Hybrid Images

Federico Ulloa
Universidad De Los Andes
Cra. 1 18a-12, Bogotá, Colombia
f.ulloa10@uniandes.edu.co

Esteban Vargas
Universidad De Los Andes
Cra. 1 18a-12, Bogotá, Colombia
e.vargas11@uniandes.edu.co

Abstract

Images are said to have different frequencies depending on how abruptly the intensities change within a local region. It is possible to delimit the frequencies an image ranges on by using high pass and low pass filters. In this laboratory, we took 2 images, aligned them, took the high frequencies of one and the low frequencies of the other to merge them into a hybrid image. Hybrid images are a particular kind, as human perception makes us see one of the images from up close and the other image from a higher distance. Lastly, Gaussian and Laplacian pyramids were used to create a blended image from 2 halves from the same original images.

1.. Introduction

Images vary in many different aspects between each other. One of these characteristics is the range of frequencies of each image. In an image, a frequency can be interpreted as how rapidly the intensity of pixels change within its vicinity. For instance, a very smooth image with very little changes in intensities is said to have very low frequencies, unlike a more complex image that contains different objects and intensities in a tiny space, high frequencies are predominant.

In image processing, it is possible to delimit the frequencies contained in an image by filtering the image with a low pass and/or high pass filter. One of the most commonly used low pass filter is the Gaussian filter, which takes the size of the kernel and the standard deviation as parameters. These parameters determine how much the image will be smoothed.

Hybrid images are images in which, from a close distance, the viewer sees one thing, and from a higher distance the viewed object changes. These hybrid images are created by taking two fairly aligned images and adding the high frequencies from one and the low frequencies from the other. The effect works because human vision is more receptive to high frequencies from a close distance and they fade out when the distance increases, as vision becomes more receptive to low frequencies. [1]

2.. Methodology

2.1.. Hybrid and blended images creation

First, two photographs were taken from 2 people, trying them to be as close in size, orientation and distance as possible. These original photographs can be seen in figures 1 and 2. From these photographs, a photo editor was used to align both images. One of the images had to be cut and re sized for them to be aligned in the best way possible. The cropped photographs that were aligned are shown in figures 3 and 4.

To create the hybrid image, Gaussian filters were used, which are low pass filters. First, one of the images was filtered with the Gaussian filter in order to eliminate the higher frequencies from the image. Then, for the second image, the same procedure was made but the resulting image was the filtered one subtracted from the original, resulting in an image with only the higher frequencies. Lastly, the two resulting images were added.

For the blended image, one half of each image were concatenated to create a single image. Then, Gaussian pyramid was used, adding the corresponding Laplacian pyramid 5 times to end up with the blended image, which has a smoother transition between the 2 halves [2]. Listing 1 shows the code for the creation of these images.

3.. Results

3.1.. Hybrid and blended images creation

After applying the Gaussian filers and adding the low and high frequencies from the images 1 and 2, respectively, the hybrid image was created. The image shown in figure 5 is the resulting hybrid image. As it can be seen, the effect is pretty good but it could be improved with photographs that can be aligned even better.

In figure 6, the resulting blended image can be seen. Once again, the transition is much smoother with this procedure, but a better effect could have been achieved with photo-

graphs that could be aligned better.

4.. Conclusions

Hybrid images are the result of adding the high frequencies from one image and the low frequencies from another. When the two images are aligned well, the effect works even better. This happens because human vision perceives better the high frequencies from a close distance and rely on the low frequencies primarily when the distance is greater.

On the other hand, Gaussian and Laplacian pyramids allow us to blend two halves of different images in a very smooth way. Once again, this works better with images that are well aligned.

Referencias

[1] "Hybrid images", Cvcl.mit.edu, 2008. [Online]. Available: http://cvcl.mit.edu/hybrid/OlivaTorralb$_{H}ybrid_{S}iggraph06.pdf.[Accessed: 20-Feb-2018]$.

[2] .^{o}penCV: Image Pyramids", Docs.opencv.org. [Online]. Available: https://docs.opencv.org/3.1.0/dc/dff/tutorial$_{p}y_{p}yramids.html.[Accessed: 20-Feb-2018]$.

