

BCC36F – Processamento de Imagens

Prática 2

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

Nessa atividade prática vocês irão aplicar os **filtros de média e sobel** e uma imagem. Vocês também irão verificar os efeitos de duas abordagens diferentes para o **tratamento do limite da imagem** no processo de correlação.

1 Imagem de Entrada

A Figura 1 mostra a imagem de entrada ‘imagem.jpg’, que possui 256 níveis de intensidade de cinza e tem tamanho 306×306 pixels.



Figura 1: Imagem de entrada ‘imagem.jpg’.

2 Descrição Geral

Nessa atividade, você deve aplicar a técnica de correlação usando o filtro de média e os filtros de Sobel. Também iremos verificar o resultado dos efeitos de duas abordagens diferentes para o tratamento do limite da imagem no processo de correlação.

Passos necessários:

1. Carregar a imagem de entrada ‘imagem.jpg’.

2. Converter a imagem para *double*, use a função `im2double(...)`. Observe a nova faixa de valores de intensidade.
3. Monte um filtro de média com tamanho 9×9 .
4. Use a função `filter2(b, x, shape)` como forma para aplicar o filtro de média do passo 3 na imagem no formato *double* obtida no passo 2. Essa função irá fazer a correlação do filtro sobre a imagem. Nesse passo use o valor padrão para o parâmetro *shape*.
5. Exiba a imagem resultante do passo 4 e observe a borda da imagem. O que aconteceu e porquê? Responda essa pergunta nos comentários do código.
6. Vamos aplicar o filtro da média novamente só que agora vamos replicar os pixels da borda da imagem do passo 2 antes com a função `padarray(...)`. Forneça os parâmetros para essa função apropriadamente considerando o tamanho do filtro de média.
7. Aplique novamente o mesmo filtro da média, só que agora sobre a imagem com *padding* do passo 6, e exiba a imagem resultante.
8. Monte os filtros de Sobel (disponíveis nos slides).
9. Aplique os filtros de Sobel. No entanto, observe que a quantidade de *padding* necessário mudou, pois agora estamos trabalhando com filtros 3×3 . Ajuste o *padding* por replicação adequadamente.
10. Calcule a magnitude do gradiente como $M(x, y) = |g_x| + |g_y|$, e exiba essa imagem.
11. Observe também que todas as imagens de saída devem ter o mesmo tamanho da imagem original que é 306×306 pixels.

O uso das funções `J = imsmooth(I, name, options)` e `bw = edge(im, method)` do pacote `image` do Octave, não serão aceitos nessa atividade.

3 Resultados Esperados

A Figura 2 exibe a imagem de entrada com *padding* de zeros na borda após a aplicação do filtro de Média 9×9 .

Já a Figura 3 exibe a imagem de entrada com *padding* replicado na borda após a aplicação do filtro de Média 9×9 .

Por fim, a Figura 4 exibe a imagem de entrada com *padding* replicado na borda após a aplicação do filtro de Sobel. Vejam como as bordas dos elementos da imagem se tornam destacadas.



Figura 2: Imagem após filtro de Média com *padding* de zeros.



Figura 3: Imagem após filtro de Média com *padding* replicando os pixels da borda.

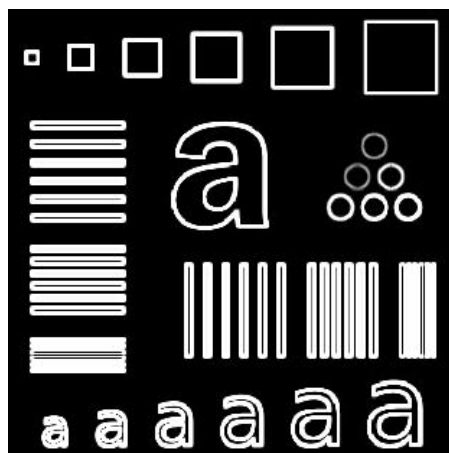


Figura 4: Imagem da magnitude do gradiente após filtros de Sobel.

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.

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 `imread(...)` e `imwrite(...)`: <https://www.gnu.org/software/octave/doc/interpreter/Loading-and-Saving-Images.html>
- Funções de utilidade para matrizes, como a função `ones(...)`: <https://www.gnu.org/software/octave/doc/interpreter/Special-Utility.html>
- Função `im2double(...)`: <http://octave.sourceforge.net/image/function/im2double.html>
- Função `y = filter2 (b, x, shape)`: <http://octave.sourceforge.net/octave/function/filter2.html>
- Função `I = padarray(I, padsize, pattern)`: <http://octave.sourceforge.net/image/function/padarray.html>
- Função `valor_absoluto = abs(z)`: <http://www.gnu.org/software/octave/doc/v4.0.1/Complex-Arithmetic.html#XREFabs>