

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: Convolução trata-se do processo de adicionar cada elemento da imagem aos seus vizinhos locais. basicamente uma operação de matriz em escala. Seus componentes podem ser descritos como:

entrada (input): matriz da imagem que será realizada a operação

transformação: matriz da imagem (entrada) * filtro a ser aplicado. Para cada pixel da imagem é feito a multiplicação pelo valor do filtro multiplicado. resumidamente pode ser especificada como uma multiplicação matricial mais uma adição de vetores.

saída: matriz da imagem gerada pela transformação (imagem final)

Esse processo é útil pois pode ser usado para desfocar imagens, destacar certas partes, detectar bordas, aguçar e realçar aspectos da imagem, etc.

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 as duas é que a convolução trata-se de uma correlação aplicada em filtro aos 180°. Casos onde o filtro é simétrico, os dois se comportam exatamente do mesmo jeito. porém em filtros assimétricos podemos ver uma grande diferença.

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: O filtro passa-baixo é o tipo de filtro de domínio de frequência usado para suavizar a imagem. Atenua os componentes de alta frequência e preserva os componentes de baixa frequência. Por isso é usado em casos onde queremos eliminar ruídos, pois ruídos são fontes de alta frequência, sendo assim são desfocados pelo filtro.

O filtro passa-alta é o tipo de filtro de domínio de frequência usado para tornar a imagem mais nítida. Atenua os componentes de baixa frequência e preserva os componentes de alta frequência. Por isso são usados quando queremos realçar a forma de objetos, por exemplo.



Original image

Figure 1: Imagem original

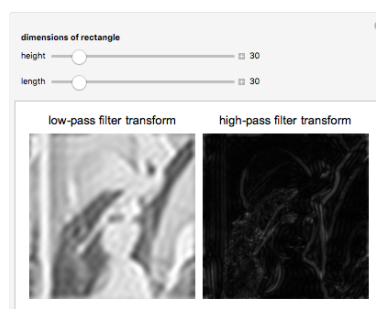


Figure 2: Aplicação dos filtros

Q4: Explain the code in file *gen_hybrid_image_fft.m*. What each line is supposed to do? What does the function $H()$ do?

```

g = 0:n,

g = ifftshift(g);
at = ifft2(g);
[x y o] = size(image1); % pega o tamanho da imagem 1
atc = at(1:x,1:y,:); % corta a imagem com o tamanho da imagem 1

figure, imshow(im2uint8(h)); % mostra a imagem 2 invertida
figure, imshow(low_frequencies = atc); % resultante em baixa frequencia

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Remove the low frequencies from image2. The easiest way to do this is to
% subtract a blurred version of image2 from the original version of image2.
% This will give you an image centered at zero with negative values.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
high_frequencies = [];

You, há 5 horas • chore: file template added ...
tam_img2 = size(image2); % pega o tamanho da imagem 2
img_2_w_0 = padarray(image2, tam_img2, "zeros", "post"); % adiciona zeros a imagem 2
img_2_c_double = im2double(img_2_w_0(:,1:3)); % converte a imagem 2 para double
tam_double = size(img_2_c_double); % pega o tamanho da imagem 2 convertida para double
img_fourier = fftshift(fft2(img_2_c_double)); % faz a transformada de fourier da imagem 2 e centraliza

[n m o] = size(img_2_c_double); % pega o tamanho da imagem 2
mat_w_0 = zeros([n,m]); % cria uma matriz de zeros com o tamanho da imagem 2
for i = 1:n
    for j = 1:m
        mat_w_0(i,j) = H(i,j,tam_double,cutoff_frequency); % calculo do filtro
    end
end

mat_w_0 = imcomplement(mat_w_0); % inverte a imagem 2

img_2_res = img_fourier.*mat_w_0; % multiplica a imagem 2 transformada pelo filtro

trans_inv = ifft2(ifftshift(img_2_res)); % centraliza a imagem 2 invertida e faz a transformada inversa da imagem 2 invertida
[x y o] = size(image1); % pega o tamanho da imagem 1
atc = trans_inv(1:x,1:y,:); % pega a imagem 2 invertida centralizada

figure, imshow(image2); % mostra a imagem 2
figure, imshow(uint8(abs(img_fourier))); % mostra a imagem 2 da transformada de fourier
figure, imshow(high_frequencies = atc); % resultante

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

figure, imshow(hybrid_image); % mostra a imagem hibrida

```

A4: