Assignment 3

1541447 马天瑶

Task 1

We know that α_1 is the eigen-vector associated with the largest eigen-value λ_1 of C, which is the covariance matrix of X. Also, α_1 is a unit-vector. So we can get that

$$\alpha_1^T\alpha_1 = 1,\ \ C\alpha_1 = \lambda_1\alpha_1$$

Now, we want to find such a unit-vector α_2, which is orthogonal to α_1

$$\alpha_2 = argmax\big(var(\alpha^T X)\big), \alpha \in R^{p\times 1}$$

Meanwhile, we can have the following constraints

$$\alpha^T\alpha = 1,\ \ \alpha^T\alpha_1 = 0$$

We can also get that

$$var(\alpha^T X) = \alpha^T C\alpha$$

Based on Lagrange multiplier method, we need to

$$argmax(\alpha^T C\alpha - \lambda(\alpha^T\alpha - 1) - \phi\alpha^T\alpha_1)$$

So, we differentiate the quantity $\alpha^T C\alpha - \lambda(\alpha^T\alpha - 1) - \phi\alpha^T\alpha_1$ and set the result equals to zero. That is to say,

$$\frac{d(\alpha^T C\alpha - \lambda(\alpha^T\alpha - 1) - \phi\alpha^T\alpha_1)}{d\alpha} = 0$$

$$\Leftrightarrow 2C\alpha - 2\lambda\alpha - \phi\alpha_1 = 0$$

Next, we left multiply α_1 into this expression and we can get that

$$2\alpha_1^T C\alpha - 2\lambda\alpha_1^T\alpha - \phi\alpha_1^T\alpha_1 = 0$$

According to the equations we get previously, we can have a series of equations as below

$$\alpha_1^T C\alpha = \alpha^T C\alpha_1 = \alpha^T\lambda_1\alpha_1 = \lambda_1\alpha^T\alpha_1 = \lambda_1\alpha_1^T\alpha = 0$$

Thus, the expression above turns into $0 - 0 - \phi * 1 = 0$, and therefore, we can see that $\phi = 0$.

Then we can get that

$$C\alpha = \lambda\alpha$$

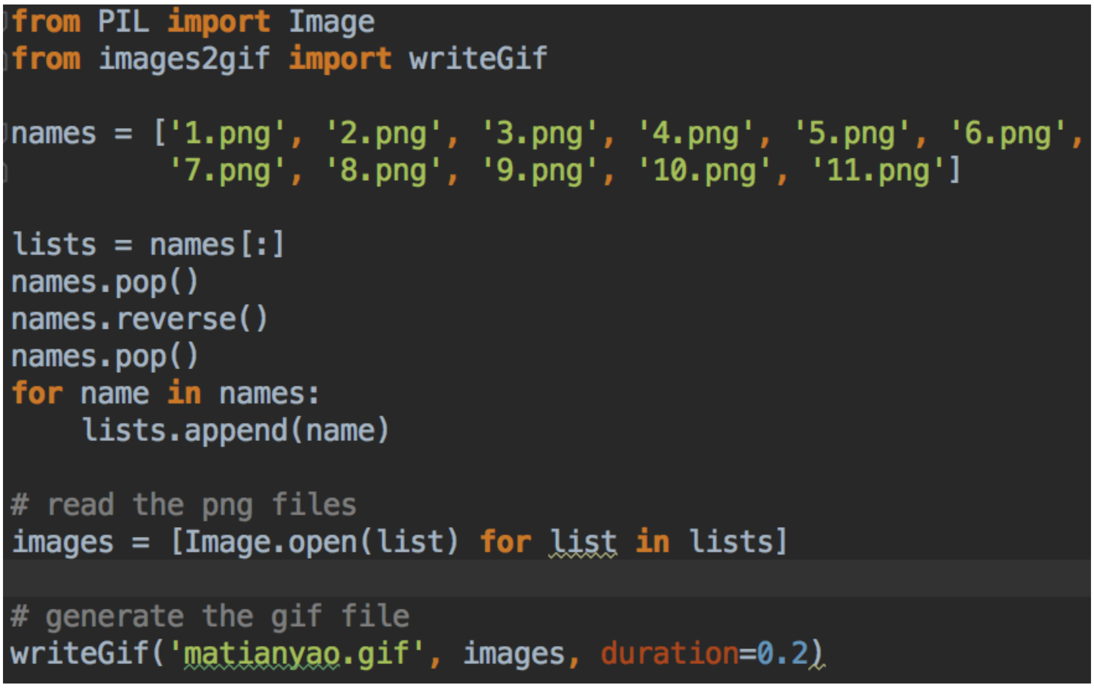
Since

$$\max\big(var(\alpha^T X)\big) = \max(\alpha^T C\alpha) = \max(\alpha^T \lambda\alpha) = \max(\lambda)$$

So we finally get the result that α_2 is the eigen-vector associated with the second largest eigen-value λ_2 of C.

Task 2

After pre-processing the 3D model using Geomagic Studio, I implement the dynamic gravatar with Python 2.7. The code is shown as below:



```
from PIL import Image
from images2gif import writeGif

names = ['1.png', '2.png', '3.png', '4.png', '5.png', '6.png',
         '7.png', '8.png', '9.png', '10.png', '11.png']

lists = names[:]
names.pop()
names.reverse()
names.pop()
for name in names:
    lists.append(name)

# read the png files
images = [Image.open(list) for list in lists]

# generate the gif file
writeGif('matianyao.gif', images, duration=0.2)
```

Here a set of PNG files is read, which are the captures of the 3D model. Afterwards, a Gif file is generated based on some 3rd-party libraries named *images2gif* and *PIL*. Additionally, the value of the parameter *duration* is set to 0.2, which controls the speed of animation.