Tpointer

October 20, 2020

1 Instrument Calibration

518021910971 裴奕博

[1]:
```python
import numpy as np
import pandas as pd
import re
from scipy.linalg import lstsq
```

1.0.1 Pre: clear data and calculate R and P

[2]:
```python
# change data format
df=pd.read_table('Tpointer2Cam.txt',header=None)
num=df.shape[0] # num of sample
df.columns=['Ri']
df['Pi']=None
df['Ri']=df['Ri'].apply(lambda x:x.replace('[',''))
df['Ri']=df['Ri'].apply(lambda x:x.replace(']',''))
df['Ri']=df['Ri'].apply(lambda x: x.split(','))
df['Ri']=df['Ri'].apply(lambda x:list(map(float,x)))
df['Ri']=df['Ri'].apply(lambda x:np.array(x).reshape(3,4))
df.head(5)
```

[2]:
```
                                                 Ri    Pi
0  [[-0.940318, 0.00452241, 0.339885, 24.7], [-0…  None
1  [[-0.94146, 0.0032108, 0.336806, 24.8], [-0.09…  None
2  [[-0.941584, 0.00107668, 0.336133, 24.86], [-0…  None
3  [[-0.941101, 0.00197468, 0.337541, 24.81], [-0…  None
4  [[-0.940003, 0.0029642, 0.340505, 24.68], [-0…  None
```

[3]:
```python
# split Ri and Pi
df['Ri']=df['Ri'].apply(lambda x:np.hsplit(x,np.array([3])))
df.loc[:,'Pi']=df.loc[:,'Ri'].apply(lambda x:-x[1])
df.loc[:,'Ri']=df.loc[:,'Ri'].apply(lambda x:x[0])
neg=-1*np.identity(3)
print(neg)
df['Ri']=df['Ri'].apply(lambda x:np.concatenate((x,neg),axis=1))
```

```
df.head(5)
```

```
[[-1. -0. -0.]
 [-0. -1. -0.]
 [-0. -0. -1.]]
```

```
[3]:                                                     Ri  \
     0  [[-0.940318, 0.00452241, 0.339885, -1.0, -0.0,…
     1  [[-0.94146, 0.0032108, 0.336806, -1.0, -0.0, -…
     2  [[-0.941584, 0.00107668, 0.336133, -1.0, -0.0,…
     3  [[-0.941101, 0.00197468, 0.337541, -1.0, -0.0,…
     4  [[-0.940003, 0.0029642, 0.340505, -1.0, -0.0, …

                                  Pi
     0    [[-24.7], [121.7], [1551.76]]
     1    [[-24.8], [122.14], [1551.4]]
     2  [[-24.86], [122.59], [1551.23]]
     3  [[-24.81], [122.44], [1551.43]]
     4  [[-24.68], [122.16], [1552.08]]
```

[4]:

```
# conatenate matrix
Ri_matrix=df['Ri'][0]
Pi_matrix=df['Pi'][0]
for i in np.arange(1,num):
    Ri_matrix=np.concatenate((Ri_matrix,df['Ri'][i]),axis=0)
    Pi_matrix=np.concatenate((Pi_matrix,df['Pi'][i]),axis=0)

print(Ri_matrix.shape)
print(Pi_matrix.shape)
```

```
(777, 6)
(777, 1)
```

1.0.2 1. Compute the coordinate of the tool tip in the pointer’ s local coordinate system

$$\begin{bmatrix} R_1 & -\mathbf{I} \\ R_2 & -\mathbf{I} \\ \vdots & \vdots \\ R_n & -\mathbf{I} \end{bmatrix} \begin{bmatrix} p_t \\ p_p \end{bmatrix} = \begin{bmatrix} -p_1 \\ -p_2 \\ \vdots \\ -p_n \end{bmatrix}$$

[5]:

```
result = lstsq(Ri_matrix,Pi_matrix)
result
```

```
[5]: (array([[ 2.08418195e+02],
             [-6.43296344e-01],
             [-3.31314000e+01],
             [-1.82426929e+02],
             [-1.33443408e+02],
             [-1.51384362e+03]]),
      array([52.0256386]),
      6,
      array([22.53733552, 22.46134983, 22.31442892,  4.47201692,  3.66339155,
              3.16246496]))
```

```
[6]: tmp=np.split(result[0],2)
     pt=tmp[0]
     pp=tmp[1]
     print('pt=',pt.flatten())
     print('pp=',pp.flatten())
```

```
pt= [208.41819488  -0.64329634 -33.13140004]
pp= [ -182.42692914  -133.44340751 -1513.84362071]
```

1.0.3 2. Further, compute the root mean square errors of the tip calibration

```
[7]: residual=Pi_matrix-np.dot(Ri_matrix,result[0])
     error=np.sqrt(np.sum(np.square(residual))/num/3)
     error
```

```
[7]: 0.2587606303880167
```