

BCC36F – Processamento de Imagens

Prática 6

Prof^a. Dr^a. Aretha Barbosa Alencar
Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Campo Mourão

Nessa atividade prática vocês irão aplicar o **Filtro Passa-Baixa Gaussiano no domínio da frequência** em uma imagem.

1 Imagem de Entrada

A Figura 1 mostra a imagem de entrada ‘pratica6.png’, que possui 256 níveis de intensidade de cinza e tem tamanho 256×256 pixels.



Figura 1: Imagem de entrada ‘pratica6.png’.

2 Descrição Geral

Nessa atividade prática vocês irão aplicar o **Filtro Passa-Baixa Gaussiano no domínio da frequência** em uma imagem.

Passos necessários:

1. Carregar o pacote `image` no Octave.
2. Carregar a imagem de entrada ‘pratica6.png’.
3. Converter a imagem para *double*. Use a função: `im2double(...)`.

4. Fazer o *zero padding* na imagem e aplicar a transformada rápida de Fourier 2D sobre a imagem de entrada, use a função `fft2(A, m, n)`. Qual a dimensão da transformada? Responda essa questão no comentário do seu código.
5. Centralizem a transformada de Fourier dessa imagem movendo a frequência zero ($u = 0, v = 0$) para o centro da matriz ($P/2, Q/2$). Use a função `fftshift(x)`;
6. Mostrar o espectro de Fourier dessa imagem, calculando a magnitude da transformada de Fourier e convertendo esse resultado para `uint8`. Use as funções: `uint8(...)` e `abs(z)`.
7. Construir o filtro passa-baixa gaussiano de dimensão $P \times Q$, usando o código abaixo. Observe que as variáveis P e Q devem estar definidas previamente. O que as duas funções do código abaixo fazem? Qual o valor de D_0 nesse filtro? Responda essas questões nos comentários do seu código.

```
function res = D(u,v,P,Q)
    res = sqrt((u-P/2)^2+(v-Q/2)^2);
endfunction

function res = H(u,v,P,Q,D0)
    res = e^(-(D(u,v,P,Q))^2/(2*D0^2));
endfunction

h = zeros(P,Q,'double');
for i = 1:P
    for j = 1:Q
        h(i,j) = H(i,j,P,Q,20);
    end
end
figure, imshow(im2uint8(h));
```

8. Multiplique a matriz da transformada de Fourier (resultado do passo 5) pelo filtro passa-baixa gaussiano construído (resultado do passo 7). Lembre-se que esta multiplicação é a multiplicação elemento por elemento (*element-by-element multiplication*).
9. Descentralize a matriz resultado do passo 8, movendo novamente a frequência zero para sua posição original. Use a função: `ifftshift(x)`.
10. Aplique a transformada inversa rápida de Fourier 2D, sobre o resultado do passo 9. Use a função: `ifft2(A)`.
11. Selecione apenas parte real da matriz do resultado do passo 10, ignorando os componentes complexos. Use a função: `real(z)`.
12. Obter o resultado final, extraíndo a região $M \times N$ do canto superior esquerdo da matriz resultado do passo 11. Olhe a documentação disponível em: <https://www.gnu.org/software/octave/doc/v4.0.1/Index-Expressions.html>.

13. Mostrar a imagem do resultado final, lembre-se de convertê-la antes novamente para `uint8(...)`.

3 Resultados Esperados

A Figura 2 exibe o **espectro de Fourier** da imagem de entrada ‘pratica6.png’

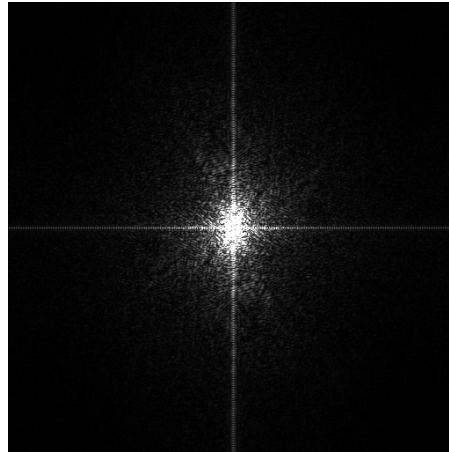


Figura 2: Espectro de Fourier da imagem de entrada ‘pratica6.png’.

Já a Figura 3 exibe o **filtro passa-baixa gaussiano** como uma imagem.

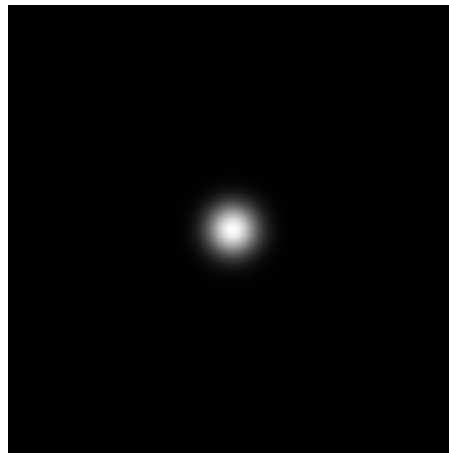


Figura 3: Filtro Passa-baixa Gaussiano como uma imagem.

Por fim, a Figura 4 exibe a imagem que apresenta o **resultado** da aplicação do filtro passa-baixa gaussiano no domínio da frequência sobre a imagem de entrada.

4 Documentação do Octave

A documentação das funções e sintaxe básicas do Octave encontra-se em https://www.gnu.org/software/octave/doc/interpreter/index.html#SEC_Contents.

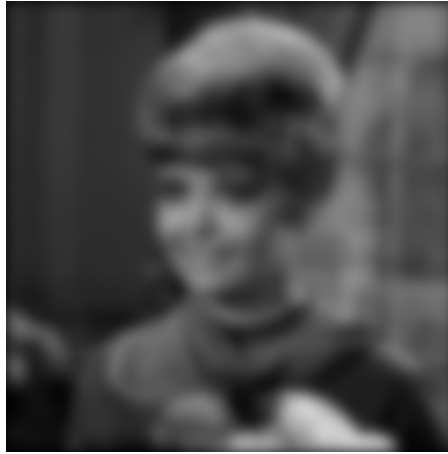


Figura 4: Resultado após a aplicação do filtro passa-baixa gaussiano na imagem de entrada no domínio da frequência.

Já a **documentação das funções do pacote `image`** do Octave encontra-se em <http://octave.sourceforge.net/image/overview.html>.

Documentação de funções específicas no Octave úteis para essa atividade:

- Funções `A = imread(...)` e `imwrite(...)`: <https://www.gnu.org/software/octave/doc/interpreter/Loading-and-Saving-Images.html>
- Função `B = im2double(A)`: <http://octave.sourceforge.net/image/function/im2double.html>
- Função `C = fft2(B, m, n)`: <http://octave.sourceforge.net/octave/function/fft2.html>
- Função `D = fftshift(C)`: <http://octave.sourceforge.net/octave/function/fftshift.html>
- Função `E = ifftshift(D)`: <http://octave.sourceforge.net/octave/function/ifftshift.html>
- Função `H = ifft2(E)`: <http://octave.sourceforge.net/octave/function/ifft2.html>
- Função `parte_real = real(z)`: <https://www.gnu.org/software/octave/doc/v4.0.1/Complex-Arithmetic.html#XREFreal>
- Função `magnitude = abs(z)`: <https://www.gnu.org/software/octave/doc/v4.0.1/Complex-Arithmetic.html#XREFabs>