Image Processing - Exercise 3

Liel Amar, liel.amar, 211771993

Introduction

In this assignment, we were to explore image blending by creating two algorithms. The first is used for pyramid blending of images, which is done through the usage of a Mask and Gaussian and Laplacian pyramids. More specifically - we build a Laplacian pyramid for each of our images and a Gaussian pyramid for the mask, blend each level, and then reconstruct the blended pyramid. The second algorithm is for creating gray-scaled, hybrid images through a Gaussian mask. This algorithm creates a Gaussian mask with a specified sigma, computes the FFT of each of the two images, applies the mask and its inverse on each of the first and second images respectively, sums them up, and computes the IFFT to get a result image.

Algorithm

The algorithm I used for the pyramid blending of two images is the following: (1) I import the three given images (two images to blend, and one image for the mask). (2) I then compute the Gaussian pyramid of the mask and compute the Laplacian pyramid of each image. (3) Then I blend the two Laplacian pyramids and the mask into a new Laplacian pyramid, and lastly, (4) I reconstruct the image from the new Laplacian pyramid.

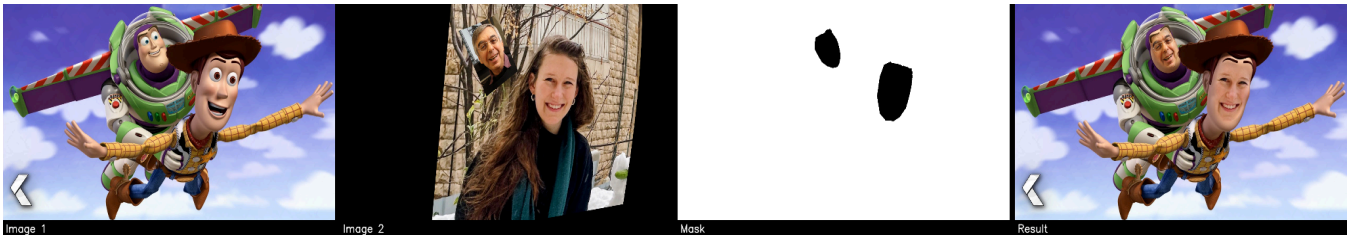
The algorithm I used for creating hybrid images is the following: (1) I import the two images to blend. (2) I then compute the Fourier Transform of each image, and (3) create a Gaussian mask. (4) Then I blend the Fourier transforms with the mask, and lastly (5) compute the Inverse Fourier Transform to get a new, blended, image.

Implementation Details

I implemented both algorithms using *numpy* and *open-cv* as they were most suitable.

To implement the first algorithm, I use *cv2.imread* to read the three images and normalize the mask by dividing it by 255, to act as a binary mask. I then compute the depth of the pyramid (minimum of np.log2 of the smallest axis of the image and 4) and iteratively compute each level of the pyramid with *cv2.pyrDown*. Similarly, I compute the Gaussian pyramid of the two images and then iterate through every two following levels, calculate the difference between the current level and the upscaled (using *cv2.pyrUp* and *np.resize*) next level, and build the Laplacian pyramid. I iterate through the three pyramids (which are the same depth) and for

each level, I blend them by summing the first image multiplied by the mask and the second image multiplied by (1 - mask). Lastly, I iterate through the newly created pyramid, upscale every level, and sum them all up to get the reconstructed image.



The process of Pyramid blending - two images, a binary mask, and the blended image

To implement the second algorithm, I use *cv2.imread* to read the two images to blend, and compute their Fourier Transform using *np.fft.fft2* with *np.fft.fftshif* to shift the transform to the center. Then, I create a Gaussian mask of the same size as the two images, using the Gaussian formula which is $e^{\frac{-((x-x_{mid})^2+(y-y_{mid})^2)}{2*\sigma^2}}$. Afterwards, I add up the Fourier Coefficients of the first image multiplied by the mask, and the Fourier Coefficients of the second image multiplied by (1 - mask) to receive the masked-out Fourier Transform of the two combined images and use *np.fft.ifft2* to compute the Inverse Fourier Transform and get the result.

