Image filtering and Hybrid Imaging

Akansha Singh
2017csb1065@iitrpr.ac.in
Indian Institute of Technology Ropar
Rupnagar, India

1 Introduction

This paper presents to you the use of image filtering to generate hybrid images following the baselines of the SIGGRAPH 2006 paper by Olivia, Torralba, and Schyns. The so generated Hybrid images are static images that changes in interpretation as a function of the viewing distance. At a distance low frequency distance dominates while closely enough we are able to see higher frequencies. Through a self made method similar in application to that of matlab function imfilter() or python function scipy.ndimage.correlate() . Once a filtered image is formed of high and low fequencies, we visualise the hybrid image by progressively downsampling the images and concatenating them together.

2 Image Filtering

Creating the my-imfilter function involves 2 tasks. The first step is to create a padded matrix such that it is able to hold the so formed correlated matrix and fix the given image to its centre. The second part involves the correlation operation, done so as to set the value of the sum to each pixel of the final matrix.

(a) (b) (c)

Figure 1: (a) is the identity image of a cat on which blur filter is applied to obtain (b) and on applying sobel filter on identity image we obtain (c).

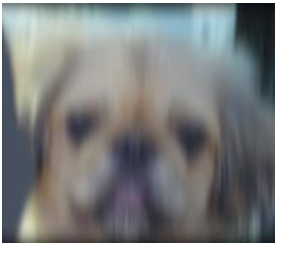
```
# my_imfilter(image, filter)
    output = np.zeros_like(image)
    image_padded = np.zeros((img_len+pad_len-1,
        img_width+pad_width-1, image.shape[2]))
    image_padded[int((pad_len-1)/2):img_len+int((pad_len-1)/2),
        int((pad_width-1)/2):img_width+int((pad_width-1)/2)] = image
```

```
for k in range(len(image[0][0])):
   for x in range(image.shape[1]):
      for y in range(image.shape[0]):
         output[y,x,k] = np.multiply(filter,
             image_padded[y:y+filter.shape[0],
             x:x+filter.shape[1],k]).sum()
output = np.clip(output, 0, 1)

#return output
```

3 Results

We obtain the following results from the code



(a: low)

(b: high)

(c: hybrid)

(d: scales)

Figure 2: Different images formed by running proj1.py on different set of results

We can depict from the above cases that on performing operations like extracting high frequency or low frequency images using filters. These filters mainly compute the correlation/convolution wrt. to each pixel of test-image and store it as a new image. For generating hybrid images we use similar images and add high frequency of one with low frequency of another. We can observe that initially high-frequencies catch our eyes but as we the size of the image gets smaller we can view low frequencies only.

(a: submarine-fish)

(b: marilyn-einstein)

(c: motorcycle-bicycle)

(d: plane-bird)

Figure 3: Different images foremd by running proj1.py on different set of images