Hybrid Images

Jorge Madrid
Universidad de los Andes
Department of Biomedical Engineering
ja.madrid2714@uniandes.edu.co

Abstract

Images contain information of various spatial frequencies; high-frequency information corresponds to details while low-frequencies correspond to large-scale features. Hybrid images contain the high-frequency information of one photograph and the low-frequency information of another one. This produces a visual effect that depends on the scale of the image, allowing the observer to identify one image or the other. In this document we present the results of the exercise of producing our own hybrid image from pictures of our authorship. We present our original images, the pre-processing applied on them, the final result and the code that allows for it.

1. Introduction

Hybrid images serve to introduce ourselves to the concept of spatial frequency in images. Hybrid images are produced by combining two filtered images, one in which the high frequencies have been removed and another one in which only such frequencies have been preserved. To produce a good hybrid image, the original images must have similar features occurring at roughly the same place.

We present the result of hybridizing two images.

2. Materials and Methods

Original Images and Pre-processing

Image 1 is a portrait of my friend Sara, while image 2 is a portrait of myself. These two images were selected based on their overall similar structure: They had features in the same places in the photograph. Coinciding spatial features is the main selection criterion for producing hybrid images.

These two photographs were cropped to make features like the eyes, the mouth, and the edges of the head coincide. This is fundamental to achieve a good hybrid image. After cropping, images were also rescaled so that their dimensions matched. This last point was executed in Matlab, while the former was executed using Microsoft Paint.

Figure 1: Portrait of Sara.

Figure 2: Portrait of myself (some years ago).

Use of color

Although both images were portraits, their backgrounds were remarkably different color-wise. Other features, such as hair color, were also different. Thus it was more convenient to use the gray-scale version of the images.

3. Results

Figure 5 was obtained by hybridizing figures 1 and 2 by means of filtering and subtraction and then by rescaling the

Figure 3: Images after pre-processing. Original images were cropped, rescaled, and converted to grayscale to achieve this.

resulting picture. All processing was conducted in the spatial domain. After initial image processing (i.e. cropping and displacing), images were filtered with a Gaussian filter. We used filtering to keep high-frequency information from Sara's portrait and low-frequency information from my portrait. Although filtering can performed in the reciprocal domain, we choose not to perform it that way to avoid having to deal with aliasing [1].

Trials for different filter size and sigma had to be made by hand to optimize the resulting hybridization of the images. This is not included in our code, but is simple to implement by producing hybrid images for multiple sigmas and sizes simultaneously.

Code

The first section of the code deals with image preprocessing, that is image rescaling and color channel selection.

```
% Initial processing
[x y z] = size(imH);
imL = imresize(imL,[x y]);  % ...
    Low-frequency image is resized to ...
    meet the size of the high-frequency ...
    image. This is fundamental to perform ...
    operations on the two images.

imL = rgb2gray(imL);        % Convert ...
    both images to grayscale. Only to be ...
    done if colors are problematic for ...
    the matching.
imH = rgb2gray(imH);
```

Figure 4: Resulting hybridized image. The size and the sigma for the Gaussian filter were 200.

Then we have the section involving filtering. As mentioned earlier, filtering is carried out in the spatial domain by means of a Gaussian filter.

```
% Filtering
filter = fspecial('gaussian', 100, 100); ...
    % A Gaussian filter is defined
imH_100 = abs(imsubtract(imH, ...
    imfilter(imH, filter))); % Only ...
    high-frequency information is kept
imL_100 = imfilter(imL, filter); % ...
    High-frequency information is ...
    filtered away
imHybrid_100 = imadd(imL_100, imH_100); % ...
    The two images are hybridized.
```

Finally, to produce the image pyramid, we must produce various rescaled copies of the original image. To do, we simply implement the following several times for different scales:

```
% Pyramid rescaling and image saving
imHybrid_2 = imresize(imHybrid_100, 0.80);
imwrite(imHybrid_2, 'pyramid2.png')
```

4. Conclusions

Hybrid images favor the understanding of frequency content in images. Although they are simple in principle (the combination of the high-frequency content of one image and the low-frequency content of another), producing them is more of an art. This is so because relevant spatial

Figure 5: Hybridized image pyramid

features (such as the eyes our the mouth in our example) must coincide in the two input images. This is achieved with pre-processing such as cropping and color selection. A good hybrid image also depends on the characteristics of the applied filter. For this reason we had to try out several combinations of filter size and sigma. Th optimal values for these two parameters were 200.

References

[1] Steve Eddins. Aliasing and the discrete fourier-transform. =http://blogs.mathworks.com/steve/2010/02/22/aliasing-and-the-discrete-time-fourier-transform/. Last accessed: 2017-02-23. 2