3	466	2.878	2.876	2058.4	2030	28.4	12.4	26.29	1.30	6
OFZ4	440	2.377	2.374	2493.7	2575	-81.3	17.9	-86.17	-3.35	1
OFZ5	452	2.322	2.322	2550.0	2580	-30.0	18.2	-31.57	-1.22	1

```
Linear velocity: 60.0mm/s  Rotational velocity: -2.0deg/s
```

Zone	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Roll Wind Er Std
		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[n
OFZ0	280	9.705	5.299	1489.8	1550	-60.2	74.9	48.48	3.13	14
OFZ1	245	9.206	5.082	1553.5	1560	-6.5	68.9	57.63	3.69	15
OFZ2	273	8.458	3.934	2006.8	2020	-13.2	131.8	163.40	8.09	27
OFZ3	259	8.161	3.823	2065.1	2030	35.1	130.1	163.80	8.07	28
OFZ4	267	7.808	3.172	2488.8	2575	-86.2	202.7	186.32	7.24	43
OFZ5	262	7.532	3.097	2549.1	2580	-30.9	205.0	198.97	7.71	46

```
Linear velocity: 60.0mm/s  Rotational velocity: -4.0deg/s
```

Zone	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Roll Wind Er Std
		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[n
OFZ0	265	14.235	5.337	1479.1	1550	-70.9	135.2	-6.37	-0.41	20
OFZ1	239	13.433	5.097	1548.9	1560	-11.1	129.8	0.54	0.03	29
OFZ2	241	13.108	3.968	1989.3	2020	-30.7	234.6	64.57	3.20	44
OFZ3	240	12.577	3.858	2046.3	2030	16.3	238.8	57.01	2.81	50
OFZ4	237	12.571	3.180	2482.5	2575	-92.5	367.2	28.55	1.11	70
OFZ5	239	12.063	3.119	2530.7	2580	-49.3	369.8	34.58	1.34	78

```
Linear velocity: 60.0mm/s  Rotational velocity: -6.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Roll Winc Er Stdl
Zone		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[n
OFZ0	252	18.891	5.436	1452.1	1550	-97.9	158.6	-5.14	-0.33	26
OFZ1	236	17.759	5.177	1524.8	1560	-35.2	168.8	8.53	0.55	4
OFZ2	242	17.885	4.059	1945.1	2020	-74.9	296.0	62.77	3.11	5
OFZ3	235	17.111	3.960	1993.8	2030	-36.2	301.3	54.85	2.70	6
OFZ4	241	17.481	3.277	2409.0	2575	-166.0	469.8	16.79	0.65	8
OFZ5	233	16.722	3.227	2446.0	2580	-134.0	457.0	28.02	1.09	9

```
Linear velocity: 60.0mm/s  Rotational velocity: 0.0deg/s
```

	n	Raw Flow	Linear Flow	Distance	Actual Dist	Frame Error	Frame StdDev	Rolling Window Error	Rolling Window Error %	Rol Winc Er Stdl
Zone		[pix/frame]	[pix/frame]	[mm]	[mm]	[mm]	[mm]	[mm]	[%]	[n
OFZ0	305	5.291	5.290	1492.4	1550	-57.6	8.0	-60.74	-3.92	
OFZ1	309	5.090	5.089	1551.3	1560	-8.7	7.4	-10.64	-0.68	
OFZ2	310	3.939	3.936	2005.6	2020	-14.4	10.5	-18.64	-0.92	1
OFZ3	308	3.838	3.839	2056.4	2030	26.4	10.4	24.92	1.23	
OFZ4	317	3.176	3.173	2488.2	2575	-86.8	13.0	-92.55	-3.59	1
OFZ5	321	3.101	3.100	2546.3	2580	-33.7	13.8	-36.37	-1.41	1

# Save the results for the plotting notebook import pickle pickle.dump(result_dict, open('{}full_analysis_results.p'.format(datadir), 'wb'))

```
In [ ]:
```

```
In [ ]:
```