Lab 4: Hybrid Images

Mariajosé Serna Ayala
Universidad de los Andes
m.serna10@uniandes.edu.co

Henry Daniel Torres
Universidad de los Andes
hd.torres11@uniandes.edu.co

Abstract

In this report we demonstrate the creation of hybrid images, which are images that can have two different interpretations depending of the distance between the image and the observer. We implement a function in MATLAB which filters two entry images, one with a high pass filter and the other one with a low pass filter. For this, we created Gaussians filters and rectangular filters, with the intention of comparing the result of each filter. At the end, it was possible to create hybrid images with these two types of filters and with different values of σ for the Gaussian filters.

1.. Introduction

Hybrid images are those who change their interpretation as a function of viewing distance. It is composed of two original images, in which we take the high frequencies of one of the images and the low frequencies of the other image. Ideally an hybrid image can be done with two any images, but in practice the final effect works better if the images follow a certain rules described in the paper "Hybrid Images" from Olvia, Torralba and Schyns. Some of the rules they mention where: The images match better if they are align. And, symmetry and repetitiveness of a pattern in the low spatial frequencies are not good because they form a strong percept that it is difficult to eliminate perceptually[1].

Therefore hybrid images are created by two filtered images, one filtered by a low pass filter and the other one passed by a high pass filter. This allows to have the high frequencies of one image and the low frequencies of the other one. This can be described by the equation:

$$H = L(X_1, T_1) + (X_2 - L(X_2, t_2)) \tag{1}$$

Where L is a low pass filter, X_1 and X_2 are the two images, and T_1 and T_2 are the cut off frequencies. In this case we used two type of filters: rectangular and Gaussians. For Gaussian filters, the cut-off frequency depends on the value of standard deviation (σ)[2].

2.. Methods and Materials

The hybrid images where obtain by implementing a function in MATLAB which filters and then merges two input images. For this, it was necessary to use the function *fft2()* which aplies the Fast Fourier Transform in the 2 dimensions of the images. It is observable in the *snipet 1* that a further transformation is needed to correctly manipulate the images in the frequency domain, this is done by aplying the function *fftshift()* that shifts the zero-frequency component to the middle of the spectrum.

Sniplet 1.

```
imfrec1 = fftshift(fft2(double(
    image1)));
imfrec2 = fftshift(fft2(double(
    image2)));
```

For filtering the input images in the frecuency domain it is necessary to multiply them by a mask that filters the high or low frenquency components. There were constructed two kind of masks: rectangular and gaussian. On one hand, as seen in the *snipet 2*, the rectangular mask should be the same as the original image and it functions by creating a rectangular black area in the center of mask while the rest is white. A mask of this charactheristic is able to filter the low frequencies and output only the frequencies that are higher than the threshold (out of the rectangle). Applying a complementary mask to the second image drove to obtain and image with only low frequencies left. On the other hand, the gaussian filter (with a standard desirable deviation of 10) acts based on the same rule but, instead of a rough rectangular edge, it uses a smooth gaussian-like transition

All of the aforementioned is valid for 2d images. In the case of colour images it's only needed to apply the mask on each colour dimension. Finally, for a correct visualization of the resulting image the functions *ifftshift()* and *ifft2()* for the Inverse Fourier Transform.

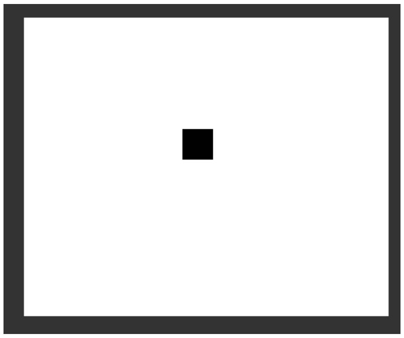


Figure 1: High pass rectangular filter.

Sniplet 2.

```
maskh = ones(size(image1, 1), size(
    image1, 2));
maskh(round(6*size(image1, 1)/15):
    round(8*size(image1, 1)/15),
    round(6*size(image1, 2)/15):round
    (8*size(image1, 2)/15)) = 0;
maskl = imcomplement(maskh);
```

Sniplet 3.

```
maskl = fspecial('gaussian', [m n],
    sigm);
maskl = maskl./max(max(maskl));
maskh = 1 - maskl;
```

Sniplet 4.

```
D(:, :, rgb) = imfrec1(:, :, rgb).*
    maskh + imfrec2(:, :, rgb).*maskl
```

Sniplet 5.

