

## Project 1 Questions

### Instructions

- 4 questions.
- Write code where appropriate.
- Feel free to include images or equations.
- **Please use only the space provided and keep the page breaks.** Please do not make new pages, nor remove pages. The document is a template to help grading.
- If you really need extra space, please use new pages at the end of the document and refer us to it in your answers.

### Questions

**Q1:** Explicitly describe image convolution: the input, the transformation, and the output. Why is it useful for computer vision?

**A1:** *Input:* imagem que será modificada. Transformação: utilização de um *kernel* (matriz que pode possuir diferentes tipos de padrões), a qual é aplicada com o objetivo de modificar os valores originais da imagem. *Output:* imagem modificada por meio do método da convolução. A técnica da convolução pode ser útil na visão computacional nas seguintes situações: detecção de bordas, aumentar os detalhes visíveis da imagem ou aprimorá-la para determinada finalidade.

**Q2:** What is the difference between convolution and correlation? Construct a scenario which produces a different output between both operations.

*Please use `imfilter` to experiment! Look at the 'options' parameter in MATLAB Help to learn how to switch the underlying operation from correlation to convolution.*

**A2:** A diferença entre a convolução e a correlação relaciona-se à rotação de  $180^\circ$ , aplicada apenas ao *kernel* da correlação durante o processamento da imagem. O seguinte cenário produz imagens distintas: aplicar filtros (*kernel*) assimétricos, a qual produzirá resultados diferentes para a convolução e correlação.

**Q3:** What is the difference between a high pass filter and a low pass filter in how they are constructed, and what they do to the image? Please provide example kernels and output images.

**A3:** *Low pass filter:* atenua as altas frequências, as quais são causadas por transições abruptas de intensidade, e deixa passar as baixas frequências (gerando *blur*). *High pass filter:* deixar passar as altas frequências e atenua as baixas frequências (aumenta os detalhes *edges* da imagem, mas reduz o contraste).

Com relação á construção dos filtros: em filtros passa baixa, por exemplo, em direção ao pixel central os pesos atribuídos são crescentes. Em filtros passa alta, o pixel central recebe maior peso quando os seus *neighbors* são diferentes.

1. Sobel (*high pass*):

$$filtroGx = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$filtroGy = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

2. *Gaussian (low pass):*

$$filtro = \begin{bmatrix} 1/256 & 4/256 & 6/256 & 4/256 & 1/256 \\ 4/256 & 16/256 & 24/256 & 16/256 & 4/256 \\ 6/256 & 24/256 & 36/256 & 24/256 & 6/256 \\ 4/256 & 16/256 & 24/256 & 16/256 & 4/256 \\ 1/256 & 4/256 & 6/256 & 4/256 & 1/256 \end{bmatrix}$$



Figure 1: Imagem original.

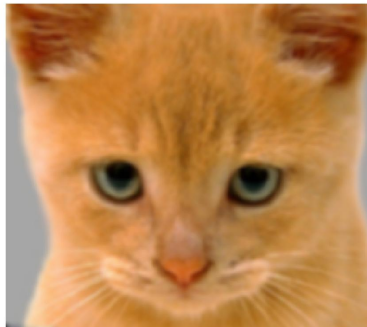


Figure 2: *output* da aplicação do filtro gaussiano na Imagem 1.



Figure 3: Imagem original.

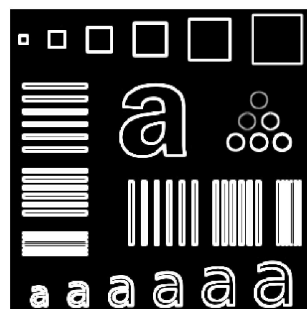


Figure 4: *output* da aplicação do filtro de Sobel na Imagem 3.

**Q4:** Explain the code in file *gen\_hybrid\_image\_fft.m*. What each line is supposed to do? What does the function *H()* do?

**A4:** Your answer here.

Explicação do conteúdo do arquivo *gen\_hybrid\_image\_fft.m*, de acordo com o trecho de código apresentado:

- *Linha 5:* realiza o *zero padding* na imagem.
- *Linha 6:* transforma determinado canal da imagem para *double*.
- *Linha 7:* aplica a transformada rápida de Fourier 2D.
- *Linha 8:* centraliza a transformada de Fourier 2D, movendo a frequência zero (posição  $(u = 0, v = 0)$ ) para o centro da matriz  $(P/2, Q/2)$ .
- *Linha 9:* seleção da região  $n \times m$  da matriz.
- *Linha 10:* cria uma matriz de zeros com dimensões  $n$  e  $m$ .
- *Linhas 11 a 15:* preenche a matriz com os valores obtidos a partir da função do filtro passa baixa gaussiano.
- *Linha 16:* multiplicação elemento por elemento da matriz da transformada de Fourier pelo filtro passa baixa gaussiano.
- *Linha 17:* move a frequência zero para a posição original  $(u = 0, v = 0)$ .
- *Linha 18:* aplica a transformada inversa rápida de Fourier 2D.
- *Linha 19:* seleção da parte real da matriz, ignorando os componentes complexos.
- *Linhas 20 e 21:* seleção e extração da região  $x \times y$  da matriz.
- *Linhas 25 a 42:* semelhante às linhas 5 a 21.

```

1 function [low_frequencies_dog, low_frequencies_cat] =
   gen_hybrid_image_fft( image1, image2, cutoff_frequency
   , X )
2
3 #####
4 ##### IMAGE1.toLowFrequencies #####
5 b = padarray(image1, size(image1), "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;

```

A função  $H$  indica o formato dos filtros passa baixa gaussianos de duas dimensões. Para determinar o resultado da função  $H$ , considera-se as seguintes variáveis:  $D(u, v)$  (distância entre um ponto  $(u, v)$  no domínio da frequência e o centro da função de frequência);  $D0$  (medida de espalhamento gaussiana em torno de seu centro).

```

1 function res = H(u,v,s,d0)
2     res = e^-( (D(u,v,s))^2 / (2*d0^2) );
3 endfunction

```