The process of Hybrid blending - two images, a Gaussian mask, and the blended image

During the implementation of these algorithms, I faced the problem of seeing my progress and tracking down my rights and wrongs. For that, I wrote functions that plot out different stages of the algorithms. Moreover, in the section on pyramid blending, some images had a glow effect around the stitching. After narrowing down the cause, I found out that using too many levels in the pyramid might cause this effect.

The two algorithms differ in the way we blend two images. The first one uses a hyper-parameter (5) to decide the depth of each pyramid, and an arbitrary mask to create a well-blended and smooth result through the usage of Gaussian and Laplacian pyramids, whereas the second one uses a Gaussian mask and a sigma hyper-parameter (40) to create a hybrid image using high frequencies of one image and low frequencies of the other with Fourier Transform.

Results

(a) Successful Pyramid Blending

(b) Successful Hybrid Blending

As can be seen in *Figure A*, the result of our pyramid blending algorithm, which blended two faces onto Woody and Buzz, worked well. The cuts and stitches are relatively smooth. As for the result of our Hybrid blending algorithm, we created a hybrid image of Vin Diesel and Groot, and once again, looking closely at *Figure B* we can see Groot, and from afar, Vin Diesel becomes clearer.

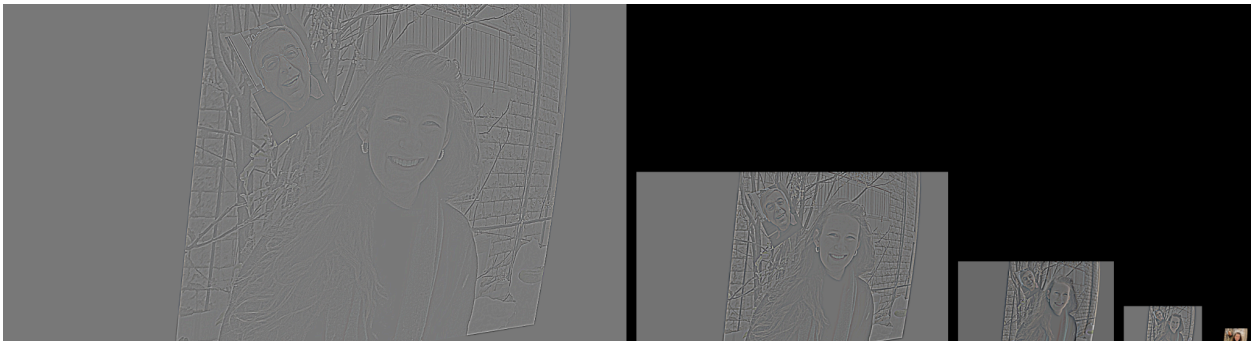
(c) Failed Pyramid Blending

(d) Failed Hybrid Blending

However, we also got some unsuccessful blendings. For example, in *Figure C,* we can see that the pyramid blending failed, and I found out it happens for two reasons. First of all, the mask and the 2nd image have a 100% fit, which makes the feathering process have very little to no effect. In addition, The blending occurs on many levels instead of only 4, which in turn, results in a much brighter image. In addition, in *Figure D*, we see a failed attempt at creating a hybrid image. The process of creating a hybrid image failed due to the sigma variable being too low thus creating a Gaussian mask that isn't descriptive enough to create a proper hybrid image.

Pyramids

Gaussian pyramid of the first image



Laplacian pyramid of the second image

In *Figure E* and *Figure F,* we can see the Gaussian pyramid of the first image, and the Laplacian pyramid of the second image respectively. We can see that with each level, the details in the Gaussian pyramid are lost, and the original image becomes more and more blurry. This sits right with the material learned in class. In contrast, the Laplacian pyramid describes the changes and details that were lost between every pair of consecutive levels in the matching Gaussian pyramid, and we can see that only the fine lines are shown in each level. Each level differs by a bit as it describes different details that were lost. Lastly, the image in the lowest level in the Laplacian pyramid is similar to the lowest level of the matching Gaussian pyramid, as we don't lose any more information.

Conclusion

In this exercise, we explored two methods for blending images: using pyramids and creating hybrid images. The pyramid blend seamlessly merges two images around stitches for a natural look, while the hybrid blend combines the high frequencies of one picture with the low frequencies of another to create a multi-distance effect. Each method requires unique adjustments and parameters but yields impressive results.