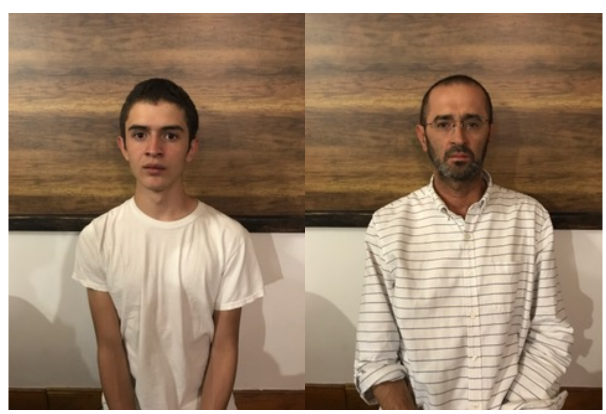
```
 D = imfrec1.*maskh + imfrec2.*maskl
     ;
```

There was possible to generate hybrid images from photos that already existed, with only one couple of photos (Figure5a and Figure5b) because these where the only ones framed in a similar way. Therefore we decided to take our own images based on the criteria mentioned in the introduction. The images used to generate the final hybrid images where:

(a) (b)

Figure 2: First pair of images.Where (a) correspond to Mariajosé and (b) corresponds to her mother.

This first couple of images, figure2a,and figure2b, correspond to Mariajosé and her mother. These, as the couple of next images where taken with the same background and with the camera positioned in the same place, so there would not be problem with the alignment. Pictures from figure3a, and figure3b corresponds to Mariajose's brother and father respectively. Non of these four photos needed to be modify when creating the hybrid images.



(a) (b)

Figure 3: Second pair of images used for the creation of an hybrid image.(a) Corresponds to Mariajose's brother and (b) Corresponds to her father.

(a) (b)

Figure 4: Third pair of images used for the creation of an hybrid image. (a) Corresponds to brother (b) Corresponds to father.

The third couple of images used, figure4a and figure4b, are close ups to the eyes of the same persons of the second pair of images. These were harder to take because it was more complicated to leave the camera in the same please, therefore they where not too align. And finally, the last two images where the only ones that where already taken. These two photos coincidentally had a similar alignment. The only problem was that those where not the same size, so we used the function $imresize()$ from MATLAB to leave both images of the same sizes.

(a) (b)

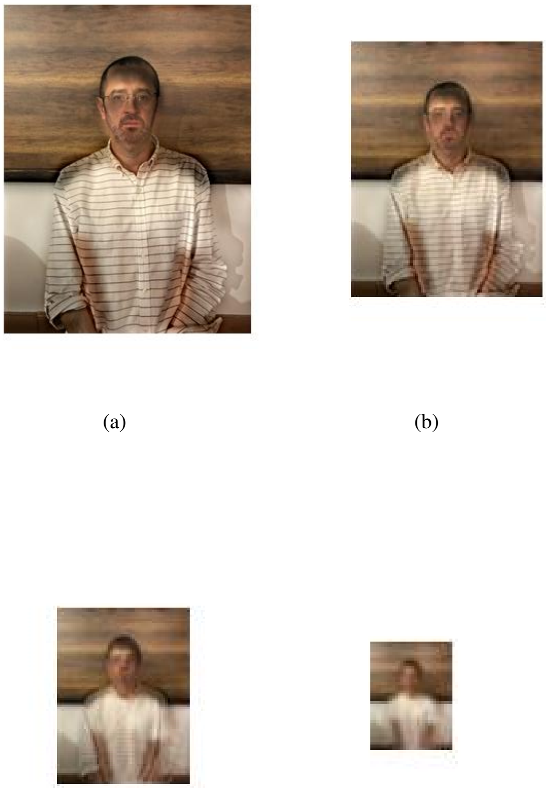
Figure 5: Fourth pair of images used for the creation of an hybrid image.(a) Jaime, Mariajose's couple (b) Corresponds to Mariajose.

3.. Results

The results shown consist in only images filtered with Gaussian filters, because this was the filter that gave better results. Also, all of them have a σ value between 9 and 11.

