



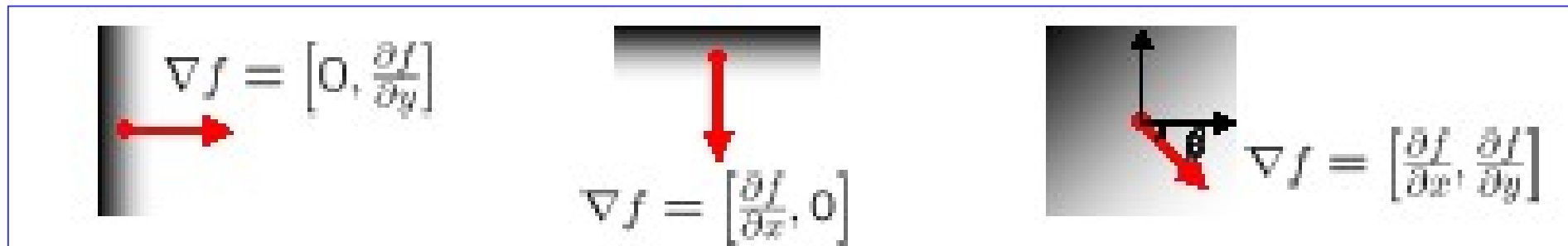
# Aula 5.4

## Realce no Espaço



## Gradiente da imagem

$$\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$$



A direção do gradiente é dada por:  $\theta = \tan^{-1} \left( \frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} \right)$

A intensidade da borda é dada pela magnitude do gradiente:

$$\|\nabla f\| = \sqrt{\left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2}$$



## Sobel

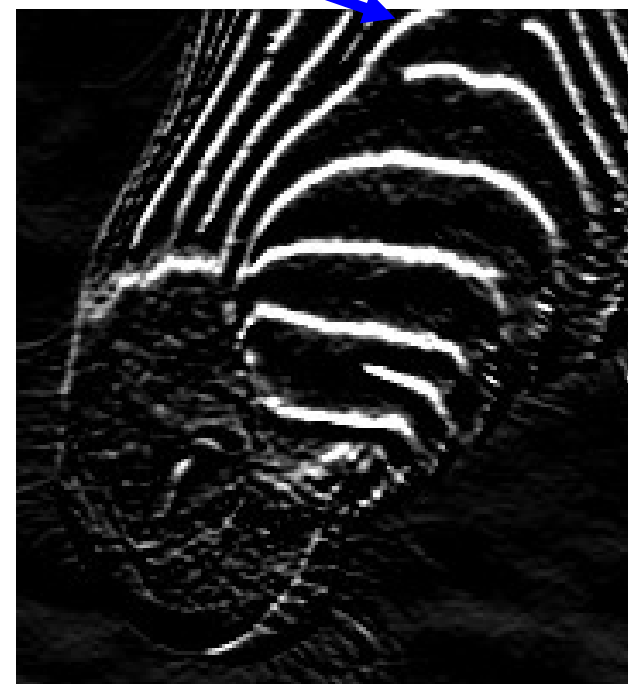
$$\frac{\partial f}{\partial x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial f}{\partial y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{\partial f}{\partial x}$$

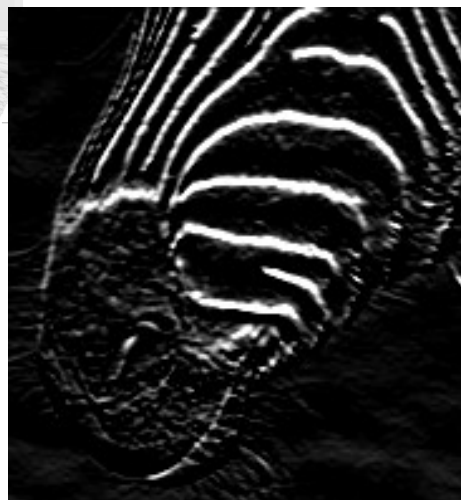


$$\frac{\partial f}{\partial y}$$

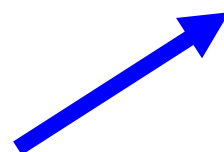
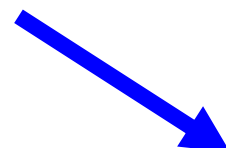




$$\text{magnitude} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

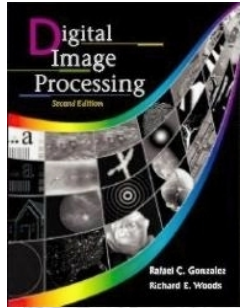


Sobel ( vertical e horizontal)

