Project 1 Questions

Questions

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

A1: A convolução de uma imagem envolve três matrizes bidimensionais, que são: a imagem de entrada, aonde será aplicado a máscara, a máscara, que será aplicada na imagem de entrada e a imagem de saída, que será o resultado da máscara aplicado na imagem de entrada. Os passos para realizar a convolução em uma imagem são:

- Fazer a borda de zeros na imagem de entrada (para um filtro de dimensões de m linhas e n colunas, preenchemos com zeros a imagem com um mínimo de m-1 linhas acima e abaixo e n-1 colunas à esquerda e à direita), como pode ser visto no exemplo abaixo.

$$f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 1: exemplo de imagem de entrada.

$$w = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Figure 2: exemplo de máscara que será aplicada na imagem de entrada.

$$f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 3: exemplo da imagem de entrada com a borda.

- Antes de utilizar a máscara na convolução a mesma deve ser rotacionado em 180º.

$$w = \begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

Figure 4: máscara rotacionada em 180º.

- Agora aplicamos na imagem de entrada a máscara, pela fórmula abaixo.

$$w(x,y)f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)f(x-s, y-t) \tag{1}$$

Figure 5: fórmula da convolução.

- Após aplicar a fórmula acima devemos obter o seguinte resultado.

$$w(x,y)f(x,y) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 8 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 6: imagem de entrada após a aplicação da máscara.

- Por fim, devemos remover a borda de zeros que adicionamos no primeiro passo.

$$w(x,y)f(x,y) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 7: imagem de entrada após a aplicação da máscara e sem a borda de zeros.

Uma das importâncias da convolução na visão computacional, é que ela nos permite, por exemplo: criar imagens híbridas. Para criar esse tipo de imagem, é necessário duas imagens de entradas, sendo que uma delas nos interessa somente as baixas frequências e a outra somente as altas frequências. Para obter essas duas imagens, podemos utilizar a transformada rápida de Fourier em que a mesma emprega a convolução, que é o processo de mover uma máscara pela imagem e calcular a soma dos produtos em cada posição, como foi explicado anteriormente.

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 principal diferença entre a correlação e a convolução, é a máscara que se utiliza nos processos. A máscara que se utiliza na correlação está na figura 2, e a máscara que da convolução na figura 4, que basicamente é a máscara da correlação rotacionada em 180º. O processo de aplicar a máscara na imagem de entrada é o mesmo, em que movemos a máscara pela imagem e calcuamos a soma dos produtos em cada posição. Um cenário que podemos ver a diferença de saída entre essas duas operações é aplicando as máscaras que estão na figura 2 e 4 na matriz bidimensional de uma imagem de entrada, que está na figura 1. O resultado que podemos ver da correlação e da convolução, pode ser visto respectivamente abaixo.

$$w(x,y)f(x,y) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 8 & 7 & 0 \\ 0 & 6 & 5 & 4 & 0 \\ 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 8: resultado da correlação, após a aplicação da máscara na imagem de entrada.

$$w(x,y)f(x,y) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 0 \\ 0 & 4 & 5 & 6 & 0 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 9: resultado da convolução, após a aplicação da máscara na imagem de entrada.

Com relação a função imfilter presente no MATLAB, se você passar para essa função apenas como parâmetro a imagem de entrada e o filtro, ele irá realizar a opção definida por padrão que no caso é a correlação, que pode ser visto na primeira linha do código abaixo. Caso você deseje realizar a operação de convolução, além de passar a matriz bidimensional da imagem de entrada e o filtro, você deve passar um terceiro parâmetro que no caso é "conv", que se refere a operação de convolução. Essa invocação de função pode ser visto na segunda linha do código abaixo.

```
1 imfilter(entrada, filtro)
2 imfilter(entrada, filtro, 'conv')
```

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: A diferença entre um filtro passa-alta de um filtro passa-baixa, é que o filtro passa-alta aceita altas frequências, e tem como papel realçar as pequenas diferenças locais, associadas às frequências espaciais altas. O filtro passa-baixa é um filtro que aceita baixas frequências, e produz na imagem quando aplicado um efeito de borramento, de suavizar a imagem. O filtro passa-baixa pode ser montado de três maneiras, podendo ser um filtro passa-baixa ideias, passa-baixa Butterworth e passa-baixa gaussianos. Para montar o passa-baixa ideais, deve ser utilizar a seguinte função:

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

onde D_0 é uma constante positiva, e $D(u,v)$ é a distância entre um ponto (u, v) no domínio da frequência e o centro do retângulo de frequência. Já $D(u,v)$ é obtido através da seguinte fórmula:

$$D(u,v) = \sqrt{(u - P/2)^2 + (v - Q/2)^2} \tag{2}$$

onde P e Q são dados pelas respectivas fórmulas:

$$P \geq 2 * M - 1 \tag{3}$$

$$Q \geq 2 * N - 1 \tag{4}$$

onde M e N são altura e largura da imagem onde será aplicado o filtro, respectivamente. Uma outra maneira de construir um filtro passa-baixa, é você construir um filtro passa-baixa Butterworth pela seguinte fórmula:

$$H(u,v) = 1/1 + [D(u,v)/D_0]^{2n} \tag{5}$$

onde n é a ordem do filtro. Por fim, a última maneira de se construir um filtro passa-baixa é o mesmo sendo um filtro passa-baixa gaussiano que pode se utilizar a seguinte fórmula:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2} \tag{6}$$

onde e é o número de Euler. Já para o filtro passa-alta também tem três opções que são: filtro passa alta-ideiais, o Butterworth e o gaussianos, em que a fórmula para suas construções podem ser vistar abaixo, respectivamente:

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

$$H(u,v) = 1/1 + [D_0/D(u,v)]^{2n} \tag{7}$$

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2} \tag{8}$$

Abaixo iremos mostrar uma imagem na qual aplicaremos um filtro passa-baixa gaussiano, uma parte do filtro passa-baixa gaussiano e o resultado obtido, respectivamente.

Figure 10: imagem de entrada aonde será aplicado o filtro passa-baixa gaussiano.

$$\begin{bmatrix} 7.0015e-72 & 1.3262e-71 & 2.5056e-71 & 4.7223e-71 & \ldots \\ 1.3262e-71 & 2.5119e-71 & 4.7459e-71 & 8.9445e-71 & \ldots \\ 2.5056e-71 & 4.7459e-71 & 8.9669e-71 & 1.6900e-70 & \ldots \\ 4.7223e-71 & 8.9445e-71 & 1.6900e-70 & 3.1850e-70 & \ldots \\ 8.8777e-71 & 1.6815e-70 & 3.1771e-70 & 5.9877e-70 & \ldots \\ 1.6648e-70 & 3.1533e-70 & 5.9578e-70 & 1.1229e-69 & \ldots \\ 3.1142e-70 & 5.8985e-70 & 1.1145e-69 & 2.1004e-69 & \ldots \\ 5.8107e-70 & 1.1006e-69 & 2.0795e-69 & 3.9191e-69 & \ldots \\ 1.0815e-69 & 2.0485e-69 & 3.8705e-69 & 7.2945e-69 & \ldots \\ 2.0080e-69 & 3.8033e-69 & 7.1859e-69 & 1.3543e-68 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Figure 11: parte do filtro passa-baixa gaussiano.

Figure 12: resultado após a aplicação do filtro passa-baixa gaussiano.

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: Abaixo está o código do arquivo gen_hybrid_image_fft.m.

```
% Cria um padding
b = padarray(image1, size(image1), "zeros", "post");

% Converte para double
c = im2double(b(:, :, 1:3));

%Faz o padding da imagem
d = fft2(c);

%Centraliza a transformada de fourier
d = fftshift(d);

% Pega as dimensoes da vaariavel c
[n m o] = size(c);

% Faz uma matriz de zeros com as dimensoes de n e m
h = zeros([n, m]);

%Construindo o filtro passa-baixa
for i = 1:n
  for j = 1:m
    h(i, j) = H(i, j, size(c), cutoff_frequency);
  end
end

% Multiplicando a matriz de transformada de Fourier pelo filtro
g = d.*h;
```

```

% Descentralizando a matriz
g = ifftshift(g);

% Aplicando a transformada inversa rapida
at = ifft2(g);

% Tira os valores negativos de at
at = abs(at);

% Pega as dimensoes da imagem 1
[x y o] = size(image1);

% Extrai da regiao X e Y
atc = at(1:x, 1:y, :);

% Atribuindo a imagem final
low_frequencies = atc;

%%%%%%%%%%%%%%%%%%%%%%
%%%% ALTA FREQUENCIA
%%%%%%%%%%%%%%%%%%%%%%
% Cria um padding
b = padarray(image2, size(image2), "zeros", "post");

% Converte para double
c = im2double(b(:, :, 1:3));

%Faz o padding da imagem
d = fft2(c);

%Centraliza a transformada de fourier
d = fftshift(d);

% Pega as dimensoes da vaariavel c
[n m o] = size(c);

% Faz uma matriz de zeros com as dimensoes de n e m
h = zeros([n, m]);
for i = 1:n
  for j = 1:m
    h(i, j) = H(i, j, size(c), cutoff_frequency);
  end
end

% Inverter a transformada
```

```
invert = ones(size(im2uint8(h)));
h = invert .- h;

% Multiplicando a matriz de transformada de Fourier pelo
    filtro
g = d.*h;

% Descentralizando a matriz
g = ifftshift(g);

% Aplicando a transformada inversa rapida
at = ifft2(g);

% Pega as dimensoes da imagem 2
[x y o] = size(image2);

% Extrai da regiao X e Y
atc = at(1:x, 1:y, :);

% Atribuindo a imagem final
high_frequencies = atc;

% Combine the high frequencies and low frequencies
hybrid_image = abs(low_frequencies + high_frequencies);
```