normalize_diceface_values-v3

January 12, 2021

1 Code by Colin Werkhoven & Matthew Turkenburg & Adnan Akbas & Ali safdari

Literature https://en.wikipedia.org/wiki/Rotation_matrix

https://medium.com/swlh/how-to-efficiently-loop-through-pandas-dataframe-660e4660125d

1.0.1 The issue

Currently, once the ActivPal device rotates in certain directions the associated X, Y, Z values also flip positions. This will cause problems in the future with creating accurate Machine Learning models because the data won't be consistent and will therefore not produce accurate results. ### The solution This script will normalize all X, Y, Z value in the same format by looking at the `diceface` of each row to determine if the current format is correct or needs to be adjusted. ### Idea Compare two values with the same `pal_diceFace`. Take their pal_time and check if the difference is greater then 15 seconds (current interval). If the interval is greater then 15, skip the comparison, if not apply our function that changes the X, Y, Z values to the same values as pal_diceFace 2.

[1]:
```
# Imports + Initializations

from helpers import pandas_helper as helper
from helpers import math_helper as mhp
from sensors.activpal import *

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import os

activpal = Activpal()
```

[2]:
```
def opposite_of_number(number):
    return -number if number > 0 else abs(number)
```

1.0.2 Converters

All the converters convert xyz to the position of diceface 1 #### Diceface 2 to 1

[15]:
```
# return x,y,z in this order
def convert_xyz_diceface_2_to_1(x, y, z):
    #(x, y, z)
    return (x, np.negative(z), y)
```

Diceface 3 to 1

[16]:
```
# return x,y,z in this order
def convert_xyz_diceface_3_to_1(x,y,z):
    #(x y, z)
    return (y,  np.negative(z),  np.negative(x))
```

Diceface 4 to 1

[17]:
```
# return x,y,z in this order
def convert_xyz_diceface_4_to_1(x, y, z):
    #(x, y, z)
    return (np.negative(y), np.negative(z), x)
```

Diceface 5 to 1

[18]:
```
# return x,y,z in this order
def convert_xyz_diceface_5_to_1(x,y,z):
    #(x, y, z)
    return (np.negative(x), np.negative(z), np.negative(y) )
```

Diceface 6 to 1

[19]:
```
# return x,y,z in this order
def convert_xyz_diceface_6_to_1(x, y, z):
    #(x, y, z)
    return (np.negative(x), y, np.negative(z))
```

2 implementation

[49]:
```
def correct_activpal_data(correspondent):
    activpal_15s_df = helper.read_csv_activpal1_15(correspondent)
    activpal_20_df = activpal.read_data(correspondent)

    activpal_20_df['pal_accX'] = mhp.
 ↪convert_value_to_g(activpal_20_df['pal_accX'])
```

```
   activpal_20_df['pal_accY'] = mhp.
↪convert_value_to_g(activpal_20_df['pal_accY'])
   activpal_20_df['pal_accZ'] = mhp.
↪convert_value_to_g(activpal_20_df['pal_accZ'])

   activpal_20_df.drop(columns=['pal_time'], inplace=True)

   # This solve refence and duplicate issue
   corrected_activpal_df = activpal_20_df.copy()

   for index, diceface in zip(activpal_15s_df.index,␣
↪activpal_15s_df['pal_diceFace']):

       previous_index = activpal_15s_df.index.get_loc(index) - 1
       prev_index = index - pd.DateOffset(seconds=15)

       if previous_index >= 0:
           prev_index = activpal_15s_df.index[previous_index]

       activpal_20_df_filtered = activpal_20_df.loc[prev_index:index]

       x = activpal_20_df_filtered['pal_accX']
       y = activpal_20_df_filtered['pal_accY']
       z = activpal_20_df_filtered['pal_accZ']

       if diceface == 1:
           new_x = x
           new_y = y
           new_z = z
       elif diceface == 2:
           new_x, new_y, new_z = convert_xyz_diceface_2_to_1(x, y, z)
       elif diceface == 3:
           new_x, new_y, new_z = convert_xyz_diceface_3_to_1(x, y, z)
       elif diceface == 4:
           new_x, new_y, new_z = convert_xyz_diceface_4_to_1(x, y, z)
       elif diceface == 5:
           new_x, new_y, new_z = convert_xyz_diceface_5_to_1(x, y, z)
       elif diceface == 6:
           new_x, new_y, new_z = convert_xyz_diceface_6_to_1(x, y, z)


       corrected_activpal_df.loc[activpal_20_df_filtered.index, 'pal_accX'] =␣
↪new_x
       corrected_activpal_df.loc[activpal_20_df_filtered.index, 'pal_accY'] =␣
↪new_y
       corrected_activpal_df.loc[activpal_20_df_filtered.index, 'pal_accZ'] =␣
↪new_z
```

```
        corrected_activpal_df.loc[activpal_20_df_filtered.index,␣
↪'pal_diceFace'] = diceface


    corrected_activpal_df['pal_accX'] = mhp.
↪convert_g_to_scaled_value(corrected_activpal_df['pal_accX'])
    corrected_activpal_df['pal_accY'] = mhp.
↪convert_g_to_scaled_value(corrected_activpal_df['pal_accY'])
    corrected_activpal_df['pal_accZ'] = mhp.
↪convert_g_to_scaled_value(corrected_activpal_df['pal_accZ'])

    return corrected_activpal_df
```

