MAC0300 - Métodos Numéricos da Álgebra Linear

Professor Walter Mascarenhas

2º EP - Métodos de Processamento de Imagens

Data de Entrega: 23/10/2012

1 Objetivo

O objetivo deste EP é implementar e testar alguns métodos simples, mas muito usados, de processamento digital de imagens em níveis de cinza.

Os métodos a serem implementados são:

- Ajuste de contraste por equalização de histograma;
- Suavização por filtro linear de média ponderada;
- Aumento de nitidez aplicando o operador Laplaciano.

Veja a explicação mais detalhada da implementação dos métodos na Seção "Métodos".

2 Especificações

2.0.1 Sistema Operacional

O programa deve compilar e funcionar no Linux, devendo ser executado pela linha de comando.

2.0.2 Linguagem de Programação

A implementação do programa pode ser realizada na linguagem a sua escolha, sendo recomendadas as linguagens: Octave, R e Python.

2.0.3 Parâmetros de Entrada

O programa deve receber por parâmetros da linha de comando o indicador do método a ser utilizado no processamento e o nome da imagem a ser processada.

Apenas um método pode ser dado por execução do programa e ele será indicado, após o símbolo "-", por um dos seguintes comandos:

- contrast – solicita o ajuste de contraste por equalização de histograma;
- blur – solicita suavização por média ponderada;
- sharpen – solicita realce através da aplicação do operador Laplaciano.

O nome do arquivo de imagem será da forma "nome.jpg". Exemplo: lena.jpg.

No exemplo, a execução de um programa em Octave, por um terminal, seria:
$ octave -contrast lena.jpg
$ octave -blur lena.jpg
$ octave -sharpen lena.jpg

2.0.4 Arquivo de Saída

O programa deve salvar, na pasta de execução do programa, o arquivo da imagem processada em formato jpg, de nome igual ao arquivo de entrada, seguido do sufixo "final". Exemplo: lena-final.jpg

3 Métodos

3.1 Ajuste de Contraste - Equalização de Histograma

É uma técnica de ajuste das intensidades da imagem para aumentar o contraste.

O objetivo é redistribuir as intensidades de níveis de cinza dos pixeis em uma imagem, de modo a obter um histograma uniforme, onde o percentual de pixeis de qualquer nível de cinza é praticamente o mesmo.

Uma imagem X em níveis de cinza geralmente é representada digitalmente por uma matriz de $M \times N$ pixeis. Suas entradas representam o valor da intensidade de cada pixel, assim, em geral, variam de 0 a 255.

Usualmente, usa-se a função de distribuição acumulada das intensidades dos pixeis da imagem, que é o histograma da imagem.

Para obter essa distribuição acumulada, calcula-se, para cada intensidade de pixel i:

$$fda_X(i) = \sum_{j=0}^{i} p_X(j),$$

onde $p_X(j)$ é a probabilidade da ocorrência do pixel de intensidade j na imagem X, ou seja:

$$p_X(j) = \frac{n_j}{n}, \quad 0 \leq j \leq 255$$

.

A função de distribuição acumulada obtida terá um valor mínimo $fda_{X_{min}}$ e um valor máximo $fda_{X_{max}}$ iguais aos valores mínimo e máximo de intensidade da imagem original. Como queremos que os níveis de intensidade se distribuam uniformemente no intervalo $[0, 255]$, devemos fazer a normalização dessa função.

Dessa forma, os pixeis da nova imagem, com histograma normalizado, será obtido a partir da seguinte forma:

$$h_X(i) = \text{round}\left(\frac{fda_X(i) - fda_{X_{min}}}{(M \times N) - fda_{X_{min}}} \times 255\right)$$

O efeito causado por essa técnica pode ser observado nas figuras 1 e 2, onde são mostradas a imagem original e após ajuste de contraste por equalização de histograma, ambas com seus respectivos histogramas.

