

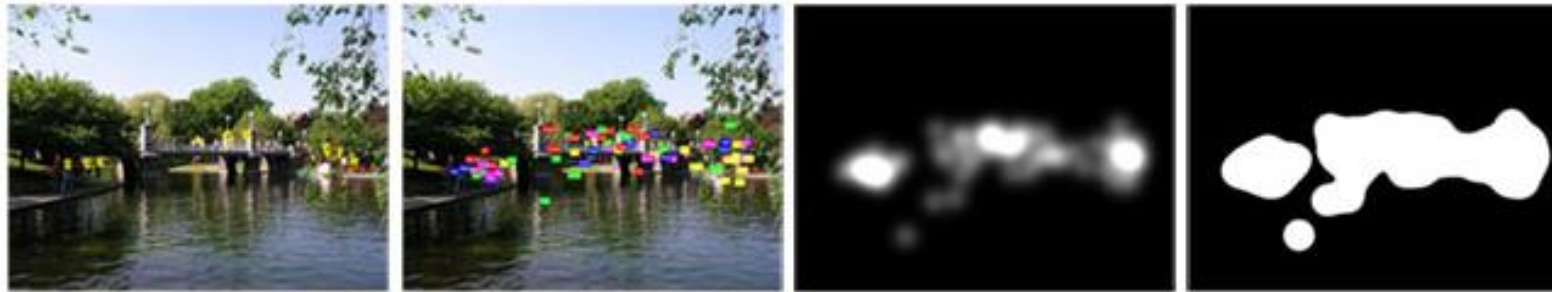
2. 헬스케어 데이터

Part2. 헬스케어 데이터의 종류_실습



실습 - Where people look

- Eye tracking database
 - ✓ natural images of objects and scenes downloaded from Flickr and LabelMe
 - ✓ 15 users on 1003 images
- 사진의 어느 곳을 응시했는지를 알 수 있다

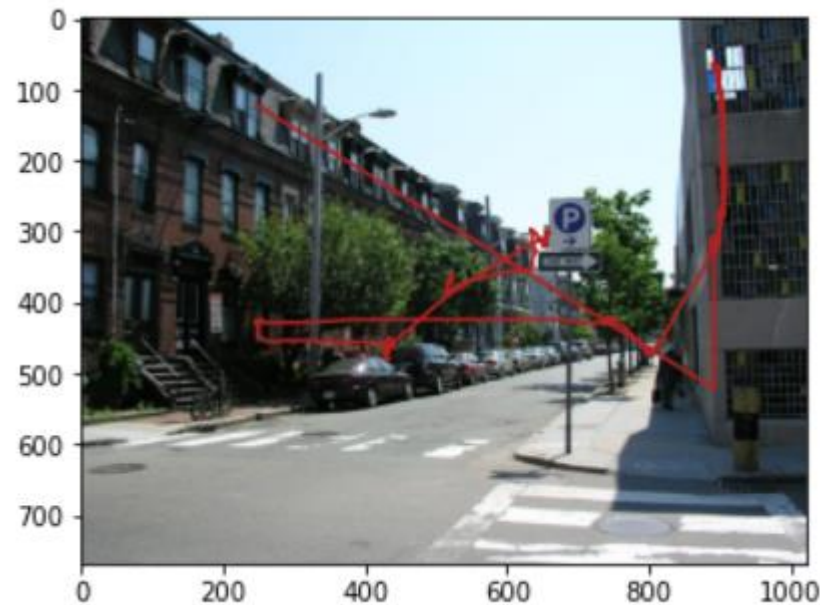


- 논문 원본:
<http://people.csail.mit.edu/tjudd/WherePeopleLook/Docs/wherepeoplelook.pdf>



실습 - Where people look

- 해당 데이터셋에서 발췌한 eye tracking 데이터의 궤적과 자극 이미지(boston street)를 아래 예시와 같이 함께 그려보자
 - ✓ 실습에 사용할 파일: eyedata.mat, eyesample_stim.jpeg
 - ✓ 주의: 데이터의 확장자는 mat이므로 scipy.io의 loadmat을 사용해야 함