5.. Code

Listing 1: Hybrid and blended images creation

```
im1=ndimage.imread("Imagenes/Mario.jpeg
    ")
im2=ndimage.imread("Imagenes/Oscar.JPG
    ")

im1=sc.misc.imresize(im1,(800,704),
    interp="cubic")
im2=sc.misc.imresize(im2,(800,704),
    interp="cubic")

im1_bajo=ndimage.gaussian_filter(im1,5)
im2_bajo=ndimage.gaussian_filter(im2
    ,15)

im2_alto=cv2.subtract(im2,im2_bajo)
im=cv2.add(im2_alto,im1_bajo)
plt.imshow(im2_alto)
plt.imshow(im)
plt.imsave("Hybrid.png",im)

Co = im1.copy()
pIm1 = [Co]
for i in range(6):
    Co = cv2.pyrDown(Co)
    pIm1.append(Co)
    name="PG1_Layer"+str(i)+".png"
    plt.imsave(name,Co)

plIm1 = [pIm1[5]]
for i in range(5,0,-1):
    GE = cv2.pyrUp(pIm1[i])
    L = cv2.subtract(pIm1[i-1],GE)
    plIm1.append(L)

Co = im2.copy()
pIm2 = [Co]
for i in range(6):
    Co = cv2.pyrDown(Co)
    pIm2.append(Co)
    name="PG2_Layer"+str(i)+".png"
    plt.imsave(name,Co)

plIm2 = [pIm2[5]]
for i in range(5,0,-1):
    GE = cv2.pyrUp(pIm2[i])
    L = cv2.subtract(pIm2[i-1],GE)
    plIm2.append(L)

LS = []
for la,lb in zip(plIm1,plIm2):
    rows,cols,dpt = la.shape
    mit=int(cols/2)
    ls = np.hstack((la[:,0:mit], lb[:,
        mit:]))
    LS.append(ls)

ls_ = LS[0]
for i in range(1,6):
    ls_ = cv2.pyrUp(ls_)
    ls_ = cv2.add(ls_, LS[i])

plt.imsave("Blending.png",ls_)
```

6.. Images

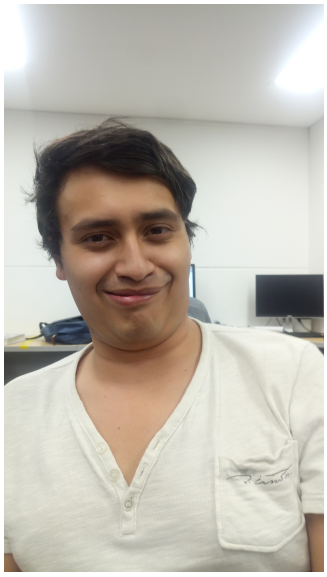

Figura 1: Original image 1

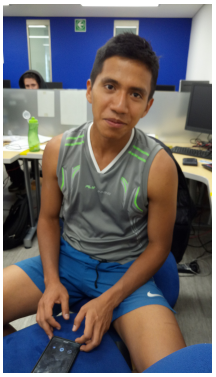

Figura 2: Original image 2

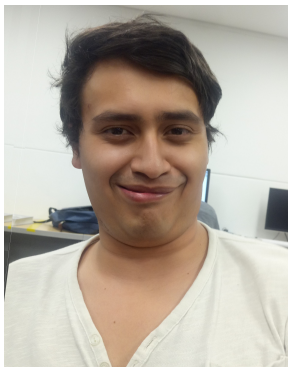

Figura 3: Cropped image 1

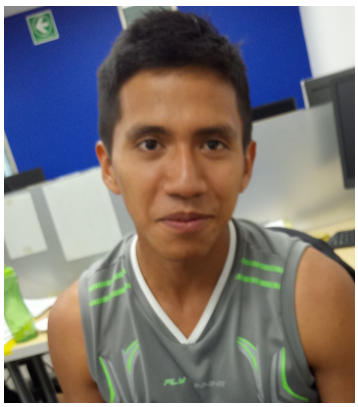

Figura 4: Cropped image 2

Figura 5: Hybrid image

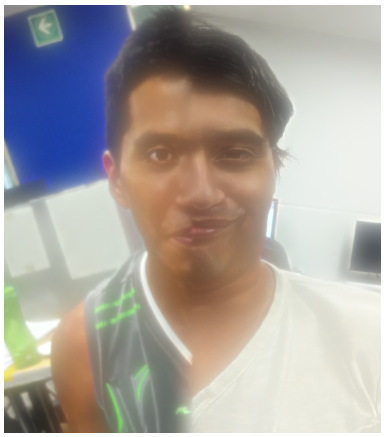

Figura 6: Blended image