Hybrid images and blending using Gaussian and Laplacian pyramids

Saul Gomez
Universidad de Los Andes
sc.gomez11@uniandes.edu.co

Diego Valderrama
Universidad de Los Andes
df.valderrama@uniandes.edu.co

Abstract

A hybrid image is the sum of low and high frequencies form filtered versions of two images that creates a visual illusion that changes with viewing distances. The original images were resized, cropped and aligned as part of preprocessing stage. Gaussian filter was used to extract low and high frequencies and the paremeters were varied in order to get a better response. Finally, Gaussian and Laplacian pyramid representations were implemented to visualize hybrid and blended images.

1. Introduction

Hybrid images are illustrations that create visual illusions by providing two different interpretations to the observer depending on the viewing distance or the presentation time. This particular characteristic is achieve by superimposing the high frequencies of one image over the low frequencies of another one, so that when the observer is closer the high frequencies can be observed, while at the distance the low frequencies are appreciated. One of the most useful techniques, was the one stated by Oliva and Torralba, where low frequencies are extracted by filtering the image with a low-pass filter and high frequencies with a high-pass filter to finally add the two resulting images [1].

2. Methods

The original images were taken by the author in order to focus people's faces and avoid as much as possible preprocessing work (See figure 1). The figure 1(a) depict a picture of one of the authors in a wooden background. He has brown hair and beard and is wearing a gray shirt. The head is rotate a few degrees to left, so left ear is almost hidden. The cheeks are large, the mouth and eyes small. On the other hand, figure 1(b) shows the face of one author's classmate in a brick background. This was especially chosen because they have similar color intensities. He also have brown hair and beard, he is wearing a white t-shirt and dark gray jacket. He has a larger mouth with fine lips and one of the eyes is a little closed.

2.1. Image alignment

Despite the attempt to capture aligned images, it was necessary to process both images. Daniel's picture was resized and cropped in order to get similar head measures. Additionally, the images were aligned following the eyes and nose. The resulting images are shown in figure 2

2.2. Hybrid images

First, Daniel's image was used to extract low frequencies while high frequencies were extracted form Saul's image. For Daniel's image a $55x55$ kernel was implemented to filter the image with a Gaussian filter and a standard deviation $\sigma = 9$. In contrast, Saul's image was filtered with a $55x55$ window and a Gaussian blur with standard deviation $\sigma = 3$ and then subtracted from the original image. In order to preserve contours and avoid chromatic aberration, Saul's image was converted to gray scale. Finally, the resulting filtered images (Saul's image was concatenated three times to get the corresponding color channels) were sum to generate the hybrid image.

2.3. Blending

First, the images were loaded, resized to 512,512 and saved. After, Gaussian and Laplacian Pyramid for the images were created. In the next step, the left part of the highest level of Gaussian Pyramid of the first image was joined with the highest level of Gaussian Pyramid of the second image (base image). This method was also performed for all levels of Laplacian Pyramid of both images. Subsequently, the base image was reconstructed with the image obtained by joining the fourth level of the Laplacian pyramids. This procedure was iterated until the reconstructed image had the same size as the original image. For this purpose, the images obtained by joining the levels of the Laplacian pyramids were used. This algorithm was based in different methods found in internet [2–4]

3. Results

The hybrid image and the Gaussian pyramid representation of the hybrid image are shown in figures 3 and 4, respectively. Likewise, the blended image is illustrated in figure 5 while its Gaussian pyramid is shown in figure 6.

4. Conclusion

Based on the technique stated by Oliva and Torralba it was possible to create an hybrid image. The selection of the parameters that improves the response of the Gaussian filter, such as kernel's size and standard deviation, is arbitrary and depends on the features of the original images. The chosen parameters were established heuristically to obtain an optimal hybrid image.

Previous image processing as alignment and filtering images might a good idea in order to enhance the details. Furthermore, pyramid method allows join two images and blending them. However, the number of levels used in the pyramid is an arbitrary selection and depends on the images.

Based on the final blending image we are able to conclude that the blended image with four levels of Laplacian pyramid is good for the images selected. Nevertheless, might a good idea select to images with higher similarity in order to obtain better visual results.

