**Computer Vision Homework 2**

Yao Cai

23533599

**Hybrid Images**

**1\ Code**

import matplotlib.pyplot as plt

from align\_image\_code import align\_images

from scipy import signal

from math import pi, sqrt, exp

from skimage import io

import numpy as np

from mpl\_toolkits.mplot3d import Axes3D

from matplotlib import cm

from skimage import color

import os

**def gaussian\_kernel(sigma, truncate=4.0):**

n = int(2\*sigma\*truncate + 1)

gaussian = np.zeros((n, n))

mean = np.array([n//2, n//2])

for i in range(n):

for j in range(n):

gaussian[i, j] = 1/(2\*pi\*sigma\*\*2)\*exp(-((i-mean[0])\*\*2+(j-mean[1])\*\*2)/2/sigma\*\*2)

return gaussian/gaussian.sum()

**def LoG(sigma, truncate=4.0):**

lowpass = gaussian\_kernel(sigma, truncate)

highpass = np.zeros(lowpass.shape)

i = lowpass.shape[0]//2

highpass[i, i] = 1

highpass = highpass - lowpass

return highpass

**def hybrid\_image(im1, im2, sigma1, sigma2):**

highpass = LoG(sigma = sigma1)

im1\_filter = np.zeros(im1.shape)

for i in range(3):

im1\_filter[:, :, i] = signal.convolve2d(im1[:, :, i], highpass, boundary='symm', mode='same')

im1\_filter = (im1\_filter-im1\_filter.min())/(im1\_filter.max()-im1\_filter.min())

lowpass2 = gaussian\_kernel(sigma = sigma2)

im2\_filter = np.zeros(im2.shape)

for i in range(3):

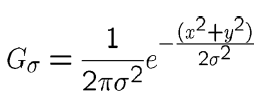
im2\_filter[:, :, i] = signal.convolve2d(im2[:, :, i], lowpass2, boundary='symm', mode='same')

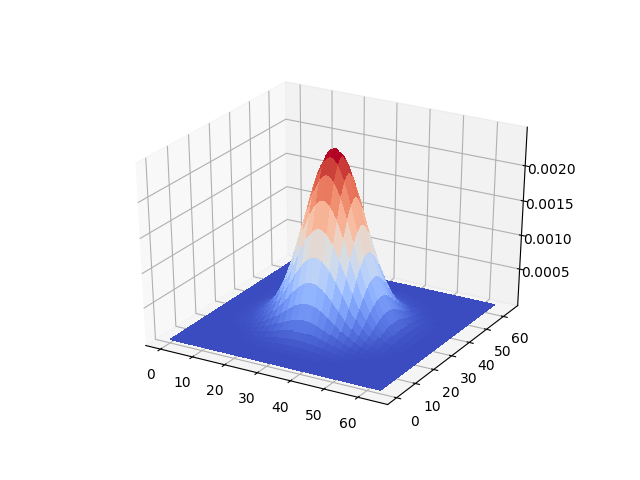
return im1\_filter, im2\_filter, im1\_filter\*0.5+im2\_filter\*0.5

**2\ Results**

Lowpass the first image(convolve with lowpass filter), highpass the second image(convolve with highpass filter), and linearly combine them, so when we take a closer look we recognize the second image and when we take several step back, we recognize the first image.

Lowpass filter: Gaussian, the only parameter is the standard deviation sigma. Larger sigma, more blurred image, which means more high frequency components have been removed. Generally, we want a filtered image with some high frequency removed, but not too much, so that we can still be able to recognize the image. So sigma for lowpass filter can not be too large.

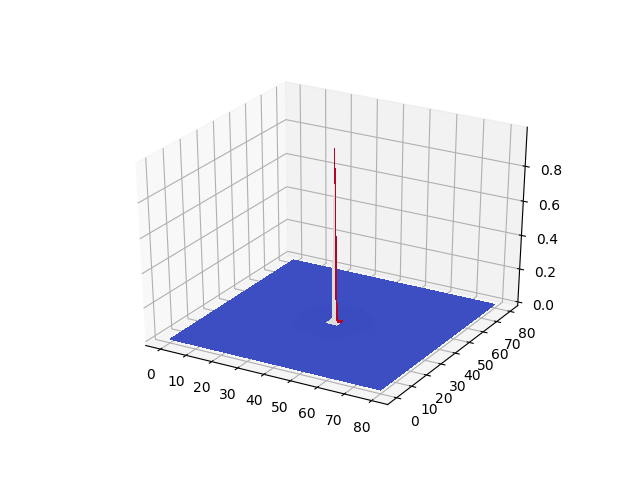
****



Highpass filter:

