slant_tilt_demo

Monica Li[1]

[1]City University of Hong Kong

Jun 11, 2019

Working dx.doi.org/10.17504/protocols.io.322gqge

Monica Li

Main

1 Get the writing data from a digital tablet as attached. example.pkl

2 Download the data and the command in script, save them in the same folder

slant_tilt_demo, step 2

```
# coding: utf-8

# In[ ]:


import pandas as pd
import numpy as np
from sklearn import linear_model
#from sklearn.metrics import mean_squared_error, r2_score, explained_variance_score
STROKE = 'STROKE'
CHARACTER = 'CHARACTER'


# In[ ]:


#Judge whether a stroke is vertial or horizonal
def vert(chardf):
    regrv = linear_model.LinearRegression()
    regrh = linear_model.LinearRegression()
    newdf = pd.DataFrame(columns =['SAMPLE','CHARACTER','STROKE','POINT','X','Y','Z','Vertical'])
    for i in range(chardf[STROKE].max()+1):
        chardfX = chardf.loc[chardf.STROKE == i].loc[:,['X']]
        chardfY = chardf.loc[chardf.STROKE == i].loc[:,['Y']]
        regrv.fit(chardfX, chardfY)
        regrv.fit(chardfX, chardfY)
        regrh.fit(chardfY, chardfX)
        slope = regrv.coef_[0]
        slope_r = regrh.coef_[0]
        #vertical stroke
        if (slope_r > -0.5773) and (slope_r < 0.5773):
            vertdf = chardf.loc[chardf.STROKE == i].assign(Vertical = 1)
            newdf = newdf.append(vertdf)
         #horizontal stroke
        elif (slope <0.3839) and (slope > -0.8098):
            vertdf = chardf.loc[chardf.STROKE == i].assign(Vertical = -1)
            newdf = newdf.append(vertdf)
        else:
            vertdf = chardf.loc[chardf.STROKE == i]
            newdf = newdf.append(vertdf)
    return newdf


# In[ ]:


#read file
df = pd.read_pickle('example.pkl')


# In[ ]:


#calculate linear regression result
df = df.assign(Vertical = 0)
charlist = df.loc[:,'CHARACTER'].drop_duplicates().tolist()
wholedf = pd.DataFrame()
for i in charlist:
    wholedf =wholedf.append(vert(df.loc[(df.CHARACTER == i)]))


# In[ ]:


raw = wholedf[['CHARACTER', 'STROKE', 'POINT','X', 'Y', 'Vertical']]
new = pd.DataFrame(columns = ['CHARACTER', 'STROKE', 'POINT','X', 'Y', 'Vertical'])
new = new.append(raw.iloc[0])


# In[ ]:


#drop the points too close to each other
count = 0
while count < raw.shape[0]:
    x0 = raw.iloc[0]['X']
    y0 = raw.iloc[0]['Y']
    if (raw.iloc[count][3] - x0)**2 + (raw.iloc[count][4] - y0)**2 < 100:
        count += 1
    else:
        new = new.append(raw.iloc[count])
        raw = raw.iloc[count+1:]
        count = 0
#take difference and calculate angle
new['X_diff'] = np.nan
new['Y_diff'] = np.nan
new['Angle'] = np.nan
for i in range(1,new.shape[0]):
    if new.iloc[i,1] == new.iloc[i-1,1]:
        new.iloc[i,6] = new.iloc[i,3]-new.iloc[i-1,3]
        new.iloc[i,7] = new.iloc[i,4]-new.iloc[i-1,4]
        new.iloc[i,8] = round(np.angle(complex(new.iloc[i,6],new.iloc[i,7]),deg = True))


# In[ ]:


#prepare data for slant and tile calculation
new = new.dropna()
slantdf = new[new['Vertical']==1]
tiltdf = new[new['Vertical']==-1]


# In[ ]:


print("slant: ", slantdf['Angle'].mode()[0], ", tilt:", tiltdf['Angle'].mode()[0])
```

A demonstration for "Measuring the tilt and slant of Chinese handwriting in primary school students: A computerized approach"

python 3.6

3 Run the command