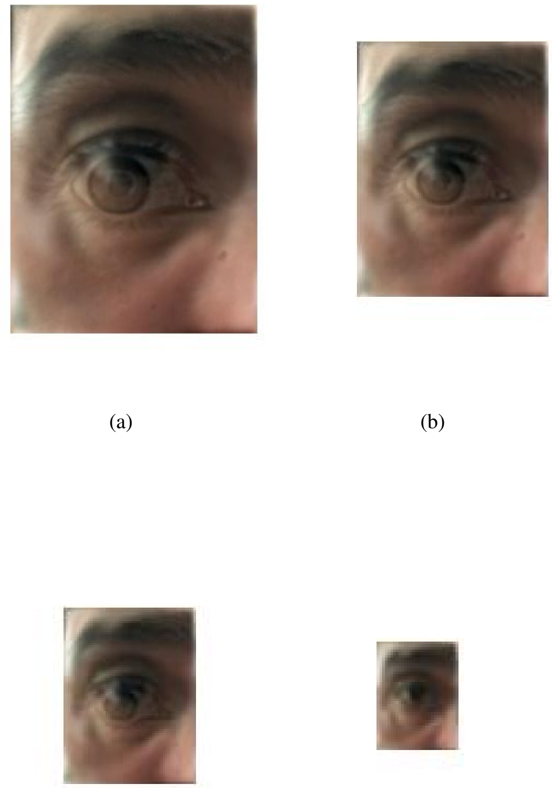
From each hybrid image the picture that can be seen in the bigger scale, and that looks a little transparent, corresponds to the image that passed through the high-pass filter. And the one seen in the background, and that can be distinguish in the smaller scales, correspond to the image that passed though the low-pass filter. This is also intuitive because: We can perceive details (high frequencies) in images filtered with a high-pass filter, and we can differentiate a general form of the images filtered with low -pass filter (low frequencies).

Figure 6: Hybrid image resulting from the first pair of images (Figure 2.) and a 4-scale Gaussian pyramid.



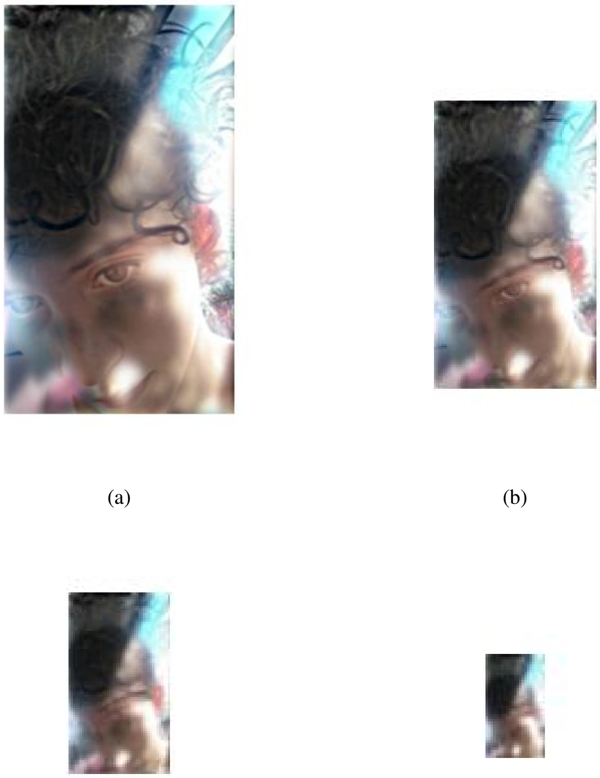
(a) (b) (c) (d)

Figure 7: Hybrid image resulting from the second pair of images (Figure 3.) and a 4-scale Gaussian pyramid



(a) (b) (c) (d)

Figure 8: Hybrid image resulting from the third pair of images (Figure 4.) and a 4-scale Gaussian pyramid



(a) (b)

(c) (d)

Figure 9: Hybrid image resulting from the fourth pair of images (Figure 5.) and a 4-scale Gaussian pyramid

4.. Conclusions

- Proving different values of σ for the Gaussian filters we find that at lower values of σ the image that is supposed to be passed through the high pass filter is the one that predominates in all the scales of the Gaussian pyramid.
- Rectangular filter does not preserved really well details in the images passed through the high-pass filter. Which lead as to say that Gaussian filters gives better results for this application.

References

[1] Oliva, A., Torralba, A., AND Schyns, P. Hybrid Images. SIGGRAPH 2006.

[2] Wikipedia. Gaussian Filters. Retrieved from: `https://en.wikipedia.org/wiki/Gaussian_filter`