

# Assignment 2

Meryem Ülkü Kara, 21727355  
Department of Computer Engineering  
Hacettepe University  
Ankara, Turkey  
b21727355@cs.hacettepe.edu.tr

December 12, 2022

## 1 Introduction

In homework, we tried to image blending. Image blending is the one of most fundamental task ans used for many purposes. The most important thing for image blending methods to blend images seamlessly. In other words for an successful image blending method, seams where images or image regions are stitched must be invisible. We did image blending using pyramids. It was a different and fun experience.

## 2 Experiment

My program take an image as input and a masked image region from another or same image and produce blended image.

First, I created Laplace pyramids for an image. To create a laplace pyramid, we first need to create a gaussian pyramid and perform the laplace pyramid creation algorithm. This, is my gaussian pyramids code.

```
def gaussian_pyramid(img,n):
    gp=[img]      #gaussian pyramid array first value is image
    for i in range(n): #5 time going to resolution. n-1
        layer=cv2.pyrDown(gp[i])
        gp.append(layer)
        #cv2.imshow(str(i),layer)
    return gp
```

Gaussian Pyramids Code

Second, I create Laplacian Pyramids. This, is my laplacian pyramids code.

```

def laplacian_pyramid(gp,n):
    lp=[gp[n]]
    for i in range(n,0,-1):
        size_img1=(gp[i-1].shape[1],gp[i-1].shape[0])
        gaussian_extended= cv2.pyrUp(gp[i], dstsize=size_img1) #upper level of gaussian pyramid
        laplacian=cv2.subtract(gp[i-1], gaussian_extended)
        # cv2.imshow(str(i),laplacian)
        lp.append(laplacian)
    return lp

```

Laplacian Pyramids Code

Afterwards I created a mask. I then created a Gaussian pyramid for each region mask. This is my mask image and mask code.

```

mask=np.zeros((500,900,3), dtype='float64')
mask[125:350,320:800,:]=(1,1,1)

```

Mask Code



Mask Image

After creating the mask, I apply the mask for the two photos and combine the photos.

```

# now reconstruct
def reconstruct_image(blended_image,n):
    reconstruction= blended_image[0]
    reconstruction_list=[reconstruction]
    for i in range(n-1):#lr=cv2.pyrDown(img) #lr is low resolution
        size = (blended_image[i+1].shape[1], blended_image[i+1].shape[0])#lr2=cv2.pyrDown(lr) #lr is low resolution2
        reconstruction_expanded = cv2.pyrUp(reconstruction, dstsize = size) #up to sixth level
        reconstruction = cv2.add(blended_image[i+1],reconstruction_expanded)#hr=cv2.pyrUp(lr2) #hr is higher resolution
        reconstruction_list.append(reconstruction)
    return reconstruction_list

```

Blending Code For Images

Here is an example for my code output.



Image 1

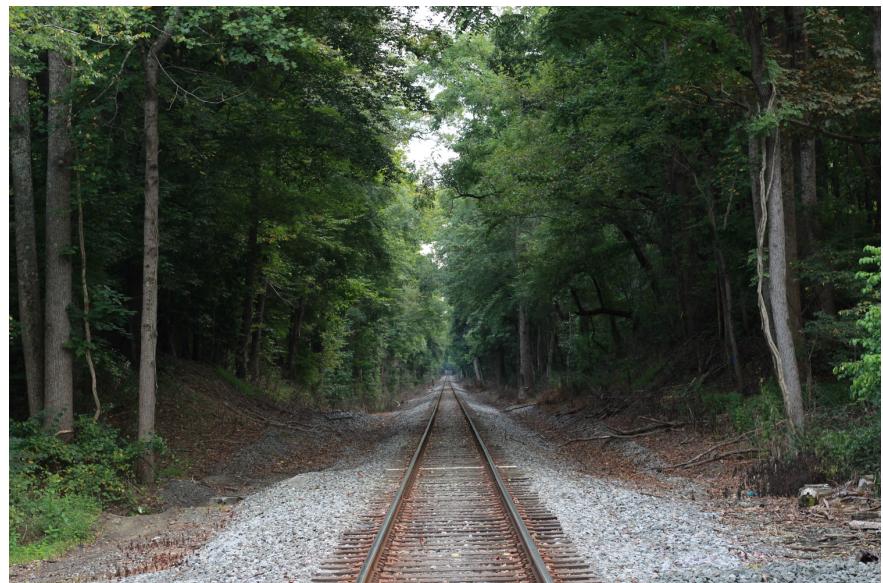


Image 2



Image After Blendig

### 3 Conclusion

The most interesting part for me was applying a mask and being able to combine neatly without any lines. I learned this and I think it has given me a lot.

### References