HW02 Report

1. Introduction

The **hybrid image** is created by combining two images, each processed separately through a low-pass filter and a high-pass filter. Low-pass filter removes the high frequency part in the image, resulting in a smoother image and reduces the noise. High-pass filter eliminates the low frequency part. It preserves edges and fine details in the image. By combining the processed images, it reveals the contours from the low-pass processed image and the edge details from the high-pass processed image.

In **task 1**, we implement the hybrid image and deeply understand the method in the process.

The **Gaussian Pyramid** is beneficial for capturing low-frequency information by blurring and down-sampling the image. At the same time, the Laplacian Pyramid isolates high-frequency details such as edges by subtracting successive levels of the Gaussian Pyramid. These pyramids are widely used in image blending, compression, and feature detection.

Task 2 aims to construct and analyze Gaussian and Laplacian Pyramids for a set of images and examine their corresponding frequency spectra.

Template matching is a technique in computer vision used to find small parts of an image that match a template image. It works by sliding the template across the input image (like a window) and comparing the template with the underlying image section to find the best match. This is typically done by calculating a similarity metric, such as correlation or squared differences, at each location.

In **task 3**, we utilized this technique to recover the colorful image from the given 3 channels.

2. Implementation

a. Hybrid image

The hybrid image is created by combining two images, each processed separately through a low-pass filter and a high-pass filter.

In our code, we preprocess the images first. We need to pair the images and ensure that both images have the same size.

After preprocessing the images, we generate the ideal low-pass filter and ideal high-pass filter masks. The formulas shown below.

Ideal low pass filter

$$H(u,v) = \begin{cases} 1, & if \quad D(u,v) \leq D_0 \\ 0, & if \quad D(u,v) > D_0 \end{cases}$$

Ideal high pass filter

$$H(u,v) = \begin{cases} 0, & if \quad D(u,v) \leq D_0 \\ 1, & if \quad D(u,v) > D_0 \end{cases}$$

D_0 is the cutoff frequency, and $D(u, v)$ is the function to calculate the distance from the center, (c_x, c_y), to the point (u, v).

Next, we split the RGB channels of the image to three variables and do the **2D Fourier transform** separately to transform the image from time domain to frequency domain. The formula of the 2D Fourier transform shows below.

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

In the code, we use *np.fft.fft2()* to implement the 2D Fourier transform. Also, we use *np.fft.fftshift()* to center the zero-frequency.

Then, multiply the Fourier transform result of the RGB channels and the filter mask. Sum the low-frequency part of the image and the high-frequency part of the other image in the frequency domain.

After doing the **inverse** of fftshift, *np.fft.ifftshift()*, and the **2D inverse fourier transform**,

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

We calculate the **magnitude** of the results by using *np.abs()*, and **normalize** the value, *cv2.normalize()*, so that the value will be in the range 0 ~ 255. Finally, we **merge** the RGB channels and obtain the hybrid image.

b. Image pyramid

The **Gaussian Pyramid** is created by applying Gaussian blur and down-sampling the image at each level using cv2.pyrDown(). This process is repeated for five levels, progressively reducing the image resolution and high-frequency content. Gaussian blur smooths the image, removing high-frequency details.

The pyramid is constructed using the formula:

$$G_{i+1}(x, y) = DownSample(GaussianBlur(G_i(x, y)))$$

The **Laplacian Pyramid** captures high-frequency details by subtracting the up-sampled version of the next level of the Gaussian Pyramid from the current level:

$$L_i(x, y) = G_i(x, y) - UpSample(G_{i+1}(x, y))$$

The **Discrete Fourier Transform (DFT)** converts the image from the spatial domain to the frequency domain, represented as:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

All the pyramid layers (Gaussian and Laplacian) and their frequency spectra are saved as image files in corresponding directories.

c. Colorizing the Russian Empire

We first cut the image into 3 smaller ones with the same size, representing B, G, and R channels, then we used **template matching** to align the 3 images together. We chose to align the B and R channel to the G channel here.

In the process, we selected the 160*160 pixels square in the middle of the image to be the template to match. We then slide the template in a [±15, ±15] pixels window, trying to find a match.

