

Projeto 01 Questões da Atividade Imagens Híbridas

Questões

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

A1: Convolução de imagens são operações em uma matriz realizada por outra matriz no domínio espacial, possibilitando assim manipular imagens, onde desloca -se a máscara sobre ela e calcula -se a soma dos produtos em cada local para a obtenção dos efeitos na imagem, entretanto a máscara é rotacionada em 180°, sendo diferente da correlação.

Inicialmente as imagens são inseridas como entrada logo são representadas como uma matriz, onde cada ponto desta matriz é referente à um pixel da imagem e o valor de cada ponto é referente as cores dela. Sendo assim, uma vez a imagem transformada em uma matriz de pixels é necessário aplicar algum tipo de operação nela para realizar determinados efeitos na imagem, como por exemplo os efeitos de desfoque, borrão, realce de bordas e contornos, são obtidos após a multiplicação de cada ponto da imagem pela máscara correspondente , sendo esta máscara entendida como o tipo de efeito desejado na foto. Por fim após a soma dos produtos em em cada local da matriz, pois se trata de operações matemáticas nas linhas e colunas de uma matriz é resultado na imagem final com o efeito aplicado.

Logo conhecer estas técnicas de manipulação de imagens por meio de operações matemáticas envolvendo matrizes, é de grande importância para a área da visão computacional, uma vez que soluções para esse aspecto de problema possam ser resolvidos utilizando técnicas de correlação ou convolução, permitindo assim maiores possibilidades de gerenciamento dos efeitos ou qualidade das imagens.

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

A2: As diferenças entre as duas operações é ressaltado a seguir:

Correlação: A máscara gerada é deslocada sobre a imagem pixel a pixel, calculando a soma dos produtos em cada local;

Convolução: A máscara gerada também é deslocada sobre a imagem pixel a pixel, calculando a soma dos produtos em cada local, entretanto a máscara é rotacionada em 180° antes de aplicar as operações.

A diferença entre as duas operações está na rotação de 180° que uma operação recebe, logo os resultados de ambas as operações só irão ser diferentes se as máscaras não forem simétricas. A seguir o cenário onde os resultados se tornam diferentes, por causa desta rotação aparecem linhas horizontais e verticais. Para este exemplo foi utilizado a imagem `"../Data/re.bmp"`.

```
1  filtro1 = [0,0,0; -1,1,0; 0,0,0];  
2  filtro2 = [0,0,0; 0,1,0; 0,-1,0];  
3  
4  image_01 = imread("Data/re.bmp");  
5  image_01 = rgb2gray(image_01);  
6  
7  image_final_conv = conv2(image_01,filtro1);  
8  image_final_corr = conv2(image_01,filtro2);
```

A seguir o conjunto de imagens apresentado pela Figura 1, são os resultados obtidos após a execução do código. A primeira imagem mais a esquerda foi utilizado a técnica de convolução, enquanto na foto mais a direita foi utilizado a correlação.



Figure 1: Conjunto de imagens com convolução e correlação da esquerda para a direita.

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: As diferenças entre os dois filtros é ressaltado a seguir:

Filtro Passa Baixa: Este filtro permite que ondas de baixas frequência passem enquanto atenua a amplitude das frequências maiores que a frequência de corte. Este tipo de filtro causa um efeito de borrão na imagem, pois ondas de baixa frequência estão ligadas ao esboço dos objetos presentes na foto.

Filtro Passa Alta: Este filtro permite que ondas de alta frequência passem, enquanto atenua as ondas de baixa frequência. Este tipo de filtro realça os contornos dos objetos na imagem, uma vez que as altas frequências estão ligadas aos detalhes dos objetos contidos na foto.

A seguir é apresentado o trecho de código para extrair as altas e baixas frequências, podendo ser acessado através do diretório *"Hybrid-Images/Gerador_Hybrid/Code.m"*

```
1     [...]
2     low-pass-Spectre = image-Fourier .* filtro; '
        aplicando filtro na imagem transformada
3
4     low-pass-Image = ifftshift(low-pass-Spectre); '
        obtendo passa baixa
5     low-pass-Image = ifft2(low-pass-Image); 'obtendo
        passa alta
```

A seguir o conjunto de imagens apresentado pela Figura 2, são referentes aos resultados da passa alta e baixa respectivamente. Para este exemplo foi utilizado a imagem *"../Data/bru.bmp"*.

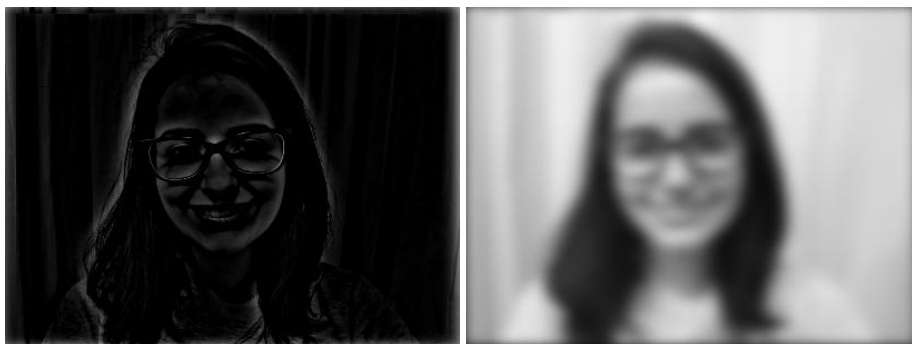


Figure 2: Resultados dos Filtros Passa Alta e Passa Baixa após a execução do código.

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: O código presente neste arquivo utiliza filtros no domínio da frequência, após a transformada da imagem para o espectro de fourier pela função "*fft()*", na qual é utilizada a transformada rápida de fourier, é gerado a matriz filtro que irá ser aplicada à imagem após a transformação de fourier, gerando assim uma atenuação nas altas frequências, causando um efeito de borrão na imagem, após a conversão da imagem de volta para o domínio espacial. Por fim, a imagem com o efeito de borrão é subtraída da imagem original resultando assim nos contornos dos objetos da imagem, representando as altas frequências.

```
1     image_padd = padarray(image_01, size(image_01), "zeros", "post"); 'padding da imagem
2     image_double = im2double(image_padd); 'convertendo a imagem para double
3     image_Fourier = fft2(image_double); 'transformada de fourier
4     image_Fourier = fftshift(image_Fourier); 'centralizacao da componente DC
5     [image_Y_Padd, image_X_Padd, image_Z_Padd] = size(image_double); 'obtendo tamanho da imagem
6     filtro = zeros([image_Y_Padd, image_X_Padd, image_Z_Padd]); 'alocando filtro
7
8     for i = (1:image_Y_Padd)
9         for j = (1:image_X_Padd)
10             filtro(i,j) = H(i,j,size(image_double), 20); 'gerando filtro
11         endfor
12     endfor
13
14     low_pass_Spectre = image_Fourier .* filtro; 'aplicando filtro na imagem transformada
15     low_pass_Image = ifftshift(low_pass_Spectre); 'descentralizacao da componente DC
16     low_pass_Image = ifft2(low_pass_Image); 'voltando a imagem para o donio espacial
17     final_HighPassImage = image_01 - im2uint8(abs(final_LowPassImage)); 'altas frequencias
18     final_LowPassImage = im2uint8(abs(final_LowPassImage)); 'baixas frequencias
```