ShapeModel

December 5, 2021

[12]:
```
import numpy as np
import glob
import menpo.io as mio
import matplotlib.pyplot as plt
import cv2
```

[13]:
```
landmarks = sorted(glob.glob("frontalshapes_manuallyannotated_46points/*"))
faces = sorted(glob.glob("frontalimages_spatiallynormalized*/*"))
# for face, file in zip(faces,files):

#     print(face , file)

print(type(faces))
# faces = 200
```

```
<class 'list'>
```

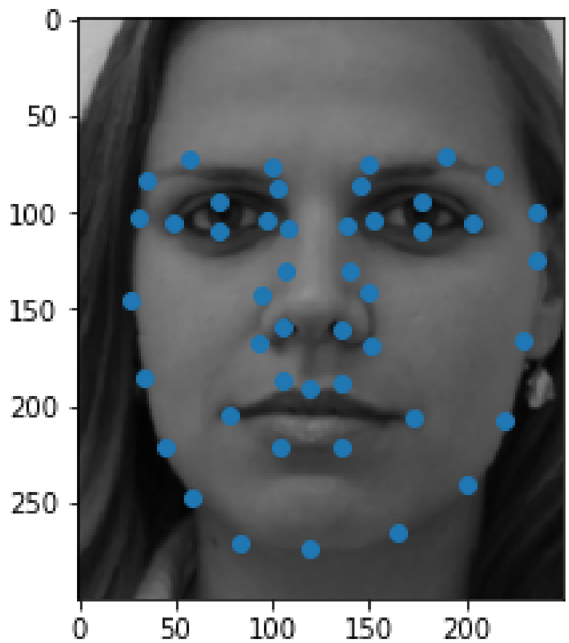
[14]:
```
len(faces)
```

[14]: 400

[15]:
```
test = mio.import_landmark_file(landmarks[0])
print(landmarks[0])
img = cv2.imread(faces[0])
print(test['PTS'].points)

plt.imshow(img)
plt.scatter(test['PTS'].points[:,1], test['PTS'].points[:,0])
# menpo reads the coordinates as ZYX
# plt.gca().invert_yaxis()
plt.show()
```

```
frontalshapes_manuallyannotated_46points/100a.pts
[[159.686  105.447 ]
 [160.683  135.598 ]
 [204.476   77.5498]
 [186.722  105.089 ]
 [190.508  119.702 ]
 [187.499  136.291 ]
```

```
[205.663  172.537 ]
[220.91   135.26  ]
[221.072  104.152 ]
[104.643   48.8948]
[ 94.3955  72.6546]
[103.738   97.1924]
[109.473   71.9728]
[104.342  152.687 ]
[ 94.2905 177.701 ]
[105.282  203.153 ]
[108.826  177.208 ]
[102.835   30.3199]
[145.453   27.0488]
[185.731   33.5807]
[221.349   45.09  ]
[247.785   58.8706]
[270.536   82.8855]
[274.151  118.578 ]
[264.663  164.211 ]
[240.265  200.355 ]
[206.831  219.783 ]
[165.717  229.271 ]
[124.151  236.5   ]
[ 99.472  236.057 ]
[ 83.4876  34.3393]
[ 71.5991  56.7249]
[ 75.731   99.898 ]
[ 86.797  102.897 ]
[ 86.6024 145.771 ]
[ 74.8597 148.976 ]
[ 71.1592 189.967 ]
[ 80.3253 214.813 ]
[107.667  108.521 ]
[130.173  106.194 ]
[142.08    94.5205]
[167.399   93.2719]
[168.258  150.533 ]
[141.116  150.085 ]
[130.609  139.477 ]
[107.021  138.577 ]]
```

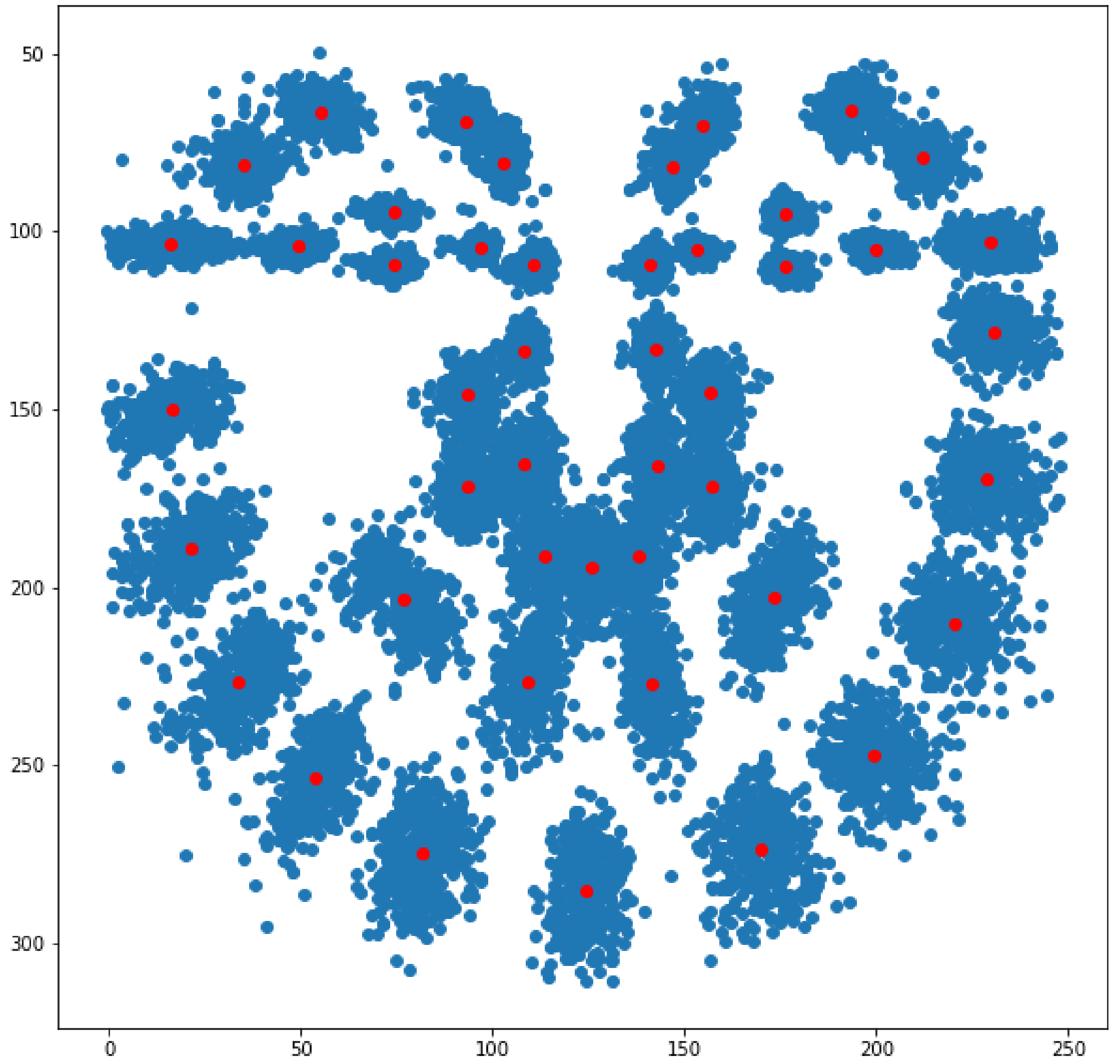


1 Scatter plot of the the landmark distribution together with the mean shape

[61]:

```
landmarkpoints = []
for points in landmarks:
    landmarkpoints.append(mio.import_landmark_file(points)['PTS'].points)

landmarkpoints = np.array(landmarkpoints)

x = landmarkpoints[:,:,1].reshape(-1)
y = landmarkpoints[:,:,0].reshape(-1)

mean = np.zeros([92])

for i in range(46):
    mean[i] = np.mean(landmarkpoints[:,i,1])
    mean[i+46] = np.mean(landmarkpoints[:,i,0])


# print(meany.shape)
# x.shape
# print(landmarkpoints[0])
# print(landmarkpoints[:,:,0])
plt.figure(figsize = (10,10))
```

```
plt.scatter(x, y)
plt.scatter(mean[:46], mean[46:], color = 'r')
plt.gca().invert_yaxis()
plt.show()
```



2 Histogram of eigenvalues as obtained from the SVD of the data covariance matrix

[62]:
```
print(landmarkpoints.shape)
X = landmarkpoints.reshape(400,-1)
print(X.shape)
```

```
(400, 46, 2)
```

```
(400, 92)
```

[63]:
```
# print(X.shape)
CovarMatrix = np.cov(X.T)
CovarMatrix.shape
```

[63]:
```
(92, 92)
```

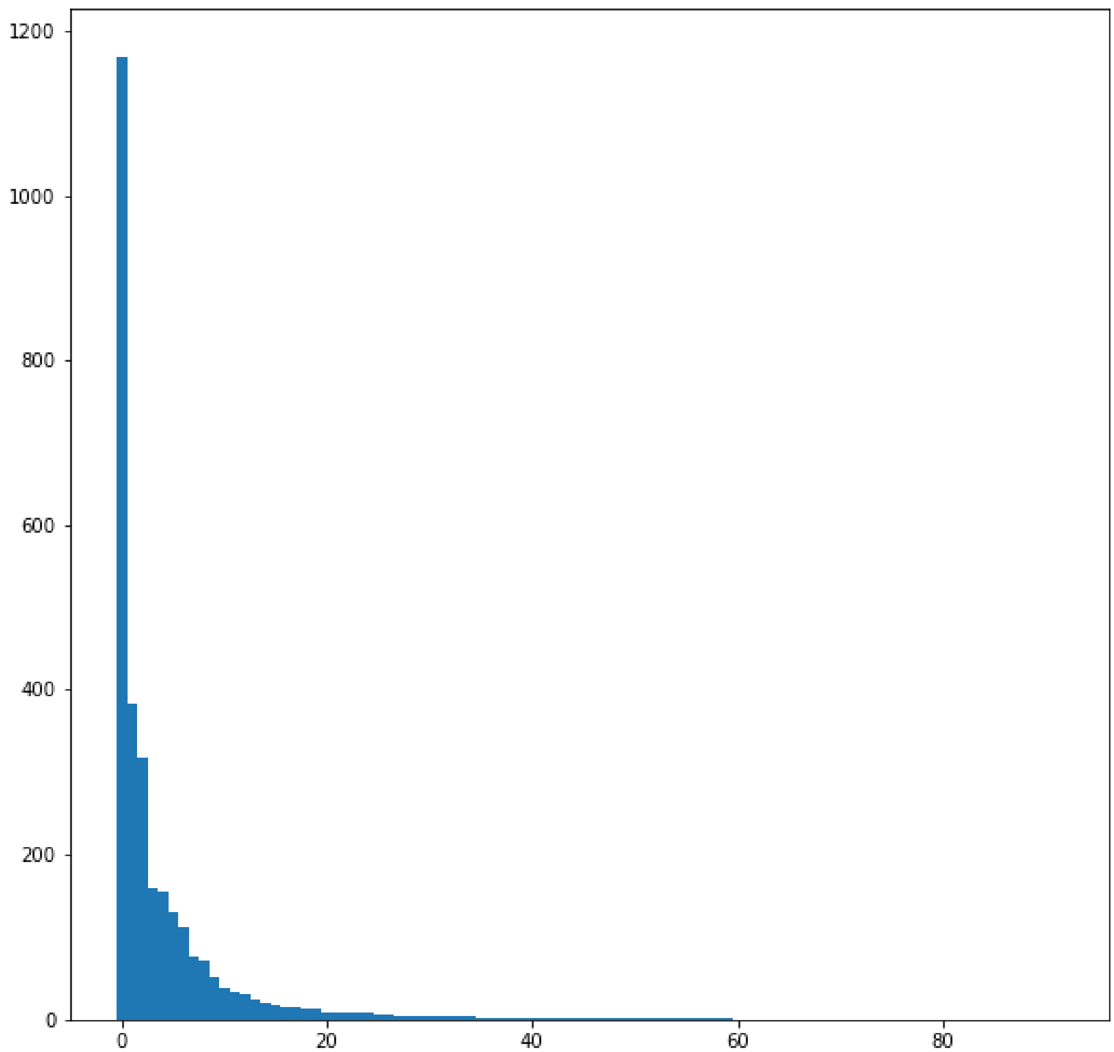
[64]:
```
U,S,VT = np.linalg.svd(CovarMatrix)
eigval, eigenvec = np.linalg.eig(CovarMatrix)
# eigenval = np.vstack([np.arange(0,46,1), eigval])
# eigval
print(eigval.shape)
print(eigenvec.shape)
```

```
(92,)
(92, 92)
```

[65]:
```
print(S.shape)
xlabel = np.arange(0,92, 1)
print("Using eig function:\n", eigval[:10])
print("\nUsing SVD function:\n",S.T[:10])
```

```
(92,)
Using eig function:
 [1167.9764863   382.55719475  318.43578404  160.3621351   154.87818493
  129.79082387  111.62111697   77.10411958   71.91281368   51.92322807]

Using SVD function:
 [1167.9764863   382.55719475  318.43578404  160.3621351   154.87818493
  129.79082387  111.62111697   77.10411958   71.91281368   51.92322807]
```

[66]:
```
plt.figure(figsize=(10,10))
plt.bar(xlabel,S,width=1)
plt.show()
```

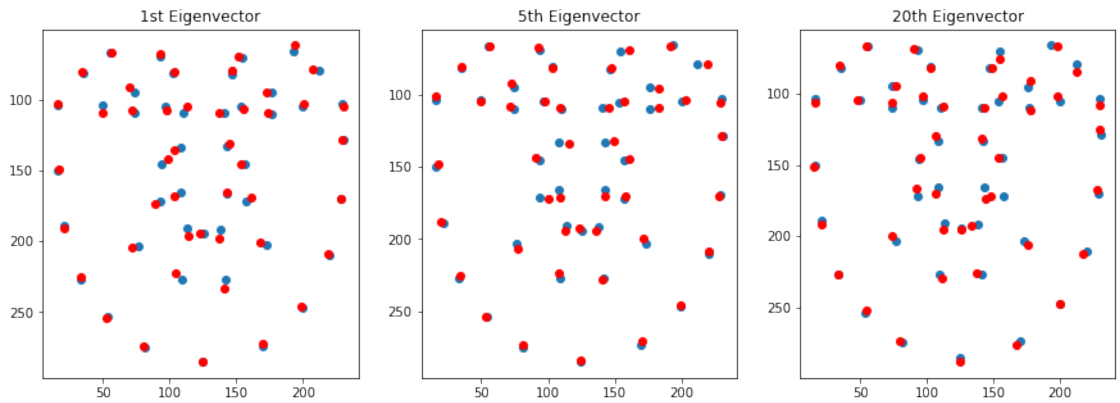


3 Synthetic face shapes by varying the main modes of variation

[113]:
```
var = eigenvec[:,1]*25+ mean
var2 = eigenvec[:,5]*25 + mean
var3 = eigenvec[:,20]*25 + mean

print(test.shape)
fig, axs = plt.subplots(1, 3, figsize=(15,5))

axs[0].set_title('1st Eigenvector')
axs[0].scatter(mean[:46], mean[46:])
axs[0].scatter(var[:46], var[46:], color='r')
axs[0].invert_yaxis()
```

```
axs[1].set_title('5th Eigenvector')
axs[1].scatter(mean[:46], mean[46:])
axs[1].scatter(var2[:46], var2[46:], color='r')
axs[1].invert_yaxis()

axs[2].set_title('20th Eigenvector')
axs[2].scatter(mean[:46], mean[46:])
axs[2].scatter(var3[:46], var3[46:], color='r')
axs[2].invert_yaxis()


# plt.scatter(test2[:46], test2[46:], color='g')
plt.show()
```

```
(92,)
```



[114]:
```
print(eigval.shape)
```

```
(92,)
```

Exaggerated face shapes by varying the main modes of variation beyond the plausible range of variance

[115]:
```
exag = eigenvec[:,1]*eigval[1]*2+ mean
exag2 = eigenvec[:,5]*eigval[2]*2 + mean
exag3 = eigenvec[:,20]*eigval[3]*2 + mean

print(test.shape)
fig, axs = plt.subplots(1, 3, figsize=(15,5))

axs[0].set_title('1st Eigenvector')
axs[0].scatter(mean[:46], mean[46:])
axs[0].scatter(exag[:46], exag[46:], color='r')
```

```
axs[0].invert_yaxis()

axs[1].set_title('5th Eigenvector')
axs[1].scatter(mean[:46], mean[46:])
axs[1].scatter(exag2[:46], exag2[46:], color='r')
axs[1].invert_yaxis()

axs[2].set_title('20th Eigenvector')
axs[2].scatter(mean[:46], mean[46:])
axs[2].scatter(exag3[:46], exag3[46:], color='r')
axs[2].invert_yaxis()


# plt.scatter(test2[:46], test2[46:], color='g')
plt.show()
```

(92,)