Image Processing - Exercise 3

Shir Rashkovits, shirashko, 209144013

Introduction

The primary goal of this exercise is to explore the usage of frequency domain in images and its ability to process images, specifically by the tasks of image blending and hybrid image creation. Image blending aims to seamlessly merge two images into one cohesive unit, while hybrid image creation seeks to combine two images in such a way that one image dominates at a close viewing distance, and the other becomes prominent when viewed from afar.
The central technique employed to achieve these objectives is the manipulation of image frequencies. In the first task, construction and manipulation of image pyramids, specifically, Gaussian and Laplacian pyramids, gives an intermediate way to manipulate frequencies while technically staying in the spatial domain. In the second task, explore the characteristics of high versus low frequencies effect on the spatial domain visual effect, and how to merge these distinct attributes to create a new image carrying this special characteristic from each image, using high-pass and low-pass filtering in the frequency domain.

Algorithm

1.2 Image Blending Algorithm:
1. Initialization: - Load two target images (a, b) and a binary mask image (m).
   - Assumption: all images are of the same size, N×M. - Define K as the minimum of N and M.
2. Gaussian Pyramid Computation:
   - For each image and the mask, create Gaussian pyramids (Ga, Gb, Gm).
   - Start with G0 as the original image.
   - Iteratively blur and downsample each level to construct the pyramid until reaching the smallest level determined.
3. Laplacian Pyramid Construction:
   - Build Laplacian Pyramids (La, Lb) for both images with parameter chosen log2(K).
   - For each level, subtract the expanded version of the next Gaussian level from the current one until it reaches the top of the pyramid.
4. Blending Laplacian Pyramids:
   - Construct a new Laplacian Pyramid (Lc) by blending corresponding levels of La and Lb using Gm as weights: for each pixel (i, j), calculate Lc(i, j) as Gm(i, j) * La(i, j) + (1 - Gm(i, j)) * Lb(i, j).
5. Image Reconstruction: reconstruct the blended image by summing all levels of Lc.
6. Output: return the final blended image.

Implementation Overview:
All manipulations are done with float and at the end return the image to uint8 type.
Image Loading: Utilized OpenCV's cv2.imread to load RGB images and a mask.
Normalization: Converted the mask's intensity values from the standard [0, 255] range to a binary [0, 1] scale for blending calculations.

Pyramid Levels: Determine the number of levels using numpy based on the image size.
Gaussian Pyramid: Created using OpenCV's cv2.pyrDown, which applies Gaussian blurring followed by downsampling.
Laplacian Pyramid: Formed by subtracting an upsampled Gaussian pyramid level from the current level, using OpenCV's cv2.pyrUp for upsampling.
Blending Process: Performed by iterating through each pyramid level and blending using the mask's Gaussian pyramid as weights.
Reconstruction: Combined the blended Laplacian pyramid levels into a single image using numpy addition.

Images used: (diagonal mask for bad result)

The algorithm approach is grounded in the fundamental concept that images can be dissected into distinct layers of detail by the frequencies component in the spatial domain, and then reassembled to accentuate various features across different scales. This algorithm utilizes this in order to make a seamless pass between the images. The blend is executed in a manner where higher frequencies, denoting finer details, are merged using a more binary approach, ensuring sharp transitions, while lower frequencies, which encapsulate broader, smoother variations, are combined in a softer, more gradient fashion. This gradient blend aligns with the Gaussian blur effect applied at the higher levels of the pyramid, resulting in a subtle and natural transition within the composite image.

1.2 Hybrid Images Algorithm:
1. Load Images: Import the two images for close and far viewing. Convert to grayscale.
2. Fourier Transform: Apply FFT to both images.
3. Centralize Frequency Domain: Shift zero-frequency to the center.
4. Generate Low-Pass Filter: Use sigma to define the Gaussian filter.
5. Create High-Pass Filter: Subtract the low-pass filter from an all-pass filter.
6. Apply Filters: Multiply the close image by the high-pass filter and the far image by the low-pass filter.
7. Combine Images: Sum the filtered images.
8. Reverse the centralization and apply Inverse FFT to return to the spatial domain.
9. Adjust Image: Compute the magnitude and clip values to [0, 255].
10. Output: Return the hybrid image.

Implementation Details:
- FFT and IFFT: Utilize np.fft.fft2 and np.fft.ifft2 for transforming images.
- Filter Design: Implement Gaussian filtering in the frequency domain, considering the image size for filter dimensions. Using sigma as parameter, needed to be adjusted according to the specific images used (for different pairs, different sigma suits better).
- Filter Application: Element-wise multiplication in the frequency domain for applying filters.

- Magnitude Calculation: Use np.abs to compute the magnitude of the complex-valued inverse FFT result, to get real intensity values.
- Clipping: Ensure pixel intensities are within the valid range using np.clip.