[53]:
```
for directory in os.walk('../../data'):
    if directory[0] == '../../data':
        for respDirect in directory[1]:
            if respDirect not in ['output', 'throughput', 'Test data','.
↪ipynb_checkpoints']:
                print("Extracting " + respDirect)

                new_df = correct_activpal_data(respDirect)
                print("Extracted "+ respDirect)
                new_df.to_csv('../../data/'+respDirect+'/activpal_20_diceface.
↪csv', date_format='%Y-%m-%d %H:%M:%S:%f')
                print('saved' + '../../data/'+respDirect+'/activpal_20_diceface.
↪csv')


print("Done normalizing data")
```

```
Extracting BMR099
Extracted BMR099
saved../../data/BMR099/activpal_20_diceface.csv
Extracting BMR025
Extracted BMR025
saved../../data/BMR025/activpal_20_diceface.csv
Extracting BMR060
Extracted BMR060
saved../../data/BMR060/activpal_20_diceface.csv
Extracting BMR012
Extracted BMR012
saved../../data/BMR012/activpal_20_diceface.csv
Extracting BMR035
Extracted BMR035
saved../../data/BMR035/activpal_20_diceface.csv
Extracting BMR030
Extracted BMR030
```

```
saved../../data/BMR030/activpal_20_diceface.csv
Extracting BMR051
Extracted BMR051
saved../../data/BMR051/activpal_20_diceface.csv
Extracting BMR044
Extracted BMR044
saved../../data/BMR044/activpal_20_diceface.csv
Extracting BMR043
Extracted BMR043
saved../../data/BMR043/activpal_20_diceface.csv
Extracting BMR004
Extracted BMR004
saved../../data/BMR004/activpal_20_diceface.csv
Extracting BMR011
Extracted BMR011
saved../../data/BMR011/activpal_20_diceface.csv
Extracting BMR098
Extracted BMR098
saved../../data/BMR098/activpal_20_diceface.csv
Extracting BMR034
Extracted BMR034
saved../../data/BMR034/activpal_20_diceface.csv
Extracting BMR014
Extracted BMR014
saved../../data/BMR014/activpal_20_diceface.csv
Extracting BMR036
Extracted BMR036
saved../../data/BMR036/activpal_20_diceface.csv
Extracting BMR052
Extracted BMR052
saved../../data/BMR052/activpal_20_diceface.csv
Extracting BMR002
Extracted BMR002
saved../../data/BMR002/activpal_20_diceface.csv
Extracting BMR031
Extracted BMR031
saved../../data/BMR031/activpal_20_diceface.csv
Extracting BMR097
Extracted BMR097
saved../../data/BMR097/activpal_20_diceface.csv
Extracting BMR008
Extracted BMR008
saved../../data/BMR008/activpal_20_diceface.csv
Extracting BMR015
Extracted BMR015
saved../../data/BMR015/activpal_20_diceface.csv
Extracting BMR033
Extracted BMR033
```

```
saved../../data/BMR033/activpal_20_diceface.csv
Extracting BMR100
Extracted BMR100
saved../../data/BMR100/activpal_20_diceface.csv
Extracting BMR064
Extracted BMR064
saved../../data/BMR064/activpal_20_diceface.csv
Extracting BMR055
Extracted BMR055
saved../../data/BMR055/activpal_20_diceface.csv
Extracting BMR027
Extracted BMR027
saved../../data/BMR027/activpal_20_diceface.csv
Extracting BMR041
Extracted BMR041
saved../../data/BMR041/activpal_20_diceface.csv
Extracting BMR053
Extracted BMR053
saved../../data/BMR053/activpal_20_diceface.csv
Extracting BMR042
Extracted BMR042
saved../../data/BMR042/activpal_20_diceface.csv
Extracting BMR018
Extracted BMR018
saved../../data/BMR018/activpal_20_diceface.csv
Extracting BMR058
Extracted BMR058
saved../../data/BMR058/activpal_20_diceface.csv
Extracting BMR040
Extracted BMR040
saved../../data/BMR040/activpal_20_diceface.csv
Extracting BMR032
Extracted BMR032
saved../../data/BMR032/activpal_20_diceface.csv
Done normalizing data
```

