Dimensionality reduction

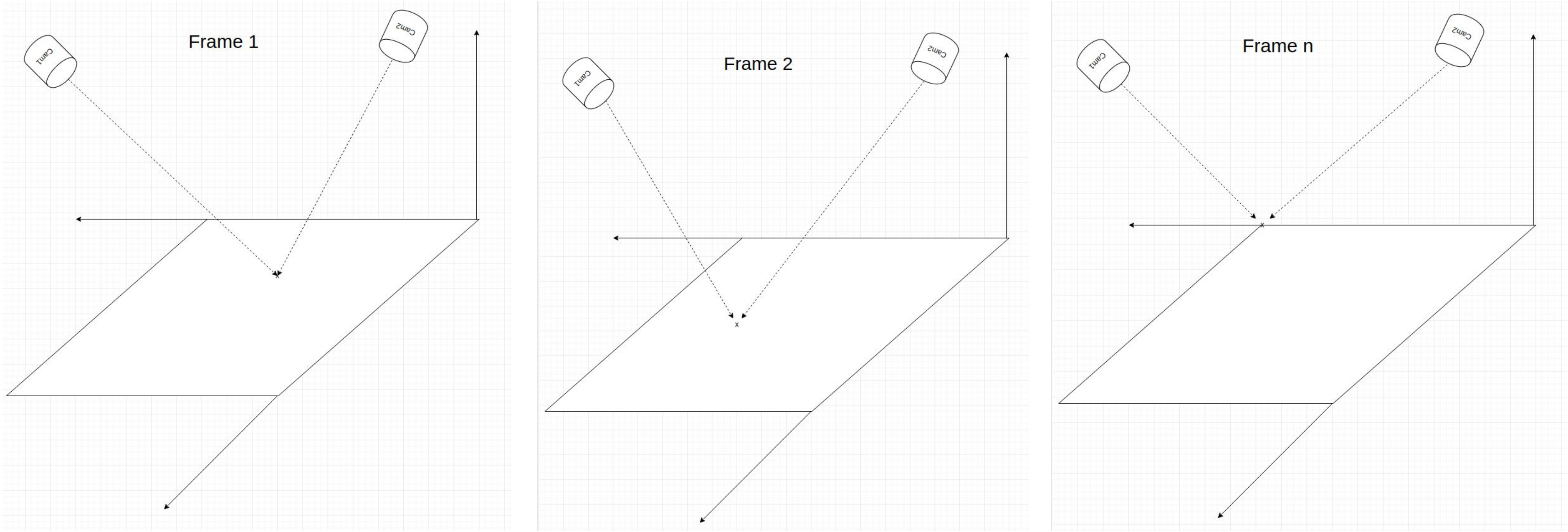
PCA and multiple cameras

A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS

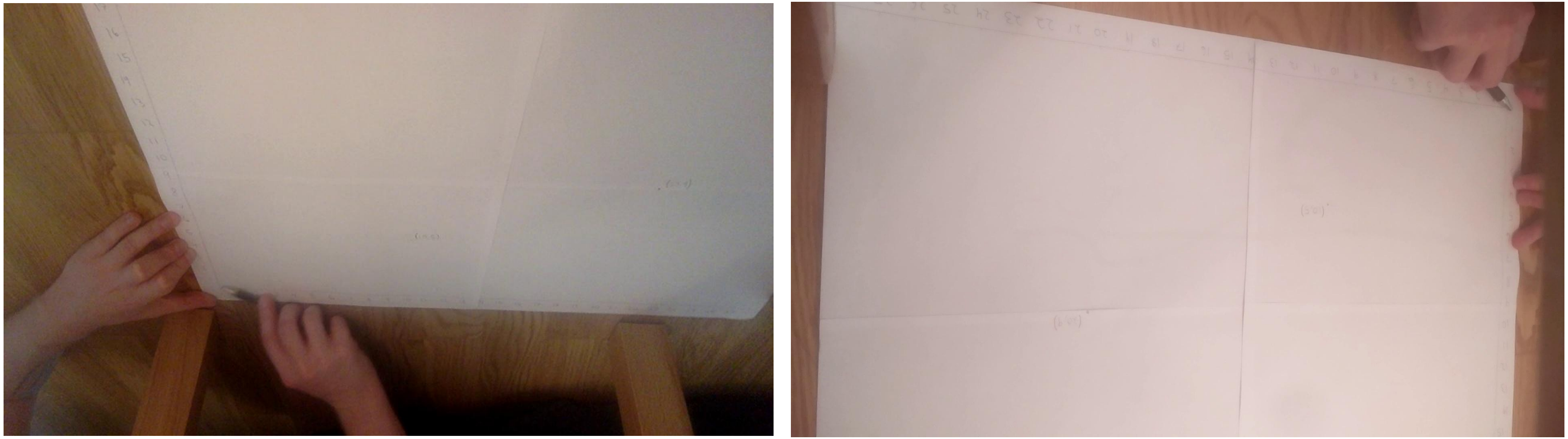
Derivation, Discussion and Singular Value Decomposition

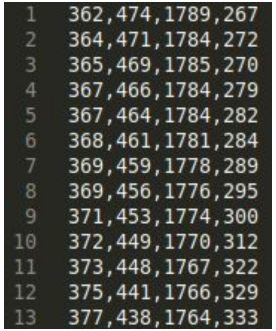
Jon Shlens | *jonshlens@ucsd.edu* 25 March 2003 | Version 1