**M-G**

Where **G** is the Gaussian filter defined above, **M** is a unit impulse filter, a matrix with the same size as G and all zero except the center pixel. The parameter to tune is sigma in G, the larger sigma, sharper the image, which means more high frequency components have been kept. Generally, we want the sigma used in highpass filter **smaller** or equal to sigma used in lowpass filter so the frequency components from the two images doesn’t interfere with each other. (Larger sigma in real space means smaller sigma in frequency space)

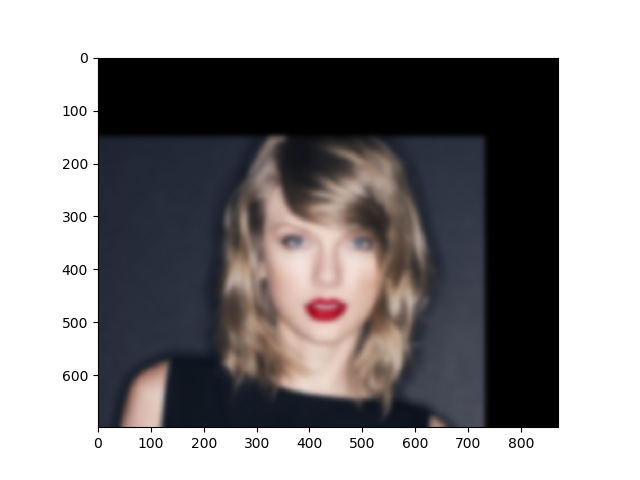
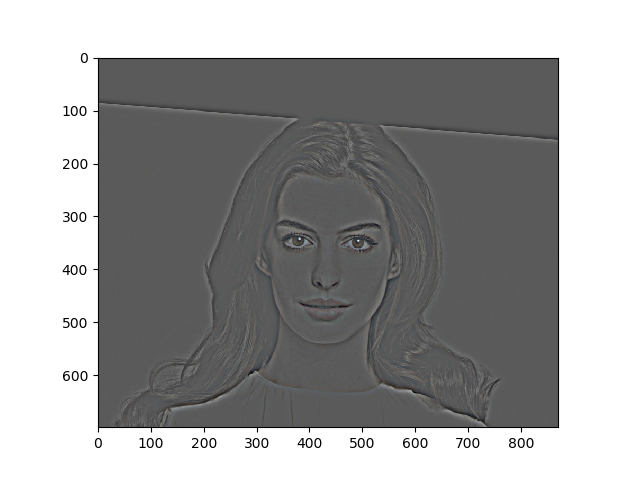


Below is my favorite hybrid image of Taylor Swift and Anne Hathaway:

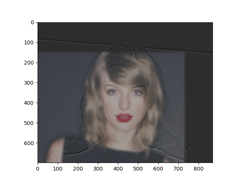
Original Images:

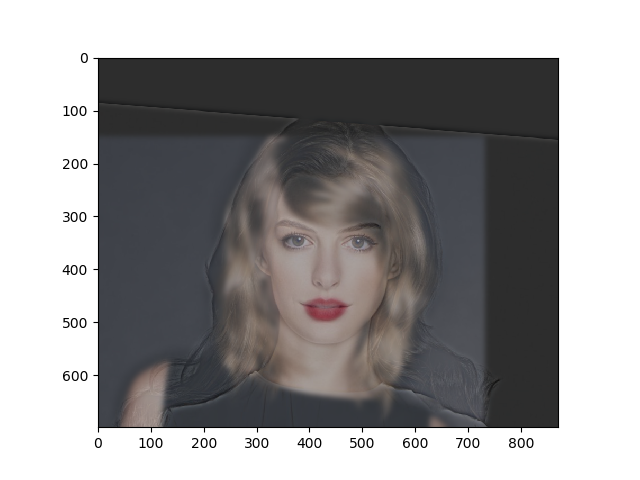
 

Filtered images(highpass sigma=5, lowpass sigma=5):