실습 - Where people look

■ mat 파일 읽기

- ✓ scipy.io의 loadmat()을 사용하여 읽어들이는 eyedata.mat에 저장된 시선추적 데이터를 추출해야 한다
- ✓ 'eyeData'에 저장되어 있음

■ 데이터프레임 변환

- ✓ data_raw에 x, y 좌표값만 저장한 후, pandas 데이터프레임으로 변환
- ✓ pdata에 좌표값을 저장함

```
import scipy.io
import pandas as pd
mat = scipy.io.loadmat('/content/drive/MyDrive/data/eyedata.mat')
mat
```

```
{'__header__': b'MATLAB 5.0 MAT-file Platform: nt, Created on: Thu Sep 16 09:40:33 2021',
'__version__': '1.0',
'__globals__': [],
'block': array([[array([[1]], dtype=uint8)], dtype=object),
'trial': array([[array([[268]], dtype=uint16)], dtype=object),
'eyeData': array([[array([[ 521.74683544,  371.328      ],
[ 521.74683544,  369.216      ],
[ 518.96202532,  364.992      ],
...,
[ 246.25316456,  122.112      ],
[ -1126.2721519 , -849.916      ],
[ -19916.29113924, -14551.752      ]]]], dtype=object))}
```

```
data_raw=mat['eyeData'][0][0]
data_raw
```

```
array([[ 521.74683544,  371.328      ],
[ 521.74683544,  369.216      ],
[ 518.96202532,  364.992      ],
...,
[ 246.25316456,  122.112      ],
[ -1126.2721519 , -849.916      ],
[ -19916.29113924, -14551.752      ]])
```

```
pdata = pd.DataFrame(data_raw)
```



실습 - Where people look

■ 데이터 탐색하기

✓ 끝에 두 개(723, 724)의 값은 outlier이므로 제거(drop)하고 시각화

pdata

	0	1
0	521.746835	371.328
1	521.746835	369.216
2	518.962025	364.992
3	521.746835	362.880
4	518.962025	362.880
...
720	892.126582	508.608
721	892.126582	521.280
722	246.253165	122.112
723	-1126.272152	-849.916
724	-19916.291139	-14551.752

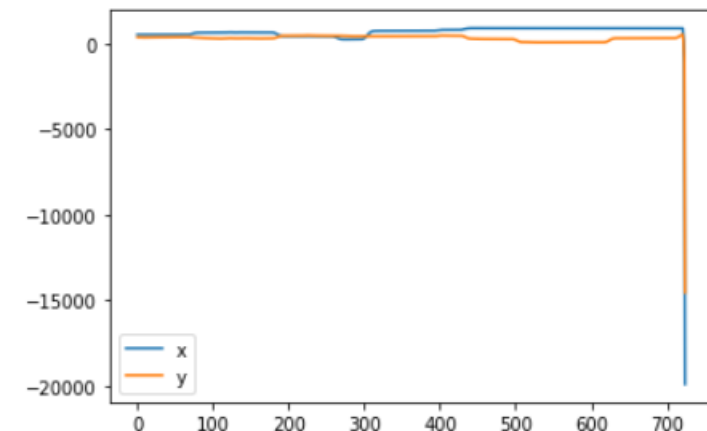
725 rows × 2 columns

```
pdata.columns=['x','y']  
pdata.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
x	725.0	676.444653	793.393759	-19916.291139	521.746835	744.531646	894.911392	908.83544
y	725.0	300.868497	568.561942	-14551.752000	278.400000	324.864000	426.240000	521.280000

```
pdata.plot()
```

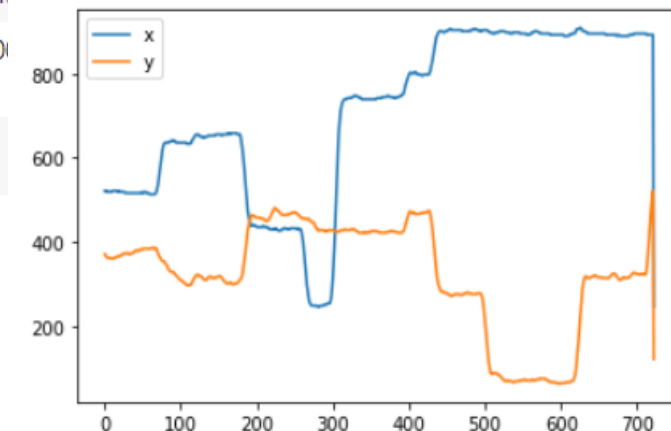
<matplotlib.axes._subplots.AxesSubplot at 0x7f1d3b0c6d90>