https://nemenmanlab.org/~ilya/images/b/ba/Shlens-09.pdf


Frame 1
Cam1
Cam2
x
Frame 2
Cam1
Cam2
x
Frame n
Cam1
Cam2
x

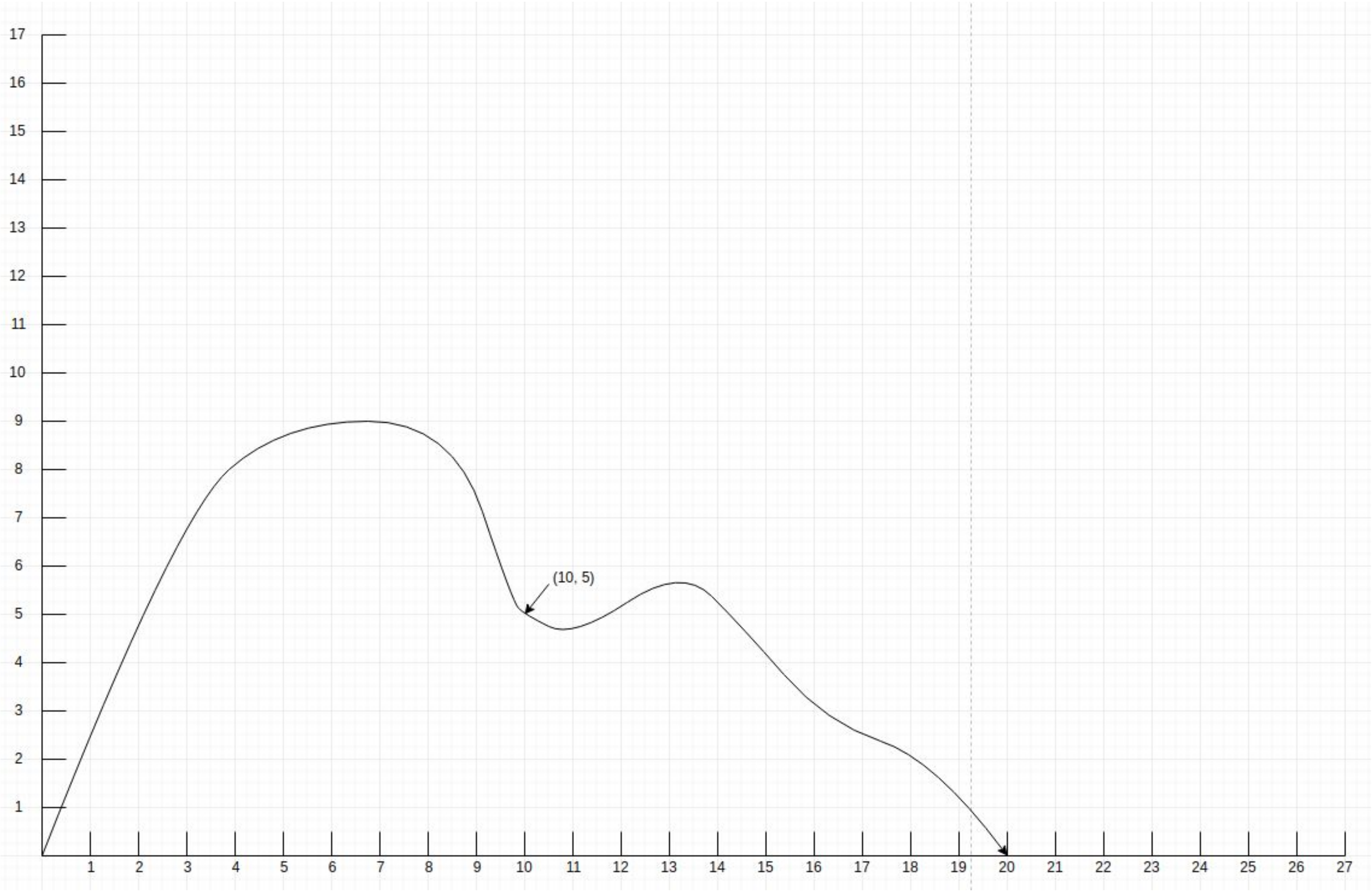
Frame 0 - Cam1 and Cam2



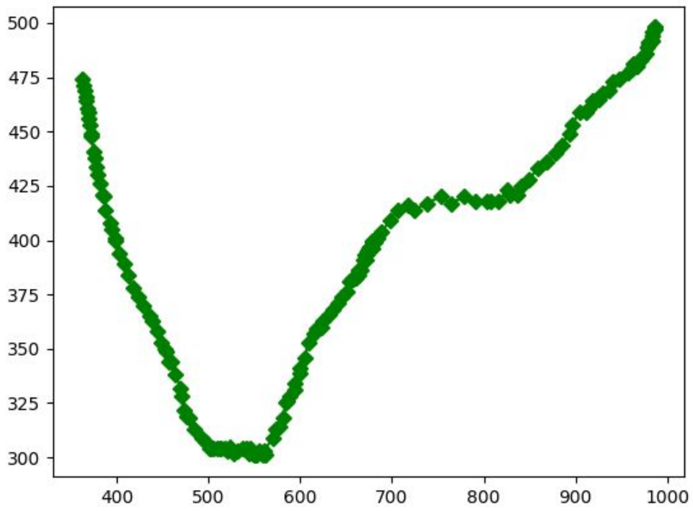


```
362,474,1789,267
364,471,1784,272
365,469,1785,270
367,466,1784,279
367,464,1784,282
368,461,1781,284
369,459,1778,289
369,456,1776,295
371,453,1774,300
372,449,1770,312
373,448,1767,322
375,441,1766,329
377,438,1764,333
```