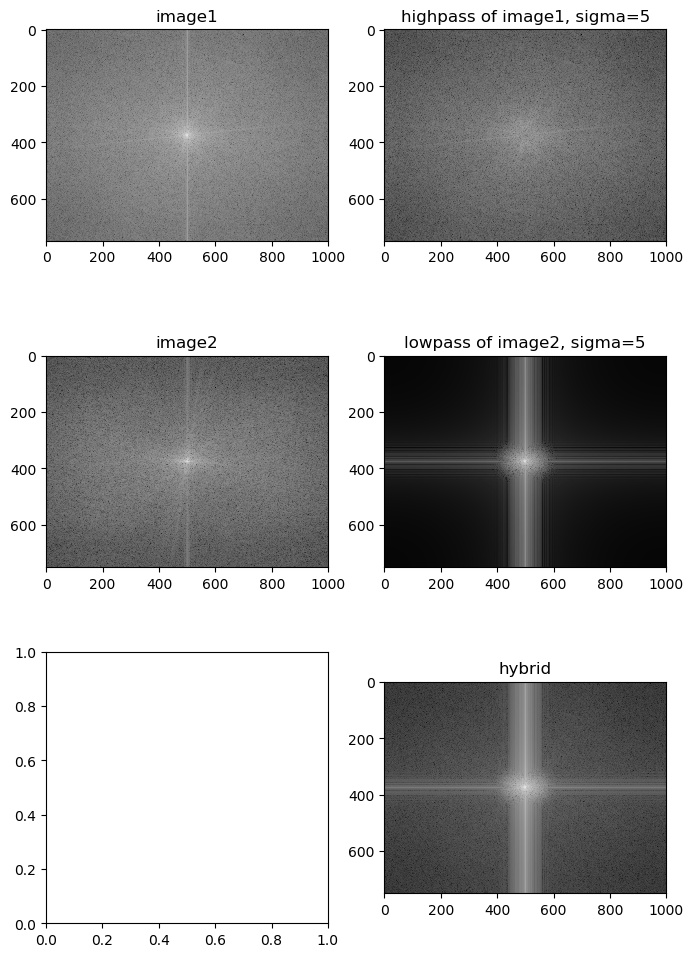


The hybrid(at two scales):





Below is the frequency analysis for both sigma = 5, we can see that indeed highpass filtered image doesn’t have low frequency components, and low pass filtered image doesn’t have high frequency components.



**More results!**

Generally good sigma should be compatible with the features on the images. If the features are small, we would need small sigmas. First choose the sigma for the low pass image, so that even blurred we can still recognize it. Then tune the sigma for the high pass image.

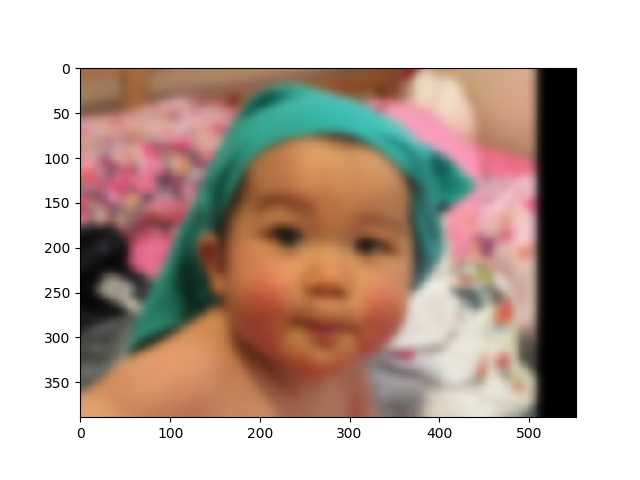
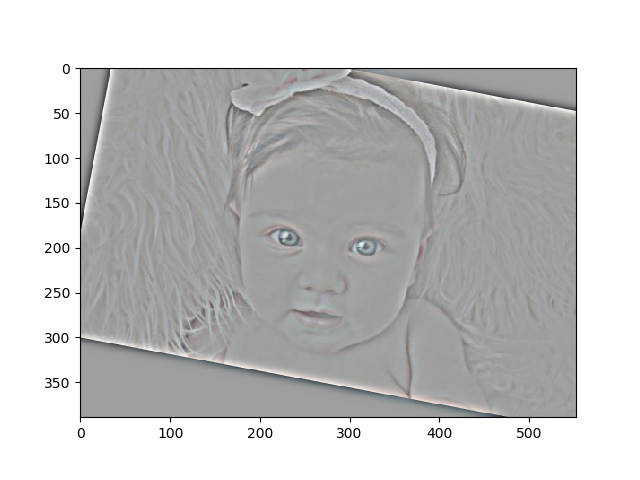
Higher highpass sigma means more low frequency component, and the filtered highpass image is more colorful with higher saturation. This is good for the big image, but not good for the small image, since the low frequency components in highpass image interfere with low frequency components in the lowpass image.