We chose to use **sum of squared differences (SSD)** as the evaluation metric here, considering both precision and speed. The result of each matching is evaluated by:

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

where smaller is better.

After conducting template matching, we can acquire the offsets of the 2 channels. Then we shift the 2 images accordingly, combine the 3 images all together using **cv2.merge**, and we can get the colorful image.

However, the above pipeline is only suitable for small images (.jpg files provided). Since in larger images, the offset in terms of pixels is significantly larger, we're gonna need a much bigger window, which takes forever to compute.

Instead of decreasing the template size, we chose to do the matching twice, in different image sizes.

For the .tif images, we first scale it down using **cv2.resize**, do the alignment once and get the offsets. Then we do the second alignment on the full-sized image, using the offsets we get from the first alignment times a scale as the initial offset. In this way, the window size can be much smaller, since we have done a rough alignment.

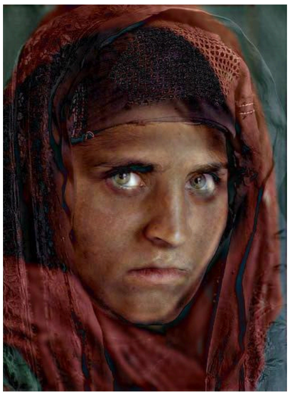
In the end, we can combine the two offsets we get from the 2 alignments, shift the images accordingly, merge them, and we're done.

3. Experimental result

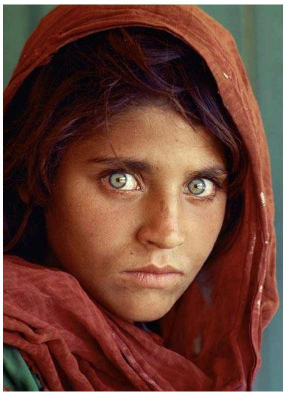
a. Hybrid image

Generate the hybrid image by using the provided data.

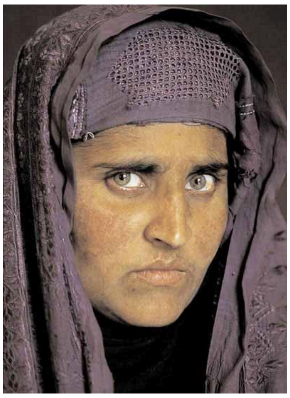
The **left** image is the result, the **ideal low-pass** processed image is in the **middle** and the **ideal high-pass** processed image is on the **right**.



(0_hybrid.jpg)



(0_Afghan_girl_before.jpg)



(0_Afghan_girl_after.jpg)

(6_hybrid.jpg)

(6_makeup_before.jpg)

(6_makeup_after.jpg)

It can be observed that when we look closer, the image still shows the high-frequency details. From a distance, the low-frequency information prevails.

Generate the hybrid image by using the provided data.

The **upper** image is the result, the **ideal low-pass** processed image is on the **left** and the **ideal high-pass** processed image is on the **right**.

(0_hybrid_mydata.jpg)

(0_killua.jpg)

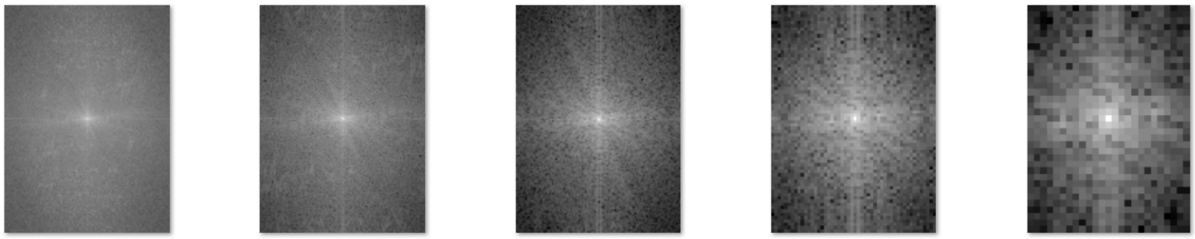
(0_gojo.jpg)

b. Image pyramid

The Gaussian Pyramid construction demonstrated a progressive smoothing of the original images at each level.

