Homework X Writeup

Instructions

- Describe any interesting decisions you made to write your algorithm.
- Show and discuss the results of your algorithm.
- Feel free to include code snippets, images, and equations.
- Use as many pages as you need, but err on the short side If you feel you only need to write a short amount to meet the brief, th
- **Please make this document anonymous.**

In the beginning...

It was a challenge to feel that computer vision is really fun.

Interesting Implementation Detail

My code snippet highlights an interesting point.

```
    [f_m, f_n] = size(filter);
    assert(mod(f_m, 2) ~= 0 && mod(f_n, 2) ~= 0, 'even
       size filter error');
    mid_f_m = (f_m + 1)/2;
    mid_f_n = (f_n + 1)/2;
    [m,n,channel] = size(image);
    output = zeros(m,n,channel);
    % padding
    pad_img = padarray(image, [mid_f_m - 1 mid_f_n - 1]);
    [m2,n2,c2] = size(pad_img);
    % fill up-pad
    x = mid_f_m;
    for i=1:mid_f_m - 1
        x = x - 1;
        for j=1:n2
            if j < mid_f_n
                y = mid_f_n - j;
            elseif j >= mid_f_n + n
                y = n - (j - (n + mid_f_n));
            else
                y = j - mid_f_n + 1;
```

```
            end
            pad_img(i,j) = image(x,y);
        end
    end
    % fill down-pad
    x = m + 1;
    for i=m+mid_f_m:m2
        x = x - 1;
        for j=1:n2
            if j < mid_f_n
                y = mid_f_n - j;
            elseif j >= mid_f_n + n
                y = n - (j - (n + mid_f_n));
            else
                y = j - mid_f_n + 1;
            end
            pad_img(i,j) = image(x,y);
        end
    end
    % fill left-pad, right-pad
    for i=mid_f_m:m+mid_f_m-1
        x = i - mid_f_m + 1;
        for j=1:mid_f_n-1
            y = mid_f_n - j;
            pad_img(i,j) = image(x,y);
        end
        for j=n+mid_f_n:n2
            y = n - (j - (n + mid_f_n));
            pad_img(i,j) = image(x,y);
        end
    end
```

above code is "Pad with reflected image content". I divide the outer part to be padded into 4 parts and find the symmetric parts in each part and map the pixel value.

```
for c=1:channel
    conv_fft = ifft2(fft2(pad_img(:,:,c),m2+f_m-1,n2+f_n
       -1).*fft2(filter,m2+f_m-1,n2+f_n-1));
    output(:,:,c) = conv_fft(f_m:m+f_m-1, f_n:n+f_n-1);
end
```

above code is about "FFT-based convolution.". I use fact that $g * h = F^{-1}[F[g]F[h]]$ When execute FFT of g,h bacause image is 2D discrete, make 2 matrix become same size(size of $g * h$) with padding. And after inverse FFT, I ignore padding part.

A Result

Figure 1: left : low frequencies of dog, right : high frequencies of cat

Figure 2: hybrid image (dog and cat)

Figure 3: hybrid image scales

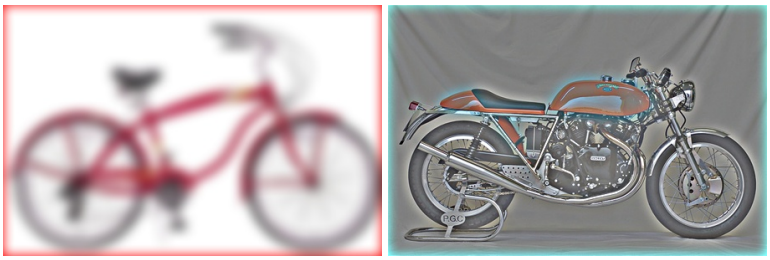

Figure 4: left : low frequencies of bicycle, right : high frequencies of motorcycle

Figure 5: hybrid image (bicycle and motorcycle)

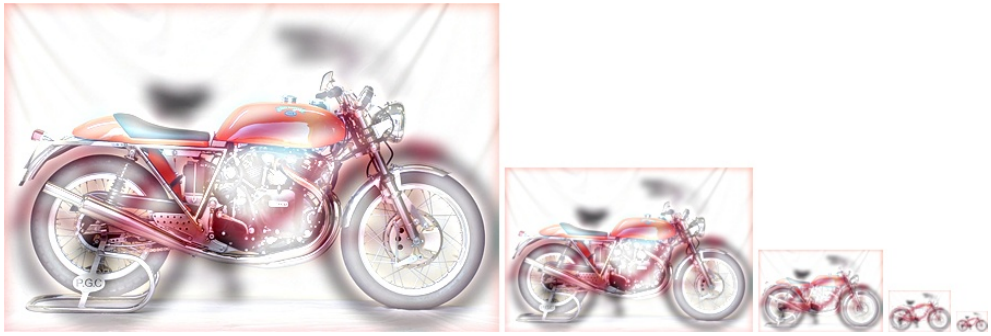

Figure 6: hybrid image scales