[38]:
```
new_df = correct_activpal_data('BMR099')
```

```
--------BEGIN----------
                           pal_accX  pal_accY  pal_accZ
pal_time
2019-09-12 09:51:55.000001  0.111111 -0.047619  1.126984
2019-09-12 09:51:55.050000  0.000000  0.190476  1.206349
2019-09-12 09:51:55.100000  0.000000  0.190476  1.174603
2019-09-12 09:51:55.150000  0.015873  0.317460  1.158730
2019-09-12 09:51:55.199999 -0.031746  0.285714  1.158730
--------Y----------
pal_time
```

```
2019-09-12 09:51:55.000001   -1.126984
2019-09-12 09:51:55.050000   -1.206349
2019-09-12 09:51:55.100000   -1.174603
2019-09-12 09:51:55.150000   -1.158730
2019-09-12 09:51:55.199999   -1.158730
2019-09-12 09:51:55.249999   -1.111111
2019-09-12 09:51:55.299999   -1.238095
2019-09-12 09:51:55.349998   -1.206349
2019-09-12 09:51:55.399999   -1.253968
2019-09-12 09:51:55.449998   -1.126984
2019-09-12 09:51:55.499997   -1.174603
2019-09-12 09:51:55.549997   -1.190476
2019-09-12 09:51:55.599998   -1.222222
2019-09-12 09:51:55.649997   -1.158730
2019-09-12 09:51:55.699996   -1.142857
2019-09-12 09:51:55.749995   -1.222222
2019-09-12 09:51:55.799995   -1.222222
2019-09-12 09:51:55.849995   -0.968254
2019-09-12 09:51:55.900004   -0.746032
2019-09-12 09:51:55.950004   -0.857143
2019-09-12 09:51:56.000003   -0.777778
2019-09-12 09:51:56.050002   -0.650794
2019-09-12 09:51:56.100003   -0.666667
2019-09-12 09:51:56.150002   -1.476190
2019-09-12 09:51:56.200002   -0.730159
2019-09-12 09:51:56.250001   -0.523810
2019-09-12 09:51:56.300001   -0.841270
2019-09-12 09:51:56.350001   -0.904762
2019-09-12 09:51:56.400001   -0.904762
2019-09-12 09:51:56.450000   -0.619048
2019-09-12 09:51:56.500000   -0.571429
2019-09-12 09:51:56.550000   -0.841270
2019-09-12 09:51:56.599999   -0.698413
2019-09-12 09:51:56.649999   -0.650794
2019-09-12 09:51:56.699998   -0.603175
2019-09-12 09:51:56.749999   -0.793651
2019-09-12 09:51:56.799998   -0.523810
2019-09-12 09:51:56.849998   -0.650794
2019-09-12 09:51:56.899998   -0.888889
2019-09-12 09:51:56.949997   -1.047619
2019-09-12 09:51:56.999997   -1.174603
Name: pal_accZ, dtype: float64
                            pal_accX  pal_accY  pal_accZ
pal_time
2019-09-12 09:51:55.000001  0.111111 -1.126984  1.126984
2019-09-12 09:51:55.050000  0.000000 -1.206349  1.206349
2019-09-12 09:51:55.100000  0.000000 -1.174603  1.174603
2019-09-12 09:51:55.150000  0.015873 -1.158730  1.158730
```

```
2019-09-12 09:51:55.199999 -0.031746 -1.158730  1.158730
--------Z----------
pal_time
2019-09-12 09:51:55.000001   -1.126984
2019-09-12 09:51:55.050000   -1.206349
2019-09-12 09:51:55.100000   -1.174603
2019-09-12 09:51:55.150000   -1.158730
2019-09-12 09:51:55.199999   -1.158730
2019-09-12 09:51:55.249999   -1.111111
2019-09-12 09:51:55.299999   -1.238095
2019-09-12 09:51:55.349998   -1.206349
2019-09-12 09:51:55.399999   -1.253968
2019-09-12 09:51:55.449998   -1.126984
2019-09-12 09:51:55.499997   -1.174603
2019-09-12 09:51:55.549997   -1.190476
2019-09-12 09:51:55.599998   -1.222222
2019-09-12 09:51:55.649997   -1.158730
2019-09-12 09:51:55.699996   -1.142857
2019-09-12 09:51:55.749995   -1.222222
2019-09-12 09:51:55.799995   -1.222222
2019-09-12 09:51:55.849995   -0.968254
2019-09-12 09:51:55.900004   -0.746032
2019-09-12 09:51:55.950004   -0.857143
2019-09-12 09:51:56.000003   -0.777778
2019-09-12 09:51:56.050002   -0.650794
2019-09-12 09:51:56.100003   -0.666667
2019-09-12 09:51:56.150002   -1.476190
2019-09-12 09:51:56.200002   -0.730159
2019-09-12 09:51:56.250001   -0.523810
2019-09-12 09:51:56.300001   -0.841270
2019-09-12 09:51:56.350001   -0.904762
2019-09-12 09:51:56.400001   -0.904762
2019-09-12 09:51:56.450000   -0.619048
2019-09-12 09:51:56.500000   -0.571429
2019-09-12 09:51:56.550000   -0.841270
2019-09-12 09:51:56.599999   -0.698413
2019-09-12 09:51:56.649999   -0.650794
2019-09-12 09:51:56.699998   -0.603175
2019-09-12 09:51:56.749999   -0.793651
2019-09-12 09:51:56.799998   -0.523810
2019-09-12 09:51:56.849998   -0.650794
2019-09-12 09:51:56.899998   -0.888889
2019-09-12 09:51:56.949997   -1.047619
2019-09-12 09:51:56.999997   -1.174603
Name: pal_accY, dtype: float64
                            pal_accX  pal_accY  pal_accZ
pal_time
2019-09-12 09:51:55.000001  0.111111 -1.126984 -1.126984
```

```
2019-09-12 09:51:55.050000  0.000000 -1.206349 -1.206349
2019-09-12 09:51:55.100000  0.000000 -1.174603 -1.174603
2019-09-12 09:51:55.150000  0.015873 -1.158730 -1.158730
2019-09-12 09:51:55.199999 -0.031746 -1.158730 -1.158730
--------END----------
                           pal_accX  pal_accY  pal_accZ  pal_diceFace
pal_time
2019-09-12 09:51:55.000001  0.111111 -1.126984 -1.126984           2.0
2019-09-12 09:51:55.050000  0.000000 -1.206349 -1.206349           2.0
2019-09-12 09:51:55.100000  0.000000 -1.174603 -1.174603           2.0
2019-09-12 09:51:55.150000  0.015873 -1.158730 -1.158730           2.0
2019-09-12 09:51:55.199999 -0.031746 -1.158730 -1.158730           2.0
```