```
pdata.drop([723,724], inplace=True)
```

```
pdata.plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f1d3afda290>



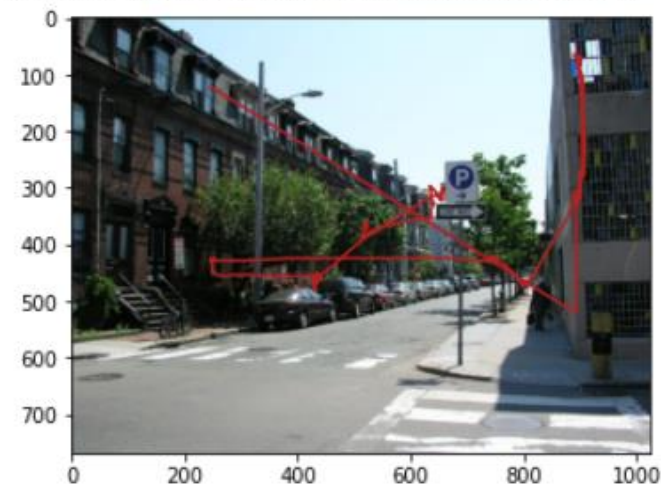
실습 - Where people look

- 추출한 pdata의 좌표값을 이용하여 eye tracking 데이터의 궤적과 자극 이미지(boston street)를 함께 그린다

```
import matplotlib.pyplot as plt  
img = plt.imread('/content/drive/MyDrive/data/eyesample_stim.jpeg')
```

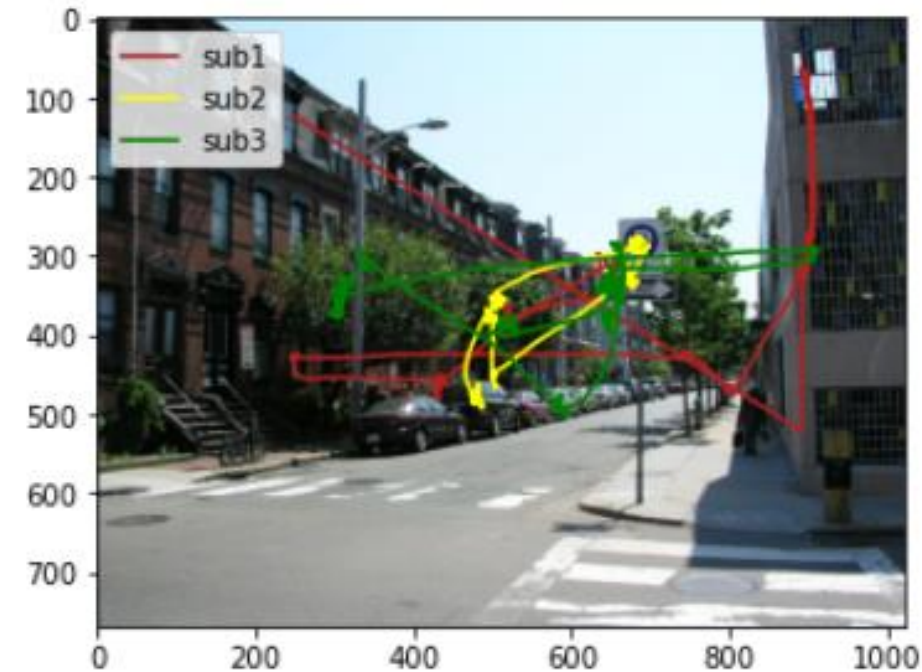
```
fig, ax = plt.subplots()  
ax.imshow(img)  
ax.plot(pdata['x'], pdata['y'], color='firebrick')
```

[<matplotlib.lines.Line2D at 0x7f1d3a9cc590>]