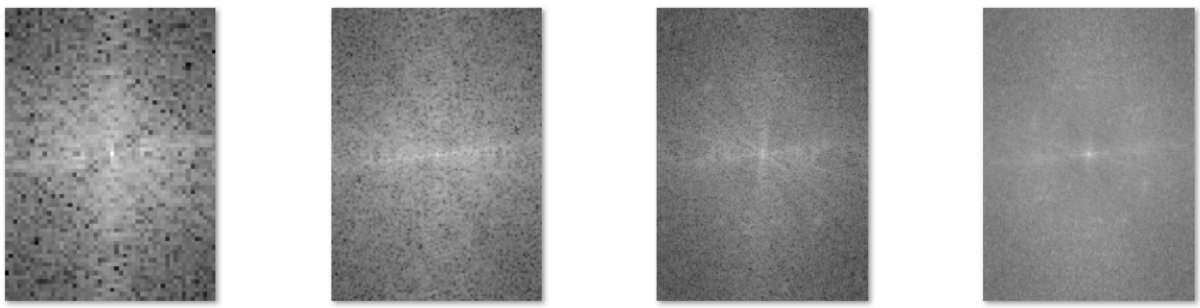
The Laplacian Pyramid highlighted the high-frequency details across the images, isolating the edges and fine structures at each level.

Gaussian Pyramid Spectra:
The spectra showed a gradual reduction in high-frequency content as the levels progressed. At Level 0, the original image retained a wide range of frequencies, with sharp transitions between light and dark areas. As we moved to Levels 1, 2, 3, and 4, the high-frequency components gradually disappeared, leaving only low-frequency components by the final level.



Gaussian_spectrum_level_0 Gaussian_spectrum_level_1 Gaussian_spectrum_level_2 Gaussian_spectrum_level_3 Gaussian_spectrum_level_4

Laplacian Pyramid Spectra:
In contrast, the Laplacian Pyramid spectra captured the concentration of high-frequency components at the upper levels, corresponding to sharp edges and fine transitions in the images. As the pyramid levels deepened, the frequency spectra indicated fewer high-frequency elements, consistent with reduced fine detail and isolation of broader transitions in the images.



Laplacian_spectrum_level_0 Laplacian_spectrum_level_1 Laplacian_spectrum_level_2 Laplacian_spectrum_level_3

General Observations:

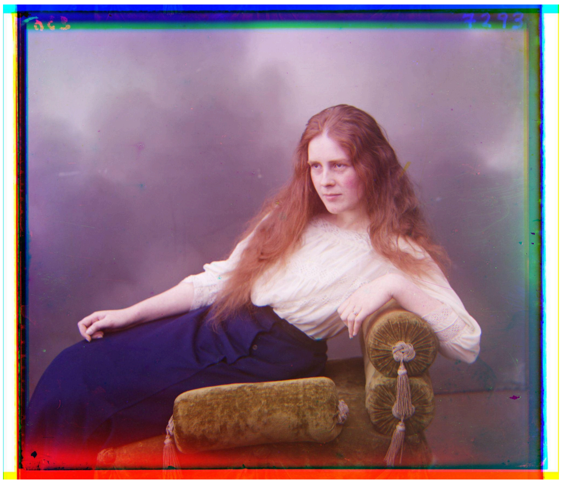
- Faces (such as those in portraits of people like the Afghan Girl or Marilyn Monroe) saw clear isolations of facial contours, hair strands, and fine wrinkles in the Laplacian Pyramid. In the Gaussian Pyramid, facial details gradually softened, but the overall structure of the face remained visible through all levels.

