

## Project 1 Writeup (Hybrid Images)

### 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 incline on the short side. If you feel you only need to write a short amount to meet the brief, then do it.
- **Please make this document anonymous.**

### 1 Criação da Imagem Híbrida

Para a criação da imagem híbrida, as imagens presentes na Figura 2 foram obtidas a partir das imagens da Figura 1.

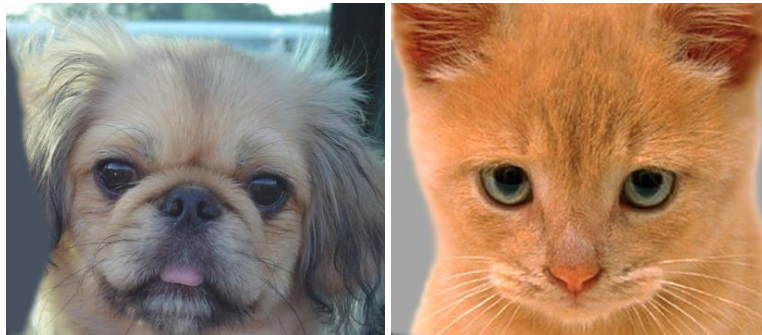


Figure 1: Imagens originais.



Figure 2: *Esquerda:* Após aplicação do filtro passa baixa. *Direita:* Após aplicação do filtro passa baixa e subtração do resultado da imagem original.

## 1.1 Modificações na Implementação

O seguinte trecho define as modificações realizadas no arquivo *gen\_hybrid\_image.m*. As linhas 2 e 3 foram adicionadas: 2 tem como objetivo aplicar o filtro passa baixa na imagem *dog*; 3 subtrai a imagem *cat* (após a aplicação do filtro passa baixa) da imagem *cat* original. A linha 4, por sua vez, realiza a soma das altas frequências do *cat* com as baixas frequências do *dog*.

```
1 filter = fspecial('Gaussian', cutoff_frequency*4+1,  
    cutoff_frequency);  
2 low_frequencies = my_imfilter(image1, filter);  
3 high_frequencies = image2 - my_imfilter(image2, filter);  
4 hybrid_image = low_frequencies + high_frequencies;
```

## 2 Criação da Imagem Híbrida FFT

De forma semelhante à criação da imagem híbrida anterior, as imagens presentes na Figura 3 foram obtidas a partir das imagens da Figura 1.

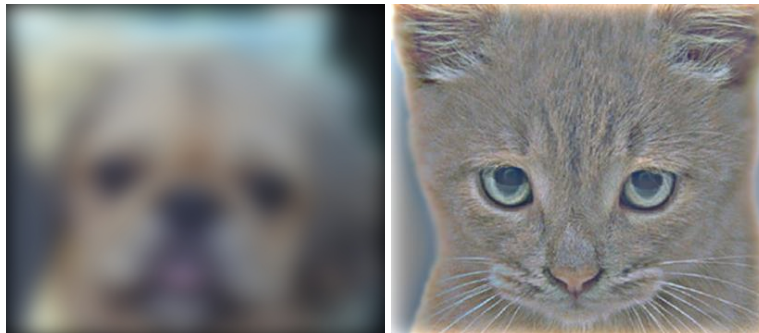


Figure 3: *Esquerda:* Após aplicação do filtro passa baixa. *Direita:* Após aplicação do filtro passa baixa e subtração do resultado da imagem original. (*Baseado em FFT.*)

### 2.1 Modificações na Implementação

No seguinte código, foi necessário aplicar o filtro passa baixa nos três canais de ambas imagens (linhas 2, 3 e 4). Dessa forma, a função *cat* (linhas 5 e 6) é utilizada para concatenar os três *arrays* (dimensões da imagem). O processo para obtenção das altas frequências do *cat* e da imagem híbrida (linhas 7, 8 e 9) é semelhante à descrição presente na Seção 1.1.

```
1 cutoff_frequency = 10;  
2 [low_frequencies_dog1, low_frequencies_cat1] =  
    gen_hybrid_image_fft( image1, image2, cutoff_frequency  
        , 1 );
```

```

3 [low_frequencies_dog2,low_frequencies_cat2] =
    gen_hybrid_image_fft( imagel, image2, cutoff_frequency
      , 2 );
4 [low_frequencies_dog3,low_frequencies_cat3] =
    gen_hybrid_image_fft( imagel, image2, cutoff_frequency
      , 3 );
5 low_frequencies_dog = cat(3, low_frequencies_dog1,
    low_frequencies_dog2, low_frequencies_dog3);
6 low_frequencies_cat = cat(3, low_frequencies_cat1,
    low_frequencies_cat2, low_frequencies_cat3);
7 high_frequencies = image2 - low_frequencies_cat;
8 low_frequencies = low_frequencies_dog;
9 hybrid_image = high_frequencies + low_frequencies;

```

O seguinte trecho de código especifica o conteúdo do arquivo *gen\_hybrid\_image\_fft.m*. Neste, é importante expor as seguintes características:

- A variável *X* é responsável por indicar o canal da imagem.
- A função *gen\_hybrid\_image\_fft.m* é invocada três vezes para os três canais de cada imagem envolvida. Dessa forma, o código para a *image1* tem início da linha 4 e fim na linha 21; o código para a *image2* tem início na linha 24 e fim na linha 42 (sendo, portanto, possível aplicar diferentes valores para a variável *D0* da função *H*, independente da imagem).

```

1 function [low_frequencies_dog,low_frequencies_cat] =
    gen_hybrid_image_fft( imagel, image2, cutoff_frequency
      , X )
2
3 #####
4 ##### IMAGE1.toLowFrequencies #####
5 b = padarray(imagel, size(imagel), "zeros", "post");
6 c = im2double(b(:, :, X:3));
7 d = fft2(c);
8 d = fftshift(d);
9 [n m o] = size(c);
10 h = zeros([n,m]);
11 for i = 1:n
12     for j = 1:m
13         h(i,j) = H(i,j,size(c),cutoff_frequency);
14     end
15 end
16 g = d.*h;
17 g = ifftshift(g);
18 at = ifft2(g);
19 at = abs(at);

```

```
20 [x y o] = size(image1);
21 low_frequencies_dog = at(1:x,1:y);
22
23 #####
24 ##### IMAGE2.toLowFrequencies #####
25 b = padarray(image2, size(image2), "zeros", "post");
26 c = im2double(b(:, :, X:3));
27 d = fft2(c);
28 d = fftshift(d);
29 [n m o] = size(c);
30 h = zeros([n,m]);
31 for i = 1:n
32     for j = 1:m
33         h(i,j) = H(i,j,size(c),cutoff_frequency);
34     end
35 end
36 g = d.*h;
37 g = ifftshift(g);
38 at = ifft2(g);
39 at = abs(at);
40 [x y o] = size(image2);
41 atc = at(1:x,1:y);
42 low_frequencies_cat = atc;
```

### 3 Resultados Obtidos

As Figuras 4 e 5 possuem as imagens híbridas resultantes das Figuras 2 e 3, respectivamente (em diferentes proporções para visualização do efeito produzido).



Figure 4: Imagem híbrida obtida a partir das imagens da Figura 2



Figure 5: Imagem híbrida baseada em FFT obtida a partir das imagens da Figura 3