Images used: (Dog in close look, cat when viewed from afar) (rightest for the bad result)

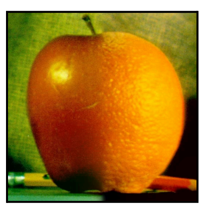
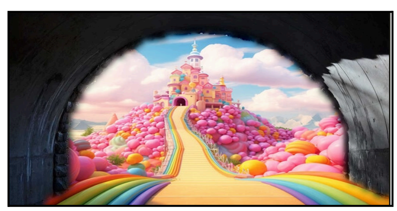


High frequencies cannot be expressed in small images, so the algorithm exploits the phenomenon that higher frequencies become less perceptible as the image size reduces. It also utilizes the fact that general prominent patterns dominate view while fine details are expressed by high frequencies. By using complement gaussian for the “high pass filter”, we ensure that the result will not only compose of very low and very high frequencies, but have all frequencies, with some threshold determined by the sigma value of the low pass filter. By manipulating the sigma value, which dictates the spread of the frequencies from each image, we can selectively combine different frequency ranges from each image. This control enables the creation of a hybrid image that reveals different aspects when viewed at varying distances or sizes.

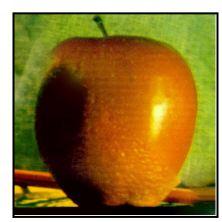
The difference between the two algorithms: The key difference lies in the approach- the first uses spatial domain multi-scale pyramids to blend images, while the second uses frequency domain filtering to create an image with dual viewing properties. Each method leverages the strength of its domain- spatial for multi-scale blending and frequency for selective frequency emphasis or suppression.

Results:

Bending results:

Intermediate bad results: unsuccessful blendings The left image is the first result I got when trying to make a hybrid image of a dog and a cat. Since the dog image also shows its body and not only the face of the animal, it is more noticeable in each point of view that it is not a natural image and that something is off. In contrast, when I usd the image presented before, the dog image also included only the dog face, and since the two animals are pretty similar (furry, flat nose), in each point of view the other animal features which weren’t dominant didn’t interfere as much.

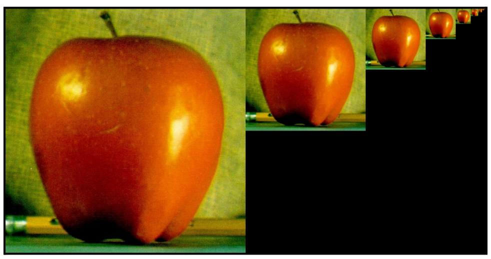



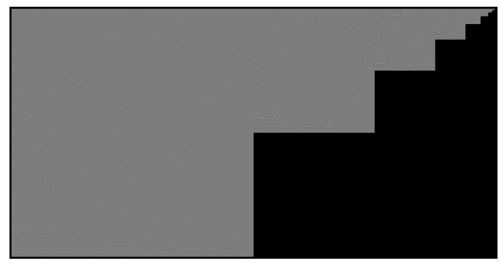
In the second bad result, I tried to apply a diagonal binary mask (instead of a middle mask). In contrast to the way the apple and the orange are aligned in size and lighting in the middle, the images do not align well in lighting and in the width and height image all across the diagonal.

We can see that in the edges of the fruits the apple width is smaller and higher than of the orange. Since the human eye is very sensitive to edges, and the diagonal leads to more noticeable mismatch in the edges, it disrupts the visual flow and makes the composite look unnatural. Additionally, since the used mask to blend the two images has a hard edge, it will not allow for a gradual transition between the two halves. This makes the boundary between the apple and the orange very obvious and contributes to an unrealistic appearance. Moreover, the shadow of the orange in the left is now revealed and contributes to the problematic appearance.

Understanding unsuccessful blendings involves recognizing that not all images are suitable for blending together in a way that achieves a desirable or coherent result. The learning objective here is to identify what factors contribute to less successful outcomes and how adjusting the blending process (selecting more compatible images, refining masks - not binary, or altering blending techniques) might mitigate these issues.

Pyramids





The Gaussian pyramid is a sequence of downscaled images - the higher the level, the smaller and less detailed the image. While these smaller images may not appear blurry, their size masks the loss of detail. When resized to the original dimensions, the blurriness from lost high-frequency details becomes evident. The Laplacian pyramid, ranging from -255 to 255 (and range adjusted for visualization), predominantly showcases gray levels, as most values hover near zero- indicative of minimal change (adjusted to 128 intensity value. Not including the highest level which its average is about the image average). It is a bit hard to see, since each image only shows details telling about some band frequencies, but in the higher level of the pyramid we see more strong patterns. This is because in the image The lower levels capture fine details like edges (high frequencies), while higher levels reflect broader patterns and textures, aligning with mid-band frequencies. Although hard to see, in the highest level L=G.

Conclusion

In this exercise I have experienced and saw the frequency domain power in image processing. It was evident that not any image pair and masks are suitable for achieving desired results, and that the power is limited to some degree, and require human refinement by selecting the sigma of the low pass filter and the number of pyramids levels (in this exercise I chose to use all possible levels). By experimenting with these techniques, I have learned how the choice of images, the suitableness of masks, and the parameters all play crucial roles in the success of the image processing tasks at hand.