[50]:
```
new_df = correct_activpal_data('BMR099')
```

[12]:
```
activpal_15s_df = helper.read_csv_activpal1_15('BMR099')
activpal_20_df = activpal.read_data('BMR099')
```

[52]:
```
new_df.head()
```

[52]:
```
                           pal_accX  pal_accY  pal_accZ  pal_diceFace
pal_time
2019-09-12 09:51:55.000001     134.0      56.0     124.0           2.0
2019-09-12 09:51:55.050000     127.0      51.0     139.0           2.0
2019-09-12 09:51:55.100000     127.0      53.0     139.0           2.0
2019-09-12 09:51:55.150000     128.0      54.0     147.0           2.0
2019-09-12 09:51:55.199999     125.0      54.0     145.0           2.0
```

[14]:
```
activpal_20_df.head()
```

[14]:
```
                                        pal_time  pal_accX  pal_accY  \
pal_time
2019-09-12 09:51:55.000001 2019-09-12 09:51:55.000001       134       124
2019-09-12 09:51:55.050000 2019-09-12 09:51:55.050000       127       139
2019-09-12 09:51:55.100000 2019-09-12 09:51:55.100000       127       139
2019-09-12 09:51:55.150000 2019-09-12 09:51:55.150000       128       147
2019-09-12 09:51:55.199999 2019-09-12 09:51:55.199999       125       145

                           pal_accZ
pal_time
2019-09-12 09:51:55.000001       198
2019-09-12 09:51:55.050000       203
2019-09-12 09:51:55.100000       201
2019-09-12 09:51:55.150000       200
2019-09-12 09:51:55.199999       200
```