References

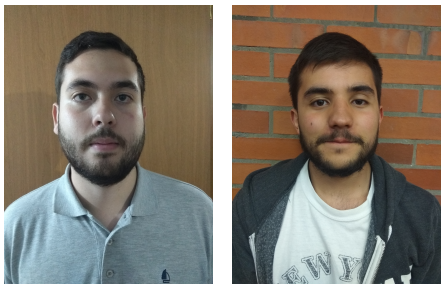
[1] A. Oliva, A. Torralba, and P. Schyns. Hybrid images. *ACM Transactions on Graphics*, 25(3):527.

[2] OpenCV. image pyramids. https://docs.opencv.org/3.1.0/dc/dff/tutorial$_p y_p yramids.html$.

[3] scikit image. image pyramids. http://scikit-image.org/docs/dev/auto$_e xamples/transform/plot_p yramid.html$.

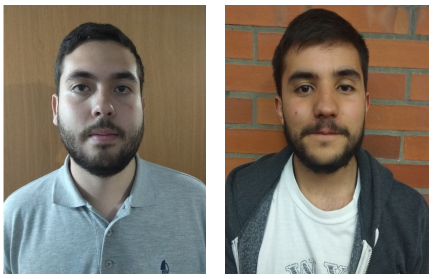
[4] SciPy.org. numpy.hstack. https://docs.scipy.org/doc/numpy/reference/generated/numpy.hstack.html.

5. Images and code snippets



(a) Saul's image (b) Daniel's image

Figure 1. Original images



(a) Saul's modified image (b) Daniel's modified image

Figure 2. Alignment images

```
# Low-pass filter
img_dan= cv2.imread(p_dan)
img_dan=cv2.cvtColor(img_dan,cv2.COLOR_BGR2RGB)
f_dan=cv2.GaussianBlur(img_dan,(55,55),9)
# Conversion to grayscale and High-pass filter
img_cam= cv2.imread(p_cam)
gs_cam=cv2.cvtColor(img_cam,cv2.COLOR_BGR2GRAY)
g_cam=cv2.GaussianBlur(gs_cam,(55,55),3)
f_cam=gs_cam-g_cam
f_cam=np.stack((f_cam,f_cam,f_cam),axis=-1)
# Sum of low and high frequencies and hybrid image
fig = plt.figure()
plt.imshow(f_dan+f_cam)
plt.show()
```

Figure 3. Hybrid image

Figure 4. Gaussian pyramid of hybrid image

```
#Generate gaussian and laplacian pyramid
#for the two images.
GP_Img_1= [Img_1]
GP_Img_2 = [Img_2]
LP_Img_1= []
LP_Img_2 = []

for i in range(1,6):
    GP = cv2.pyrDown(GP_Img_1[i-1])
    GP_Img_1.append(GP)
    GP = cv2.pyrDown(GP_Img_2[i-1])
    GP_Img_2.append(GP)
    LP = cv2.pyrUp(GP_Img_1[i])
    L = cv2.subtract(GP_Img_1[i-1],LP)
    LP_Img_1.append(L)
    LP = cv2.pyrUp(GP_Img_2[i])
    L = cv2.subtract(GP_Img_2[i-1],LP)
    LP_Img_2.append(L)

LP_Img_1.append(GP_Img_1[5])
LP_Img_2.append(GP_Img_2[5])

#Join left part of Img_1 and rigth part of
#Img_2 in a new image for each level.This
#method also do the reconstruction using
#laplacian pyramid

Blending= []
f=0
for i in range(5,-1,-1):
    img1=LP_Img_1[i]
    img2=LP_Img_2[i]
    _,cols,_= img2.shape
    B= np.hstack((img1[:,0:int(cols/2),:], ...
        img2[:,int(cols/2):,:]))
    Blending.append(B)
    if i == 4:
        Blending_UP = cv2.pyrUp(Blending[f])
        Blending_UP = cv2.add(Blending_UP, ...
            Blending[f+1])
        f=f+1
    elif i<4:
        Blending_UP = cv2.pyrUp(Blending_UP)
        Blending_UP = cv2.add(Blending_UP,...
            Blending[f+1])
        f=f+1
```

Figure 5. Blending image using pyramids

Figure 6. Gaussian pyramid of blending image