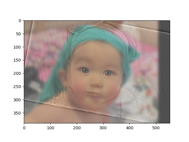
(By small image I mean viewed from a larger distance, and large image I mean viewed from a smaller distance)

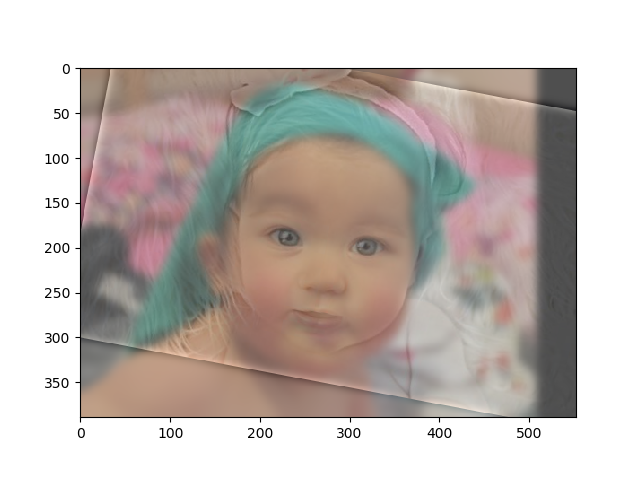
Lower highpass sigma means less low frequency component, this is good for recognition of the small image, but results in the low saturation of the high pass image, so that the colors in the big image mostly come from the low pass image, and when the two images cannot be well aligned, we will end up with a big image not so real.

This is why alignment is so important, alignment means that one feature in one image can be understood as a similar or compatible feature in another image. In all of my images, they generally have “OK” alignment but not perfect. If the alignment is super good, then we can just use a lower highpass sigma, and get both good small and big images, as discussed before. But since in all my images, they are not super good, so the sweet spot that I found for all my hybrid images is that when the sigma of the high pass filter is the same as the low pass filter.

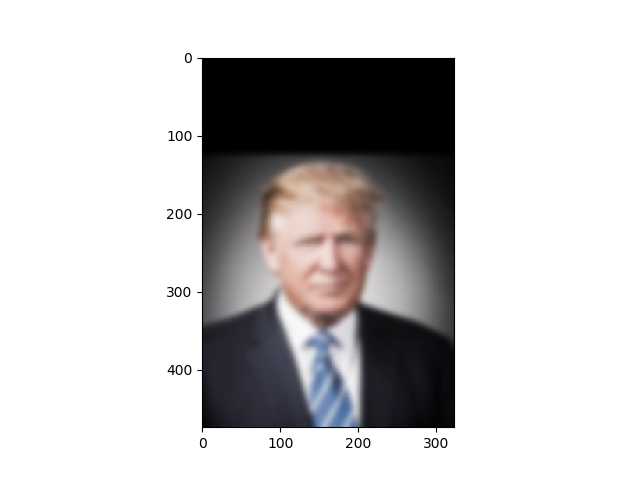
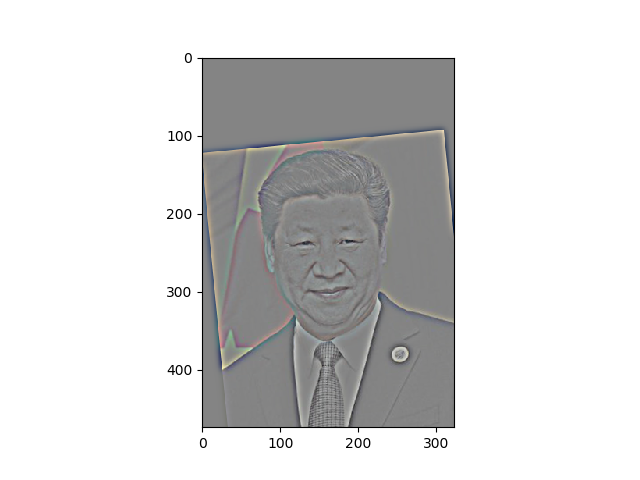
** **

