

Transformações Lineares, Equalização e Matching de Histograma, Convolução e Filtragem no Domínio Espacial

Total de Pontos do Trabalho: 200

OBJETIVO

O objetivo deste trabalho é familiarizar os estudantes com:

- 1) Cálculo de histograma, transformações lineares sobre pixels (ajuste de brilho e contraste e cálculo do negativo), equalização e “matching” de histograma;
- 2) Conceitos fundamentais como convolução e filtragem no domínio espacial.

Ao completá-lo, você terá compreendido:

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

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

Descrição do Trabalho

Estenda o programa desenvolvido para o Trabalho Nº 1 para realizar as seguintes operações:

Parte 1

- 1) (20 pontos) **Calcular e exibir o histograma de uma imagem em tons de cinza** (8 bits por pixel). Caso a imagem informada como entrada seja colorida, converta-a para tons de cinza (luminância) e então calcule seu histograma. Exiba o histograma em uma janela separada, onde cada coluna da imagem representa um tom de cinza. Normalize a altura das colunas para obter uma representação apropriada.
- 2) (15 pontos) **Ajustar o brilho de uma imagem** (e exibí-la), somando ao valor de cada pixel um escalar no intervalo $[-255, 255]$. Certifique-se que o resultado da operação aplicado a cada pixel se encontra na faixa $[0, 255]$, ajustando-o para zero ou 255 quando necessário. No caso de imagens coloridas, aplique o algoritmo para cada um dos canais (R, G, B) independentemente.
- 3) (15 pontos) **Ajustar o contraste de uma imagem** (e exibí-la), multiplicando cada pixel por um escalar no intervalo $(0, 255]$. Certifique-se que o resultado da operação aplicado a cada pixel se encontra na faixa $[0, 255]$, ajustando-o para 255 quando necessário. No caso de imagens coloridas, aplique o algoritmo para cada um dos canais (R, G, B) independentemente.

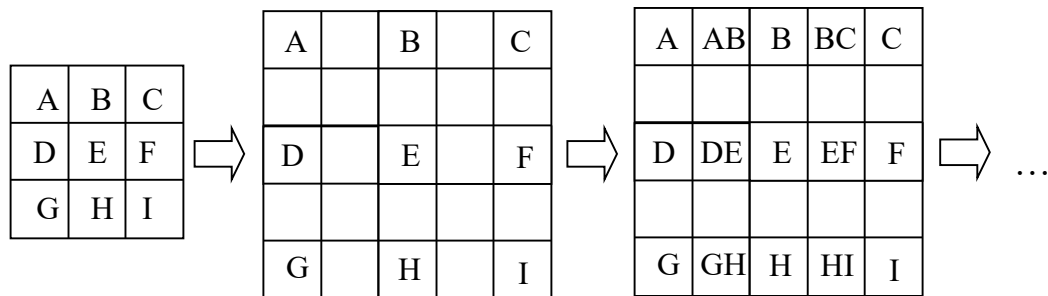
- 4) (10 pontos) **Calcular e exibir o negativo de uma imagem**, calculando o novo valor de cada pixel como: $\text{<novo valor>} = 255 - \text{<antigo valor>}$. No caso de imagens coloridas, aplique o algoritmo para cada um dos canais (R, G, B) independentemente.
- 5) (25 pontos) **Equalizar o histograma de uma imagem**, exibindo as imagens antes e depois da equalização. Para imagens em tons de cinza, exibir os histogramas antes e depois da equalização. No caso de imagens coloridas, para cada um dos canais (R, G, B), utilize o histograma cumulativo obtido a partir da imagem de luminância.

Pontos Extra: Implementação utilizando o espaço de cor L*a*b* (além da atividade acima) no caso de imagens coloridas (20 pontos).

- 6) (15 pontos) **Realizar Histogram Matching de pares de imagens em tons de cinza.**

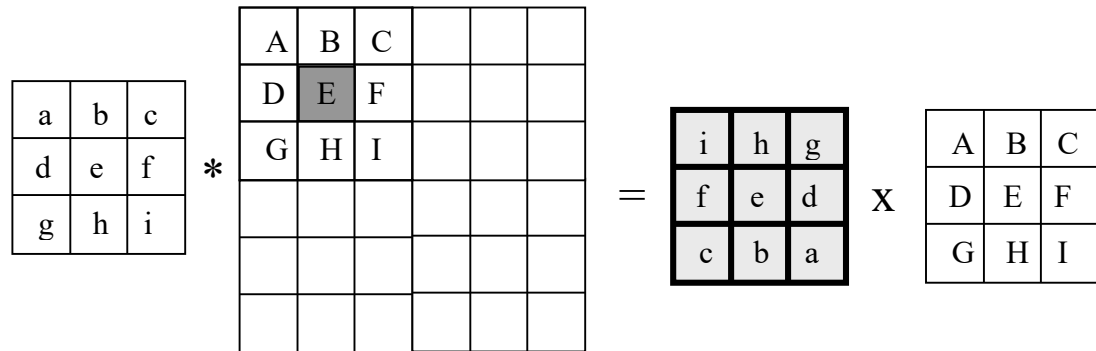
Parte 2

- 7) (25 pontos) **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.
- 8) (25 pontos) **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).



- 9) (15 pontos) **Rotacionar 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 +/- 180°, +/- 270°, etc.
- 10) (35 pontos) **Implementar um procedimento para realizar convolução entre uma imagem e um filtro 3x3 arbitrário.** De modo a simplificar sua implementação, ignore as bordas da imagem original e aplique a convolução apenas no seu interior. Exceto para o filtro passa baixas, 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 3x3. Note que o kernel é rotacionado de 180° antes da aplicação. No exemplo do pixel **E**, o resultado é: $\text{Conv}(E) = iA + hB + gC + fD + eE + dF + cG + bH + aI$. 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.



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 sensetivo).

| | | |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

iv) **Prewitt Hx** (gradiente dos tons de cinza na direção horizontal – efeito de relevo)

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -1 | 0 | 1 |
| -1 | 0 | 1 |

v) **Prewitt Hy** (gradiente dos tons de cinza na direção vertical – efeito de relevo)

| | | |
|----|----|----|
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

- vi) **Sobel Hx** (mais sensível ao gradiente dos tons de cinza na direção horizontal – efeito de relevo)

| | | |
|----|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

- vii) **Sobel Hy** (mais sensível ao gradiente dos tons de cinza na direção vertical – efeito de relevo)

| | | |
|----|----|----|
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| 1 | 2 | 1 |