libraries

```
import numpy as np
import h5py
import matplotlib.pyplot as plt
```

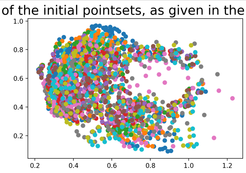
```
C:\ProgramData\Anaconda3\lib\site-packages\h5py\__init__.py:36:
FutureWarning: Conversion of the second argument of issubdtype from
`float` to `np.floating` is deprecated. In future, it will be treated
as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
```

%matplotlib auto

```
n = 2
f = h5py.File('../data/hands2D.mat','r')
shapes = f.get('shapes')
shapes = np.array(shapes)
mean = np.mean(shapes, axis=1)
mean = np.reshape(mean,(40,1,2))
pointSetsCen = shapes - mean
plt.figure(0)
for i in range(40):
    plt.scatter(shapes[i, :, 0], shapes[i, :, 1])
plt.title('Plot of the initial pointsets, as given in the dataset', fontdict = {'fontsize' : 20})

for i in range(40):
    norm = np.linalg.norm(pointSetsCen[i, :, :])
    pointSetsCen[i, :, :] = pointSetsCen[i, :, :]/norm
```

Plot of the initial pointsets, as given in the dataset



mean shape computation

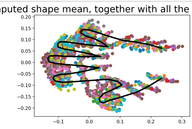
```
mean_shape = np.copy(pointSetsCen[0, :, :])
thresh = 1.e-7
error = 1
while thresh < error:
#for j in range(2):
    for i in range(40):
        a1 = np.matmul(pointSetsCen[i, :, :].T  , mean_shape)
        u, s, vh = np.linalg.svd(a1, full_matrices=True)
    #    d = np.linalg.det(np.matmul(vh, u.T))
        eye = np.identity(n)
        eye[n-1,n-1] = -1
        R1 = vh @ u
        if np.linalg.det(R1)==-1:
            R1 = vh @ eye @ u
        pointSetsCen[i, :, :] =  (R1 @ pointSetsCen[i, :, :].T).T

    best_mean_shape = np.mean(pointSetsCen, axis=0)
    best_mean_shape = best_mean_shape/np.linalg.norm(best_mean_shape)
    error = np.linalg.norm(best_mean_shape - mean_shape)
    mean_shape = best_mean_shape

plt.figure(1)
for i in range(40):
    plt.scatter(pointSetsCen[i, :, 0], pointSetsCen[i, :, 1])
plt.plot(best_mean_shape[:,0], best_mean_shape[:,1],c='black',linewidth=3, markersize=12)
plt.title('Plot of computed shape mean, together with all the aligned pointsets.', fontdict = {'fontsize' : 20})
```

```
Text(0.5,1,'Plot of computed shape mean, together with all the
aligned pointsets.')
```

Plot of computed shape mean, together with all the aligned pointsets.



Variance

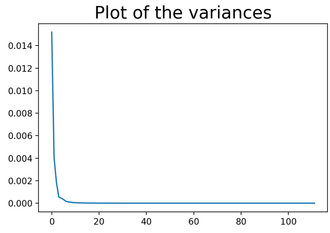
```
pointSetsCenNew = pointSetsCen - mean_shape
pointSetsCenNew = np.reshape(pointSetsCenNew, (40, 112))
covariance = np.cov(pointSetsCenNew.T)
W, V = np.linalg.eig(covariance)
W, V = np.real(W), np.real(V)

plt.figure(2)
plt.plot(W)
plt.title('Plot of the variances',fontdict = {'fontsize' : 20})

#Principal Modes of shape Variation
s1 = np.sqrt(W[0])
s2 = np.sqrt(W[1])

pm11 = mean_shape + 2*s1*np.reshape(V[:, 0],(56,2))
pm12 = mean_shape - 2*s1*np.reshape(V[:, 0],(56,2))

pm21 = mean_shape + 2*s2*np.reshape(V[:, 1],(56,2))
pm22 = mean_shape - 2*s2*np.reshape(V[:, 1],(56,2))
```



1st mode of variation

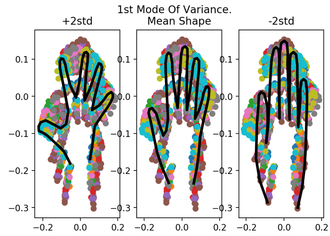
```
f, ((ax1, ax2, ax3)) = plt.subplots(1, 3, sharex='col')
plt.suptitle('1st Mode Of Variance.')

for i in range(40):
    ax1.scatter(pointSetsCen[i, :, 1], -pointSetsCen[i, :, 0])
ax1.plot(pm11[:,1],-pm11[:,0], c='black',linewidth=3, markersize=12)
ax1.set_title('+2std')

for i in range(40):
    ax2.scatter(pointSetsCen[i, :, 1], -pointSetsCen[i, :, 0])
ax2.plot(mean_shape[:,1], -mean_shape[:,0], c='black',linewidth=3, markersize=12)
ax2.set_title('Mean Shape')


for i in range(40):
    ax3.scatter(pointSetsCen[i, :, 1], -pointSetsCen[i, :, 0])
ax3.plot(pm12[:,1],-pm12[:,0],c='black',linewidth=3, markersize=12)
ax3.set_title('-2std')
```

```
Text(0.5,1,'-2std')
```



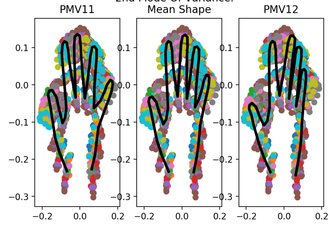
2nd mode of variation

```
f, ((ax1, ax2, ax3)) = plt.subplots(1, 3, sharex='col')
plt.suptitle('2nd Mode Of Variance.')
for i in range(40):
    ax1.scatter(pointSetsCen[i, :, 1], -pointSetsCen[i, :, 0])
ax1.plot(pm21[:,1],-pm21[:,0], c='black',linewidth=3, markersize=12)
ax1.set_title('PMV11')

for i in range(40):
    ax2.scatter(pointSetsCen[i, :, 1], -pointSetsCen[i, :, 0])
ax2.plot(mean_shape[:,1], -mean_shape[:,0], c='black',linewidth=3, markersize=12)
ax2.set_title('Mean Shape')

for i in range(40):
    ax3.scatter(pointSetsCen[i, :, 1], -pointSetsCen[i, :, 0])
ax3.plot(pm22[:,1],-pm22[:,0],c='black',linewidth=3, markersize=12)
ax3.set_title('PMV12')
```

```
Text(0.5,1,'PMV12')
```



Published from human_hand.py using Pweave 0.30.3 on 21-01-2019.