COMP6223 Computer Vision

Image Filtering and Hybrid Images

Ganiyu Ajibola Ibraheem
gai1u17@soton.ac.uk
29447267

November 15, 2017

Contents

List of Figures

List of source codes

1 Convolution and Hybrid Images Algorithm

Figure 1: Hybrid Image of a fish and a submarine

The convolution algorithm can be used to generate an hybrid image as shown in figure 1 and it works for any arbitrary image and a given odd numbered kernel size.

The hybrid image algorithm works in 3 parts which are reading in the images, finding the high frequencies and low frequencies of the images and lastly combining them to form an hybrid image. The algorithm was implemented using Matlab and its implementation are further discussed in the following subsections.

1.1 Reading in the Images

Matlab provides functions for reading in images which is the *imread* function. The

```
img1 = imread('data/fish.bmp','bmp');
img2 = imread('data/submarine.bmp','bmp');

img_1_R = double(img1(:,:,1));
img_1_G = double(img1(:,:,2));
img_1_B = double(img1(:,:,3));
```

Listing 1: Reading in an image and separation of channels

images are read and converted into doubles as they are stored as integers and need to be converted to work in Matlab.

1.2 Determining the Low and High Frequencies of the Image

The Low frequency of the image is determined by generating a Gaussian kernel of a specified size. For a given kernel size e.g 7x7, with variance (σ^2) the resulting Gaussian Kernel is calculated by

$$gaussian(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{x^2 + y^2}{2\sigma^2} \tag{1}$$

The convolution of the Gaussian kernel and the image produce a low frequency image as shown in figure 2a. The high frequency of the image is obtained by subtracting the low frequency from the original image and an example is shown in figure 2b. The image convolution code is further show in 2 where the image is

(a) Low Frequency Image (b) High Frequency Image

Figure 2: High and Low Frequency Image

convolved with the template and normalised.

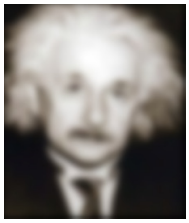
2 Results

The hybrid image pipeline is as follows for the convolution of an image of Albert Einstein and Marilyn Monroe.

- The low frequency of Einstein is obtained as shown in figure 3
- The high frequency of Marilyn Monroe is obtained as shown in figure 4

```
function convolved = convolve(image,template)

[irows,icols]=size(image);
[trows,tcols]=size(template);
temp(1:irows,1:icols)=0;
trhalf=floor(trows/2);
tchalf=floor(tcols/2);

for x = trhalf+1:icols-trhalf
  for y = tchalf+1:irows-tchalf
    sum=0;
    for iwin = 1:trows
      for jwin = 1:tcols

↪   sum=sum+image(y+jwin-tchalf-1,x+iwin-trhalf-1)*template(jwin,iwin);
        temp(y,x)= sum;
      end
    end
  end
end

minim=min(min(temp));
range=max(max(temp))-minim;
convolved = floor( (temp - minim) * 255/range );
```

Listing 2: Image Convolution and Normalisation



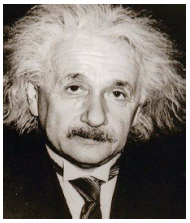


Figure 3: Low and Normal picture of Einstein

- And the Hybrid Image is shown in figure 5

2.1 Images at Different Scale

The effects of viewing the images at different scales are further shown below in figure 6.

Figure 4: High and Normal picture of Marilyn Monroe

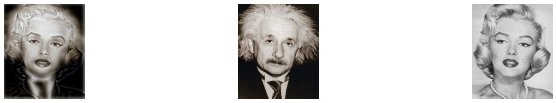


Figure 5: Hybrid Image of Einstein and Marilyn

(a)

(b)

(c)

Figure 6: Viewing images at different scales