Cam1 x, Cam1 y, Cam2 x, Cam2 y

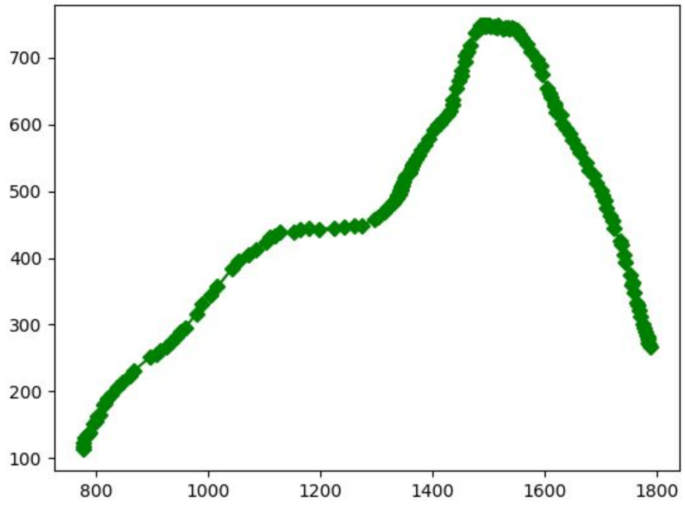

17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
(10, 5)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

Cam1

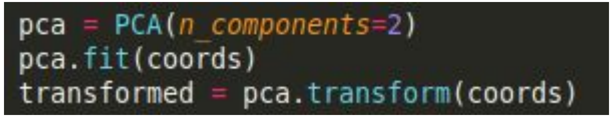


500
475
450
425
400
375
350
325
300
400
500
600
700
800
900
1000

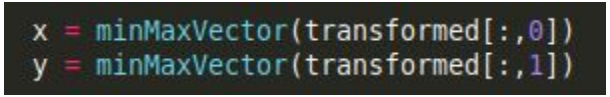
Cam2



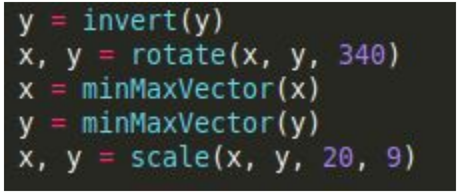
700
600
500
400
300
200
100
800
1000
1200
1400
1600
1800



```python
pca = PCA(n_components=2)
pca.fit(coords)
transformed = pca.transform(coords)
```

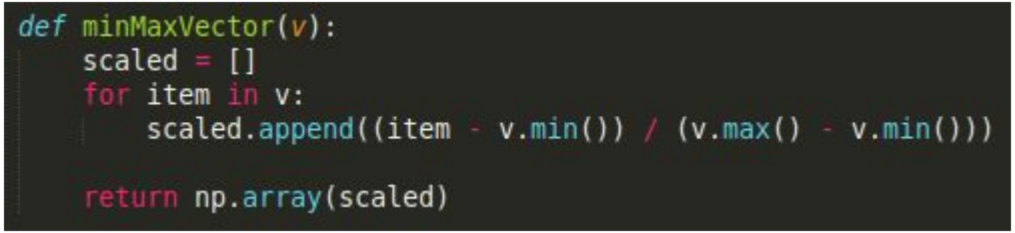


```python
 x = minMaxVector(transformed[:,0])
 y = minMaxVector(transformed[:,1])
```



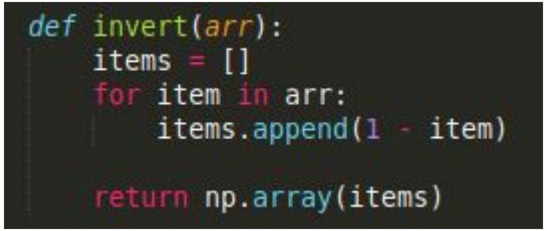
```python
y = invert(y)
x, y = rotate(x, y, 340)
x = minMaxVector(x)
y = minMaxVector(y)
x, y = scale(x, y, 20, 9)
```