- Objects (such as the bicycle and airplane) exhibited strong edge isolation in the Laplacian Pyramid, where the object contours were separated from the background. The objects' shape and structure were maintained at lower levels in the Gaussian Pyramid, even as the fine details were lost.
- Landscapes and Textures (such as the bird, plane, and fish images) showed a significant smoothing of textures in the Gaussian Pyramid, but their overall shapes and boundaries were still discernible. In the Laplacian Pyramid, the fine textures of feathers, wings, and water were effectively captured at the upper levels.

c. Colorizing the Russian Empire

Example results of small images (.jpg):


Example results of large images (.tif):

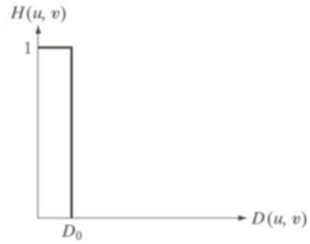
4. Discussion

a. Hybrid image

Compare the ideal low-pass filter and the Gaussian low-pass filter, the formulas of them show below.

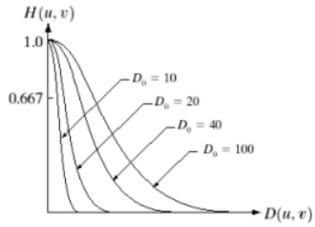
Ideal low-pass filter

$$H(u,v) = \begin{cases} 1, & if \quad D(u,v) \le D_0 \\ 0, & if \quad D(u,v) > D_0 \end{cases}$$

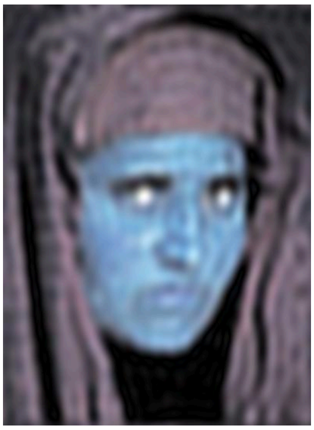


Gaussian low-pass filter

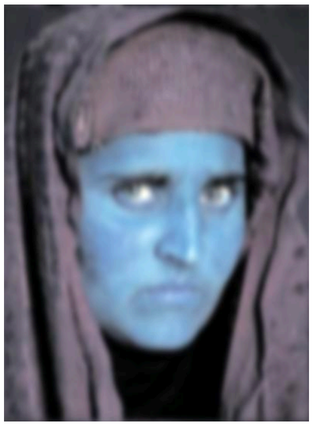
$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$



The frequency in a Gaussian low-pass filter will reduce slowly, however, the frequency in an ideal low-pass filter will cut off immediately. In the images, it shows that the image processed by the Gaussian low pass filter is smoother than by the ideal low pass filter. There are some dark lines on the image processed by the ideal low-pass filter.



(0_idealLpf.jpg)



(0_gaussianLpf.jpg)

Compare the ideal high-pass filter and the Gaussian low-pass filter, the formulas of them show below.

Ideal high-pass filter

$$H(u,v) = \begin{cases} 0, & if \quad D(u,v) \le D_0 \\ 1, & if \quad D(u,v) > D_0 \end{cases}$$

Gaussian high-pass filter

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

The frequency in a Gaussian high-pass filter will increase slowly, however, the frequency in an ideal high-pass filter will increase immediately. In the images, it shows that the image processed by the ideal high-pass filter retains more edge detail compared to the Gaussian high-pass filtered image.

(0_idealHpf.jpg)

(0_gaussianHpf.jpg)