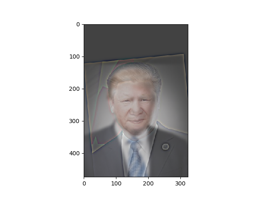
****

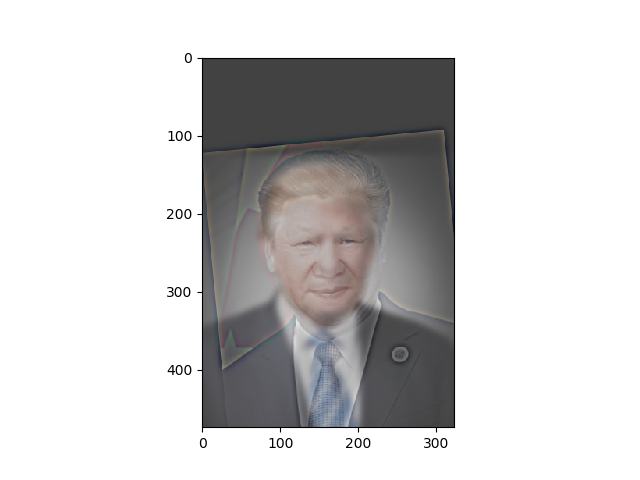
****

****

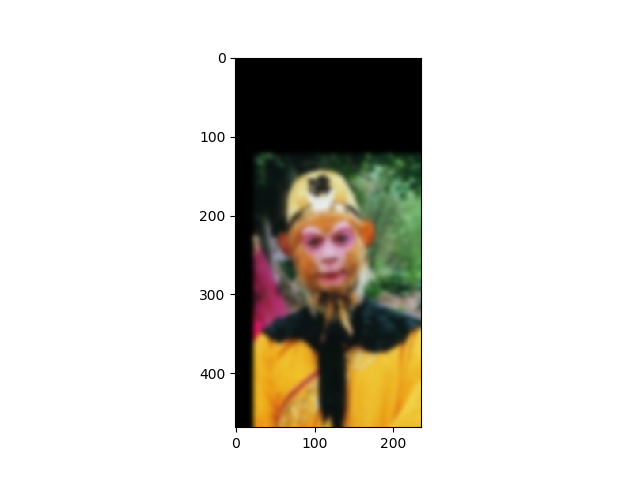
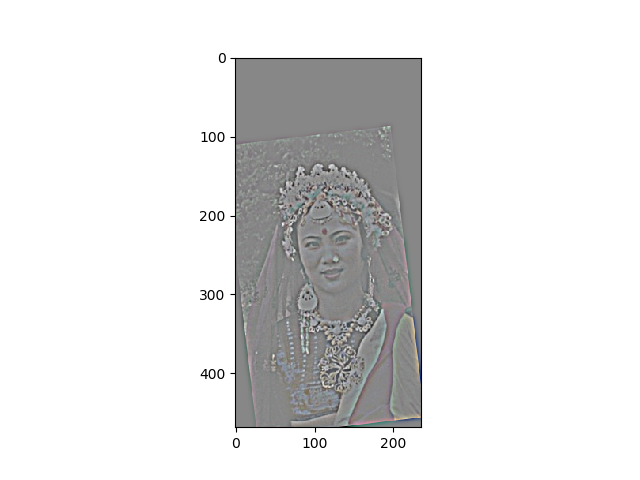
** **

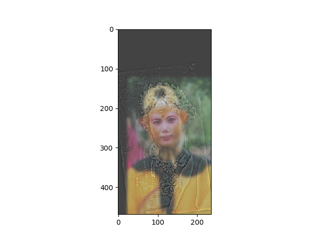
****

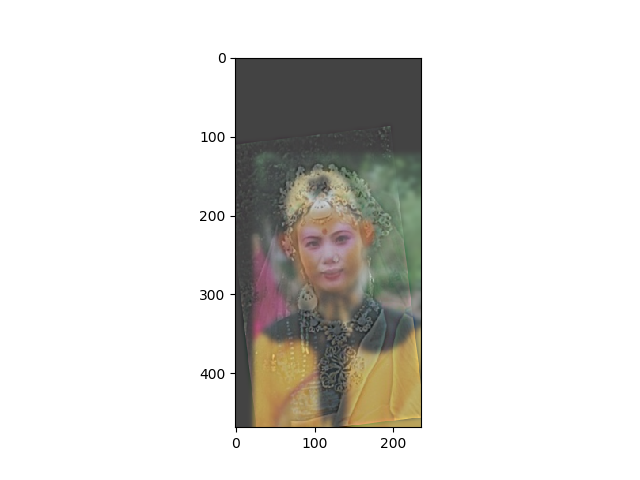
****

****

****

****

****

****