실습 - Where people look

- eyedata2.mat과 eyedata3.mat은 동일한 이미지를 응시한 다른 두 사람의 시선 추적 데이터이다
- boston street 이미지 위에 앞선 실습에서 사용한 시선 추적 데이터를 포함하여 총 세 사람의 응시 궤적을 함께 그려보자




```
import scipy.io
import pandas as pd
```

```
def fileToDataFrame(fileName):
    mat = scipy.io.loadmat(fileName)
    data_raw=mat['eyeData'][0][0]
    return pd.DataFrame(data_raw)
```

```
pdata1 = fileToDataFrame('./data/eyedata.mat')
pdata2 = fileToDataFrame('./data/eyedata2.mat')
pdata3 = fileToDataFrame('./data/eyedata3.mat')
```

pdata2

	0	1
0	496.683544	377.664
1	496.683544	379.776
2	496.683544	379.776
3	496.683544	379.776
4	496.683544	381.888
...
720	479.974684	476.928
721	479.974684	479.040
722	479.974684	479.040
723	479.974684	479.040
724	-9455.056962	-6919.528

725 rows × 2 columns

```
pdata2.drop([724], inplace=True)
```

```
pdata2.columns=['x', 'y']
```

pdata3

	0	1
0	528.638655	376.637450
1	528.638655	376.637450
2	525.865546	376.637450
3	525.865546	376.637450
4	525.865546	380.844622
...
720	647.882353	393.466135
721	650.655462	393.466135
722	647.882353	393.466135
723	647.882353	393.466135
724	647.882353	391.362550

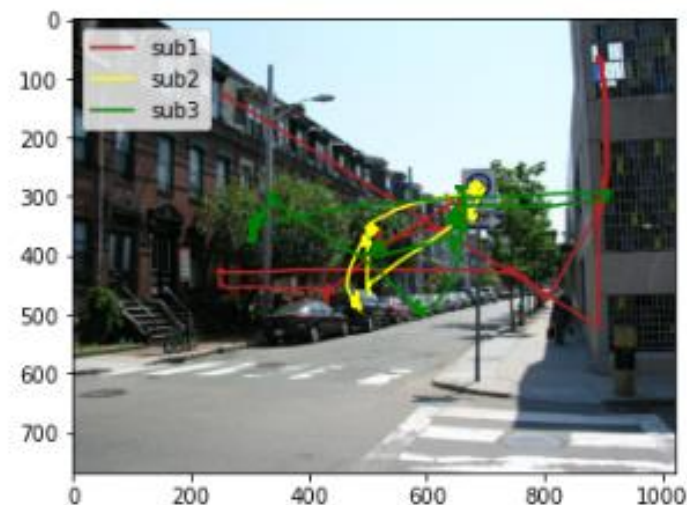
725 rows × 2 columns

```
pdata3.columns=['x', 'y']
```

```
import matplotlib.pyplot as plt
img = plt.imread('./data/eyesample_stim.jpeg')
```

```
fig,ax = plt.subplots()
ax.imshow(img)
ax.plot(pdata1['x'],pdata1['y'],color='firebrick',label='sub1')
ax.plot(pdata2['x'],pdata2['y'],color='yellow',label='sub2')
ax.plot(pdata3['x'],pdata3['y'],color='green',label='sub3')
ax.legend()
```

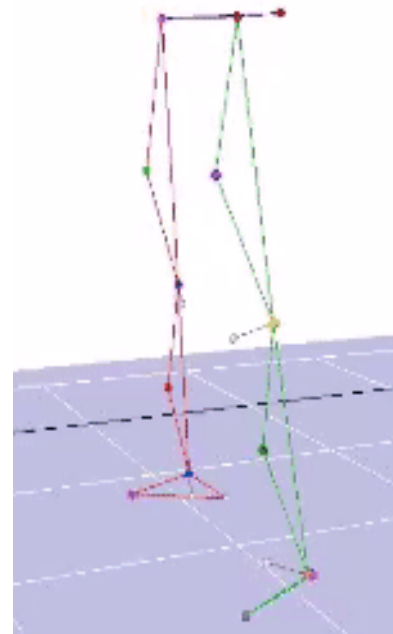
<matplotlib.legend.Legend at 0x26cefa89e20>