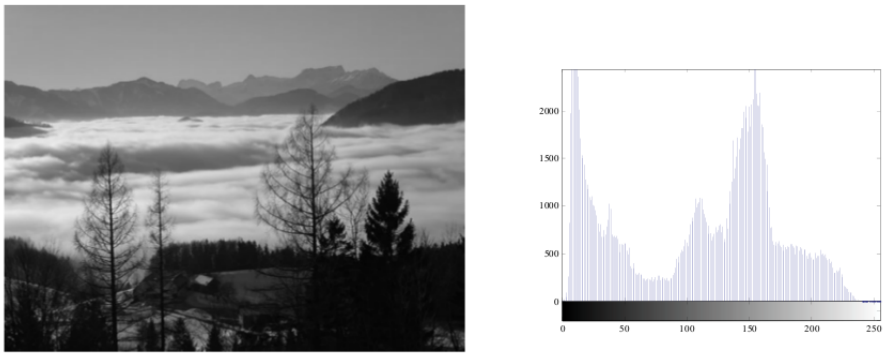

Fig. 1: *Imagem original e seu histograma.*

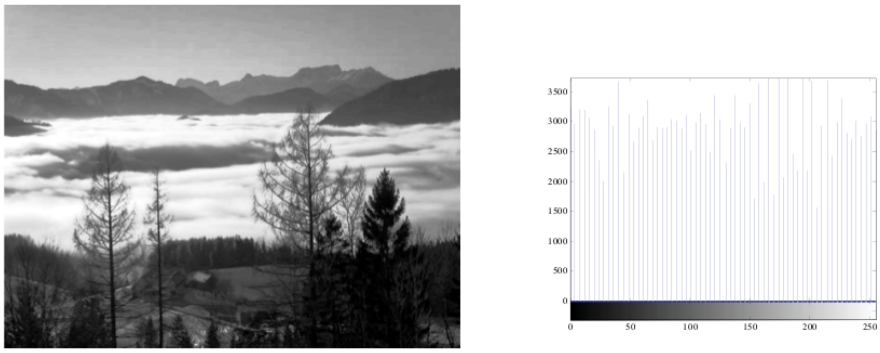

Fig. 2: *Imagem com ajuste de contraste por equalização de histograma e seu histograma.*

Para um melhor entendimento do método, veja o exemplo numérico e os resultados obtidos nas imagens na página de Histogram Equalization na Wikipedia.

3.2 Filtros por Convolução

Imagens podem ser representadas por funções definidas em domínio discreto. Assim, em processamento digital de imagens, filtros lineares são geralmente realizados por meio da convolução da imagem original f com uma máscara (também chamada de *kernel*) w.

Como já foi dito, a imagem f é uma matriz de $M \times N$ pixeis. A máscara também é uma matriz, que, por questões de desempenho, costuma ser de dimensão pequena $m \times n$. Além disso, por convenção, o centro da matriz é a entrada $(0, 0)$.

Nessas condições, para cada pixel da imagem original, $f(x, y)$, obtemos o pixel correspondente da imagem filtrada, $g(x, y)$, da seguinte forma:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x + s, y + t),$$

Onde:

$a = \frac{m-1}{2}$; $b = \frac{n-1}{2}$; $x = 0, 1, 2, \ldots, M - 1$; $y = 0, 1, 2, \ldots, N - 1$.

Para tratamento das fronteiras, pode-se considerar os valores como zero, ou replicá-los, ou ignorá-los. Aqui, nós podemos sempre ignorá-los.

Comumente, os filtros são realizados por máscaras de dimensão 3×3. Neste caso, a convolução segue como:

$$g(x, y) = \sum_{s=-1}^{1} \sum_{t=-1}^{1} w(s, t) f(x + s, y + t).$$

Intuitivamente, o que se faz é sobrepor o centro da máscara no pixel a ser filtrado e fazer a combinação linear dos valores da máscara com os da imagem. Ou seja, o valor do pixel da imagem após a convolução é a soma da multiplicação dos pixeis sobrepostos.

A imagem 3, cuja fonte é developer.apple.com, pode ajudar a visualizar esse processo:

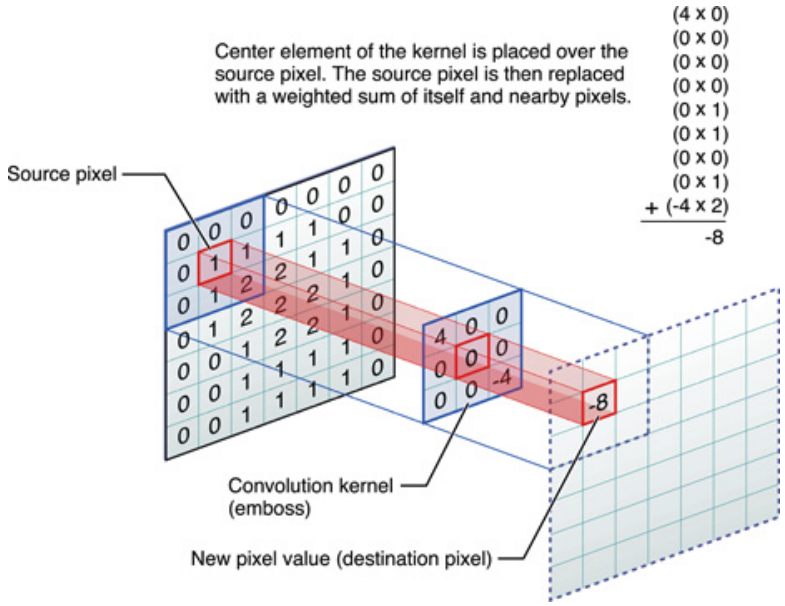

Fig. 3: *Processo de convolução de uma imagem com uma máscara.*

3.3 Filtro de Suavização por Média Ponderada

O objetivo principal desse filtro é reduzir a quantidade de ruídos da imagem.

Há varias formas de suavizar as transições entre os níveis de cinza. Veremos aqui um dos métodos mais simples, que aplica a média ponderada entre os valores do pixel as ser suavizado com os dos pixeis vizinhos. É dado um peso maior para o pixel em questão, depois para os diretamente adjacentes e, por último, os conectados pela diagonal.