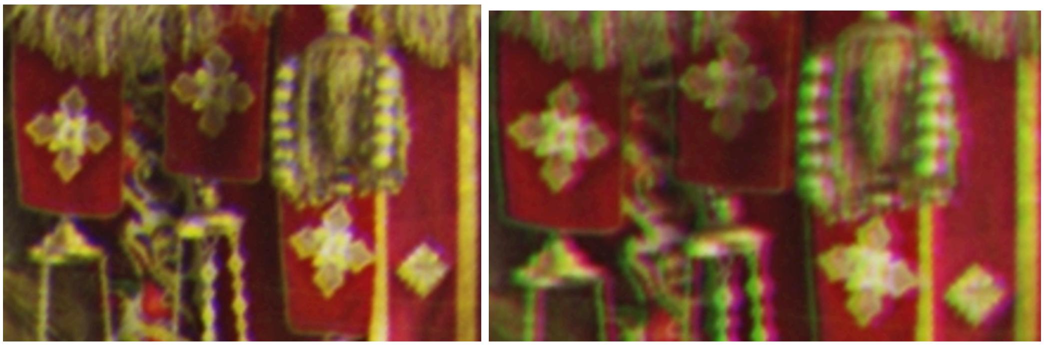
b. Image pyramid

This experiment highlighted the effectiveness of image pyramids in multi-scale representation. The Gaussian Pyramid progressively smoothed the image, effectively retaining low-frequency information. In contrast, the Laplacian Pyramid isolated and captured high-frequency components, concisely representing edges and fine details. By analyzing the frequency spectra of each level, we observed a clear transition in frequency content: high-frequency components were filtered out in the Gaussian Pyramid. This ability to separate frequency components allows image pyramids to be used in tasks such as image blending, image compression, and feature detection.

c. Colorizing the Russian Empire

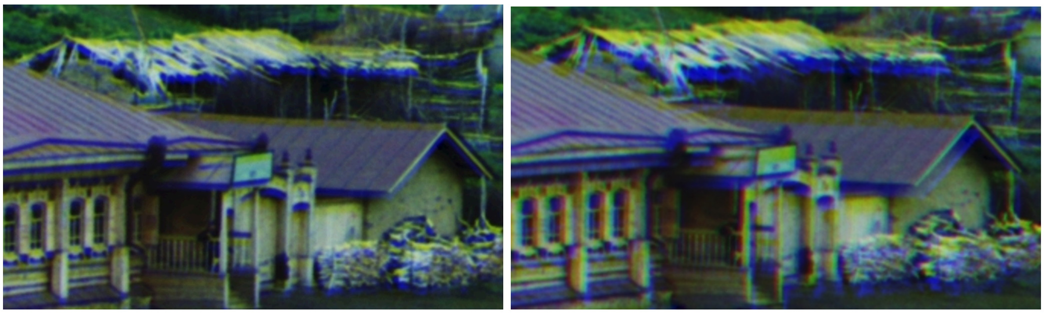
During testing, we used **cv2.matchTemplate** to see if our implementation is correct, and we found something strange. When we try to align the R, G channels to the B channel, in most images, it works fine. However, in some cases, it always fails to align one of the channels (we forgot which one) to the B channel, even using **cv2.matchTemplate**. When we switched the evaluation metric of **cv2.matchTemplate** from SSD to other options, some of them successfully aligned the channels, but some of them just still failed. Since we had totally no clue why this didn't work, we then tried to align the R, B channels to the G channel. Surprisingly, everything works fine now. It turns out that some relation between channels made the algorithm fail.

As we mentioned in previous parts, we use 2 alignments to handle large images. To see if the second alignment actually works, we also compared it with the results of doing only 1 alignment, the examples are as follows:



(the result of 2 / 1 alignment on icon.tif)

(the result of 2 / 1 alignment on lady.tif)



(the result of 2 / 1 alignment on village.tif)

(the result of 2 / 1 alignment on workshop.tif)

We can clearly see that, although there are still some rooms for improvement, the second alignment indeed helps.
In the second alignment, we also used a smaller window size, to both speed up and prevent the result from being worse than the one with only one alignment if matching failed.

5. Conclusion

The hybrid image technique combines low-pass and high-pass filtered images, creating a dynamic effect that changes with the viewer's distance. And the image will have different changes depending on the cut off frequency and the filter we used. This flexibility allows for a range of visual effects that can effectively convey different details and features.

Task 2 successfully demonstrated the construction and analysis of Gaussian and Laplacian Pyramids for diverse images. The Gaussian Pyramid captured the low-frequency content, while the Laplacian Pyramid isolated the high-frequency details such as edges and textures. Additionally, the frequency spectra demonstrated how the image's frequency content is progressively filtered and decomposed at different pyramid levels.

Template matching plays a crucial role in finding similar portions of two images, and choosing the right evaluation metric is another important problem. Some might be faster, and some might be more accurate, choose them wisely depending on the problem or your need.

6. Work assignment plan between team members.

彭宇楨: 33%, 洪義翔: 33%, 張宥楨: 34%