실습 – Squat 자세 모니터링

- 모션캡처 결과로 계산된 무릎 각도 데이터 탐색하기
 - ✓ Q) squat_angle.csv에 기록된 데이터는 총 몇 회의 squat를 수행한 데이터인가?
 - 데이터를 적절히 시각화해보고, 횟수를 카운트 하는 알고리즘을 작성해 보자
 - 이 데이터의 샘플링 레이트는 60Hz이다

	A	B	C
1	Frame#	Time	Angle
2	1000	16.65	160.5226
3	1001	16.667	160.5331
4	1002	16.683	160.5449
5	1003	16.7	160.5545
6	1004	16.717	160.5594
7	1005	16.733	160.5582
8	1006	16.75	160.5503
9	1007	16.767	160.5356
10	1008	16.783	160.515
11	1009	16.8	160.4904
12	1010	16.817	160.4646
13	1011	16.833	160.4404



실습 - Squat 자세 모니터링

■ squat_angle.csv 파일 탐색

- ✓ 시각화
- ✓ 어떤 값을 기준으로 무릎의 굴절과 신전을 판단할 것인지?

■ InClass Assignment

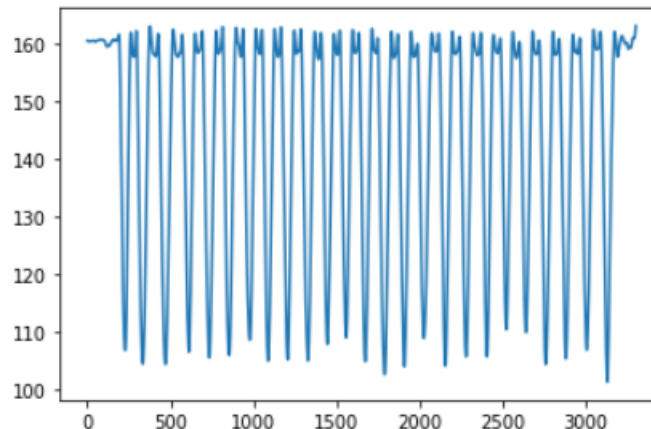
- ✓ 수업 종료 까지 코드와 그림을 한 화면에 캡처하여 업로드합니다
- ✓ *완성되지 않아도 됨! 구현한 데까지 올리기
- > 수업참여 점수 반영

```
# load dataset
data = pd.read_csv("/content/drive/MyDrive/data/squat_angle.csv", delimiter=',')
data.head()
```

	Frame#	Time	Angle
0	1000	16.650	160.52260
1	1001	16.667	160.53308
2	1002	16.683	160.54486
3	1003	16.700	160.55446
4	1004	16.717	160.55936

```
data['Angle'].plot()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7fe5d2150810>



실습 – Squat 자세 모니터링

```
import numpy as np

Fs = 60 # sampling rate; 60Hz
varCount=0 # variable for count proper squat motion
varHold = np.zeros(len(data['Angle'])) # variable for counting time point holding the squat posture
varDur = [] # list variable for storing posture duration
flag=np.zeros(len(data['Angle'])) # array for flag values per time point

Thres = 150 # arbitrary angle for thresholding; 150 deg

for i in range(len(data['Angle'])):
    if data['Angle'][i]<Thres:
        flag[i]=1
    else:
        flag[i]=0

    varHold[i] = varHold[i-1] + flag[i] # cumulate the flag value

    if varHold[i]==varHold[i-1] and varHold[i]>40: # flag becomes zero and cumulated flag reaches to few seconds,
        varCount+=1
        varDur.append(varHold[i])
        varHold[i]=0

#print(varCount)
#print(varDur)
print("Given Data performed {0:d} times of Squat with correct posture".format(varCount))
averageDur = sum(varDur)/len(varDur)
#print(averageDur/Fs)
print('Average posture duration is {0:.2f} sec'.format(averageDur/Fs))
```

Given Data performed 25 times of Squat with correct posture
Average posture duration is 0.98 sec