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$



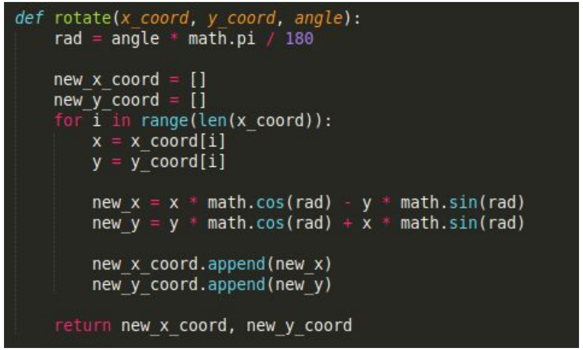
```python
def minMaxVector(v):
    scaled = []
    for item in v:
        scaled.append((item - v.min()) / (v.max() - v.min()))

    return np.array(scaled)
```



```python
def invert(arr):
    items = []
    for item in arr:
        items.append(1 - item)

    return np.array(items)
```



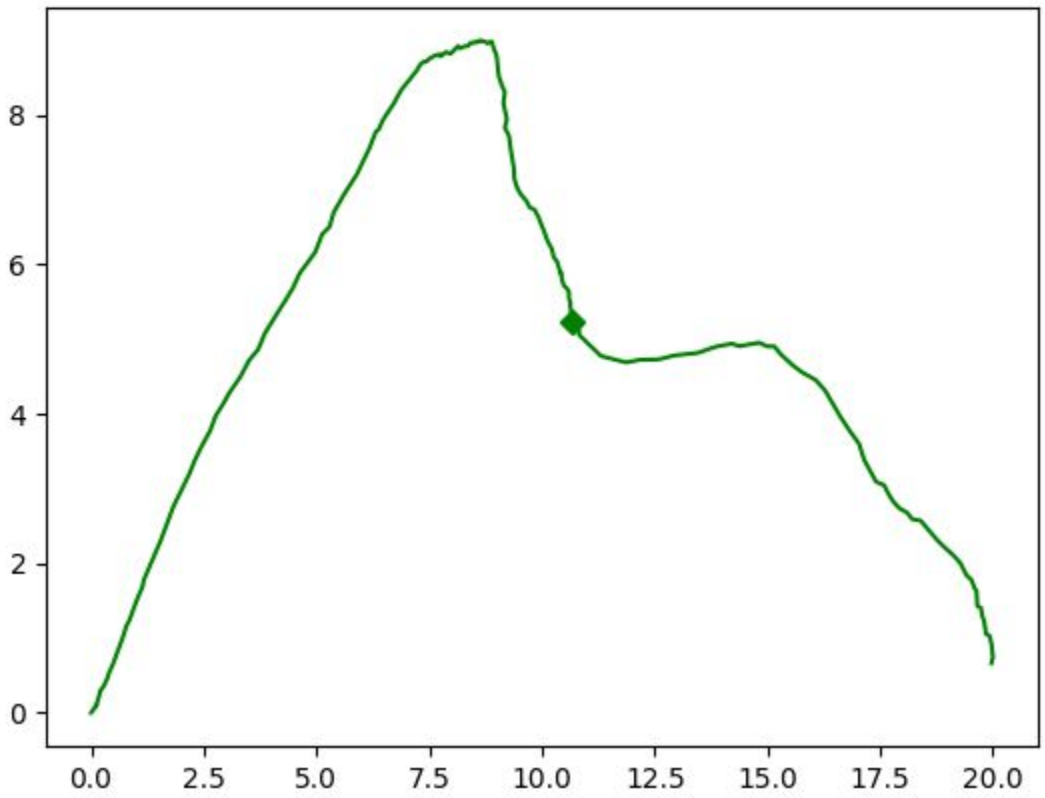
```python
def rotate(x_coord, y_coord, angle):
    rad = angle * math.pi / 180

    new_x_coord = []
    new_y_coord = []
    for i in range(len(x_coord)):
        x = x_coord[i]
        y = y_coord[i]

        new_x = x * math.cos(rad) - y * math.sin(rad)
        new_y = y * math.cos(rad) + x * math.sin(rad)

        new_x_coord.append(new_x)
        new_y_coord.append(new_y)

    return new_x_coord, new_y_coord
```


0
2
4
6
8
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

The repo