

INF01046 – Fundamentos de Processamento de Imagens

Prof. Manuel M Oliveira

3º Trabalho de Implementação

Total de Pontos do Trabalho: 100

Objetivos

O objetivo deste trabalho é familiarizá-lo com conceitos fundamentais como convolução e filtragem no domínio espacial. Ao completá-lo, você terá compreendido:

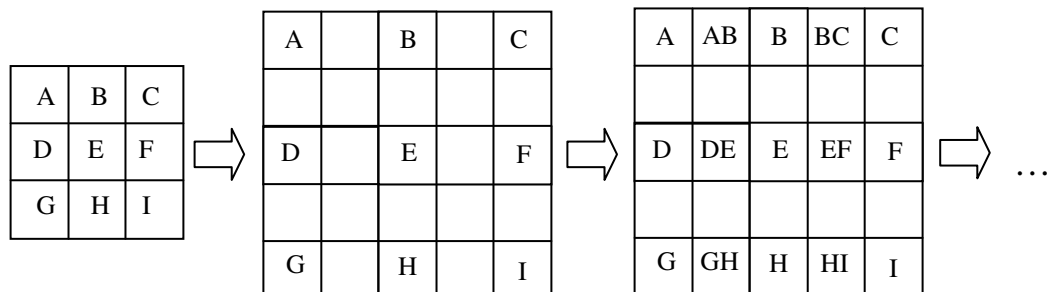
- a) O conceito de filtros separáveis e como eles podem ser utilizados para produzir filtros de mais alta ordem;
- b) Como aumentar (*zoom in*) e reduzir (*zoom out*) imagens;
- c) Como implementar convoluções para obter efeitos como borramento (filtros passa baixas), detecção de bordas (filtros passa altas) e *embossing*;
- d) Como rotacionar uma imagem de 90 graus.

Você se familiarizará com vários kernels úteis e compreenderá seus efeitos sobre imagens.

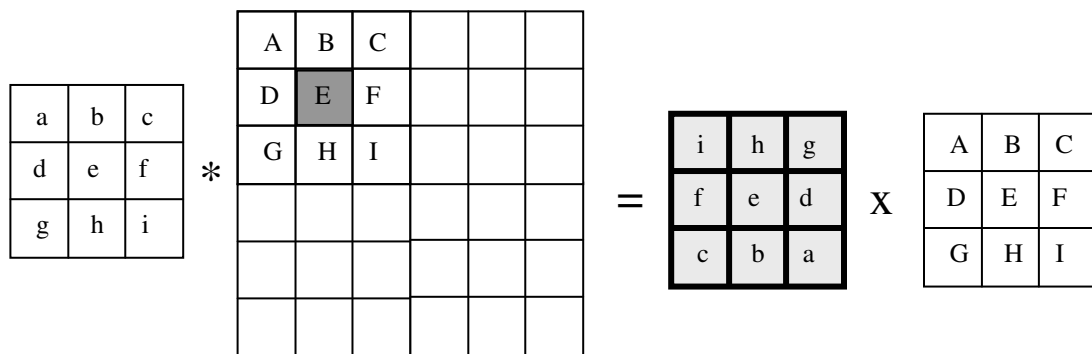
Descrição do Trabalho

Estenda desenvolvido como parte dos Trabalhos N° 1 e N° 2 para realizar as seguintes operações:

- a) Reduzir uma imagem (*zoom out*) utilizando fatores de redução s_x e s_y (s_x e $s_y \geq 1$), onde s_x não é necessariamente igual a s_y . Para tanto, defina um retângulo com dimensões s_y e s_x , e mova-o sobre a imagem, de modo que os retângulos nunca se sobreponham e que nenhum pixel da imagem deixe de ser coberto. Para cada posição do retângulo, calcule as médias (R, G e B) dos pixels sob o retângulo, utilizando estes resultados na imagem de saída. Caso o retângulo vá além das dimensões da imagem original, calcule as médias usando apenas os pixels sob o retângulo (**25 pontos**)
- b) Ampliar a imagem (*zoom in*) utilizando um fator de 2x2 a cada vez. Visto que a imagem será 4 vezes maior que a original, certifique-se de que você alocará memória para tal, apropriadamente. Implemente o processo de ampliação utilizando operações 1-D em dois passos. Primeiro, insira uma linha e uma coluna em branco entre cada duas linhas e colunas originais, respectivamente (veja ilustração na figura a seguir). Após, linearmente interpole os valores para preencher os espaços ao longo de todas as linhas (ou colunas) e, finalmente, ao longo de todas as colunas (ou linhas) (**25 pontos**).



- c) Implemente rotação de imagem de 90° (tanto no sentido horário como no sentido anti-horário). A operação de rotação deve poder ser aplicada múltiplas vezes de modo a permitir obter rotações de $\pm 180^\circ$, $\pm 270^\circ$, etc. **(15 pontos)**
- d) Implemente um procedimento para realizar convolução entre uma imagem e um filtro 3×3 arbitrário. De modo a simplificar sua implementação, ignore as bordas da imagem original e aplique a convolução apenas no seu interior. Além disso, aplique a convolução apenas a imagens de luminância. A figura a seguir ilustra a aplicação de uma operação de convolução ao pixel **E** utilizando um kernel 3×3 . Note que o kernel é rotacionado de 180° antes da aplicação. No exemplo do pixel E, o resultado é: $Conv(E) = iA + hB + gC + fD + eE + dF + cG + bH + aI$. Devido à soma de vários produtos e à possibilidade de utilização de pesos negativos, o resultado da convolução pode ser maior que 255 ou menor que zero. Na sua implementação, você deverá aproximar estes casos com os valores 255 e zero, respectivamente, para os casos (i) a (iii) abaixo. Para os demais, some 127 ao resultado da convolução antes de fazer esta aproximação (*clamping*). A interface do seu programa deverá aceitar valores arbitrários para os pesos dos filtros. Teste o seu procedimento com os kernels (i) a (vii) a seguir **(35 pontos)**.



i) **Gaussiano**, filtro pass baixas (produz borramento).

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

ii) **Laplaciano**, filtro passa altas (detecta arestas importantes).

0	-1	0
-1	4	-1
0	-1	0

iii) **Passa Altas Genérico** (detector de arestas mais sensitivo).