Esse método possui a desvantagem de causar o borramento – *blurring* – da imagem.

Sua implementação é feita pixel a pixel através da seguinte equação:

$$g(x,y) = \frac{\sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t)f(x+s,y+t)}{\sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t)}.$$

Que pode ser representada por convolução da imagem com uma máscara. A máscara que usaremos aqui é:

$w =$

	-1	0	1
-1	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$
0	$\frac{2}{16}$	$\frac{4}{16}$	$\frac{2}{16}$
1	$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$

O efeito causado por essa técnica pode ser observado nas imagens em 4.

Fig. 4: *Imagem antes e depois do processo de suavização por média ponderada.*

3.4 Filtro de Aumento de Nitizez – *Sharpening*

O objetivo deste filtro é realçar os detalhes que poderiam ter sido suavizados, por exemplo, pelo efeito "*blurring*".

O operador usado para esse filtro é o Laplaciano:

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

.

Em processamento digital de imagens, deve-se usar esse operador em sua forma discreta, onde teremos a derivada parcial de segunda ordem em relação a x como:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) + 2f(x, y)$$

E a derivada parcial de segunda ordem em relação a y, como:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) + 2f(x, y)$$

A implementação do filtro Laplaciano é feita da seguinte forma:

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y),$$

onde:

c é -1 se o centro da máscara for negativo; e 1, caso contrário.

$\nabla^2 f(x, y) = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) + 4f(x, y)$.

No entanto, podemos utilizar uma convolução da imagem com uma máscara para obtermos o resultando do operador Laplaciano. A máscara que usaremos aqui é a mais comum, conhecida como *8 neighbour Laplacian*. Ela é dada pela seguinte matriz:

$w =$

	-1	0	1
-1	-1	-1	-1
0	-1	8	-1
1	-1	-1	-1

A convolução da imagem com a máscara acima nos dá o Laplaciano da imagem. Podemos notar que o centro dessa máscara é positivo, logo adotaremos na fórmula do filtro $c = 1$, ou seja, para obtermos o efeito de *sharpening*, deveremos adicionar a resultante do Laplaciano na imagem original.

O efeito de realce por filtro Laplaciano pode ser visto nas imagens em 5.

4 Manual

Junto com o arquivo que contém seu código-fonte, submeta uma breve descrição de como compilar e usar o seu programa, inclusive com as especificações já ditas anteriormente.

5 Relatório

Elabore um relatório contendo os seguintes itens:

- Explique suscintamente o processo utilizado na implementação de cada um dos filtros;

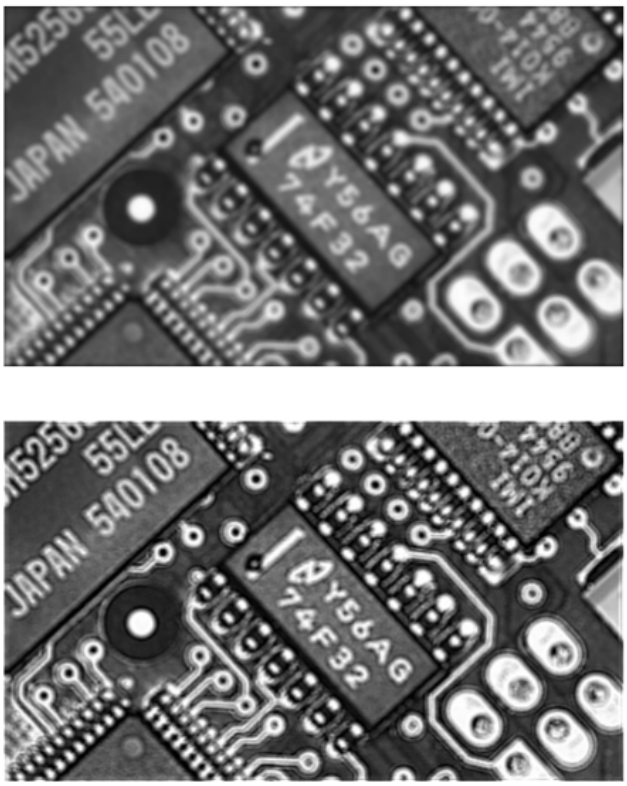

Fig. 5: *Imagem antes e depois do processo de aumento de nitidez usando filtro Laplaciano.*

- Descreva as vantagens e desvantagens de cada um dos métodos, tanto em relação ao desempenho computacional como em relação ao desempenho como filtro de imagem.
- Mostre, para cada um dos métodos solicitados, as imagens originais e as obtidas após o processamento.

  Para o método de ajuste de contraste por equalização de histograma, apresente também os histogramas correspondentes às imagens original e processada.

  No método de aumento de nitidez, apresente também a imagem intermediária resultante da aplicação do operador Laplaciano.

6 Fontes e Referências

- Histogram Equalization na Wikipedia
- Spatial filtering fundamentals, Gleb V. Tcheslavski
- Mathematics in Imaging, Bernhard Haim