MC920: Introdução ao Processamento de Imagem Digital
Tarefa 4

Martin Ichilevici de Oliveira
RA 118077

Rafael Almeida Erthal Hermano
RA 121286

Instituto de Computação, Universidade Estadual de Campinas

7 de março de 2014

1 Convolução

Convolução é uma operação de vizinhança linear, ou seja, ela é uma função que transforma um pixel de acordo com o somatório ponderado dos vizinhos. Portanto, pode ser expresso matematicamente:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \tag{1}$$

Em que $a = \frac{m-1}{2}$ e $b = \frac{n-1}{2}$, onde m e n são o tamanho da janela da vizinhança, e $w(s,t)$ é uma função que determina o peso do vizinho.

1.1 Propriedades

A convolução é uma operação linear já que ela respeita a aditividade e a homogeneidade

1.1.1 Aditividade

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

1.1.2 Homogeneidade

$$\begin{aligned}
g(af) &= \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)af(x-s,y-t) \\
g(af) &= a \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)f(x-s,y-t) \\
g(af) &= ag(f)
\end{aligned}$$

1.2 Aplicações

1.2.1 Blur

O blur é utilizado para eliminar ruidos, removendo pequenos detalhes da imagem. O blur consiste em substituir cada pixel da imagem pela média dos vizinhos eliminando transições bruscas. Um possível problema na utilização do blur é a perda de definição nas bordas dificultando a identificação dos contornos.

Standard Average Blur

Consiste em aplicar para todos os vizinhos o mesmo peso. Da equação 1 temos que $w(s,t) = \frac{1}{n*m} \forall s,t$. Para uma vizinhança 3x3 temos que a máscara é:

$$w(s,t) = \frac{1}{9} \cdot \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Weighted Average Blur

Consiste em aplicar para os vizinhos 4 conexos um peso maior aos vizinhos 8 conexos. Portanto, para uma vizinhança 3x3 temos que a mascara é:

$$w(s,t) = \frac{1}{16} \cdot \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$

1.3 Implementação

Foram usadas duas máscaras, m_1 e m_2:

$$m_1 = \frac{1}{5} \cdot \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} \qquad m_2 = \frac{1}{4} \cdot \begin{vmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

Testou-se inicialmente na matriz M abaixo:

$$M = \cdot \begin{vmatrix} & & \vdots & & \\ & 10 & 10 & 10 & \\ \cdots & 10 & 90 & 10 & \cdots \\ & 10 & 10 & 10 & \\ & & \vdots & & \end{vmatrix}$$

o que produziu (recortando as bordas):

$$M_1 = \frac{1}{5} \cdot \begin{vmatrix} 10 & 26 & 10 \\ 26 & 26 & 26 \\ 10 & 26 & 10 \end{vmatrix} \qquad M_2 = \frac{1}{4} \cdot \begin{vmatrix} 10 & 30 & 10 \\ 10 & 30 & 30 \\ 10 & 30 & 10 \end{vmatrix}$$

Aplicou-se então a convolução com as matrizes acima referidas à Figura 1a. Os resultados estão nas Figuras 1c e 1d.

2 Limiarização

Consiste em escurecer os pixels que possuem nível de cinza abaixo de M e clarear imagens com nível de cinza acima de M. O resultado final desta operação é uma imagem binária.

$$\begin{aligned} T[f(x,y)] &= 0 \text{ se } f(x,y) < M \\ &= 1 \text{ se } f(x,y) > M \end{aligned}$$

(a) Figura original[2]

(b) Convertida para tons de cinza

(c) Após convolução 1

(d) Após convolução 2

Figura 1: Imagem original e em tons de cinza antes e após convolução

Referências

[1] GONZALEZ, Rafael C.; WOODS, Richard E.. **Digital Image Processing**. 3. ed. Upper Saddle River, NJ, EUA: Prentice-hall, 2006.

[2] `http://robotix.in/tutorials/category/opencv/noise_reduction`