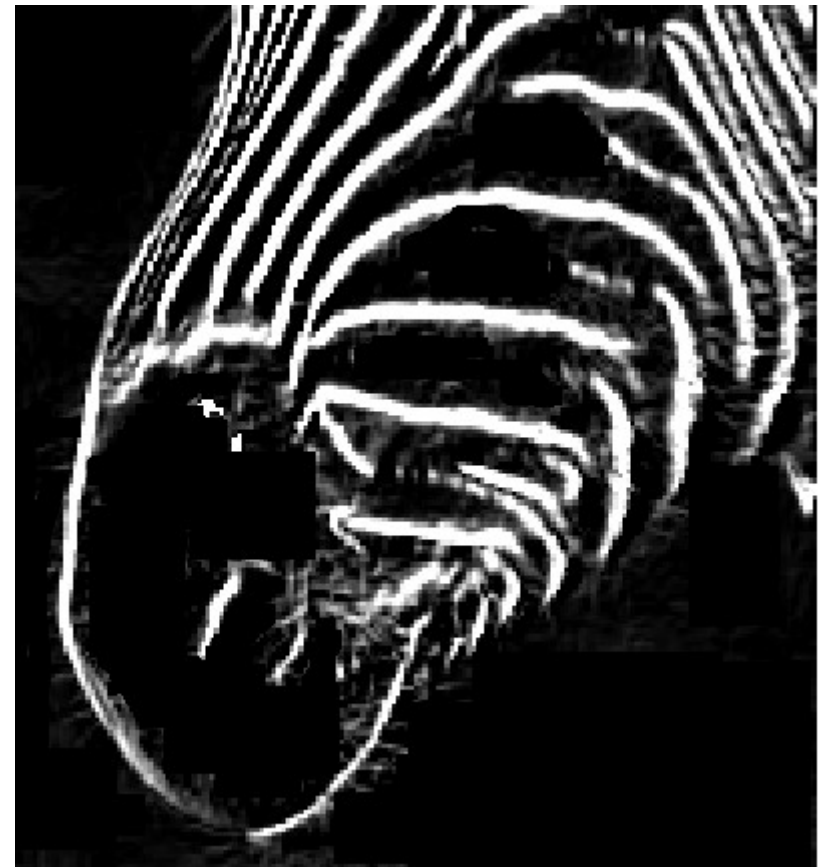


**magnitude**

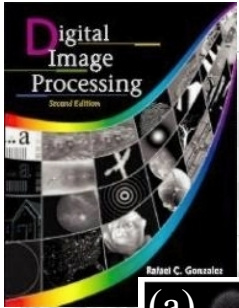




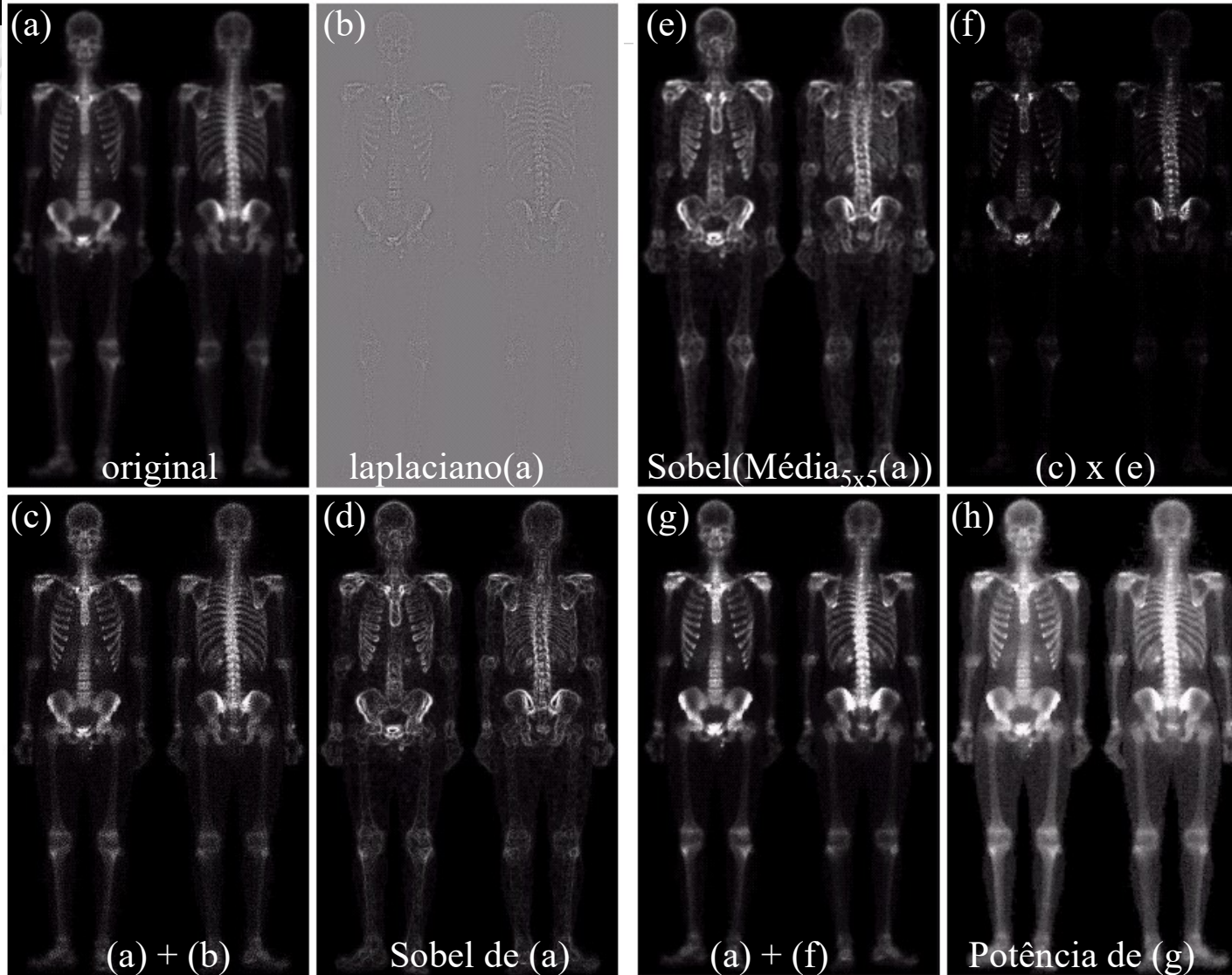
$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) < \text{limiar} \\ f(x, y) & \text{se } f(x, y) \geq \text{limiar} \end{cases}$$







## Combinação de Técnicas no Domínio do Espaço



$$S = c (r + \varepsilon)^\gamma$$



## Normalização de Valores de Saída

Para a exibição de uma imagem resultado de algum processamento, é preciso considerar a faixa de valores obtida no processamento e a faixa de valores do dispositivo de apresentação (tela do computador, que é de 0 até 255)

No caso do Laplaciano, foi usada a função  $f(x) = |x| / 8$

Para apresentar a magnitude do gradiente  $\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$

deve-se prever a faixa de valores obtidos nesta operação



## Normalização de Valores de Saída

Considerando-se os filtros

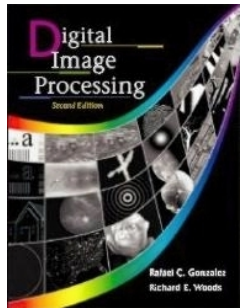
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

$$\frac{\partial f}{\partial x} = 1/4 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{e} \quad \frac{\partial f}{\partial y} = 1/4 \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

será obtido um valor extremo igual a -255 para o primeiro quando a imagem for

255	0	0	
255	0	0	
255	0	0	
...			





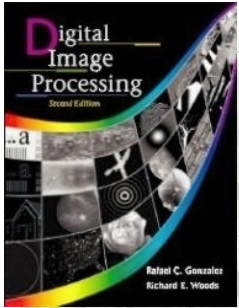
## Normalização de Valores de Saída

e será obtido um valor extremo igual a 255 quando a imagem for

$$\frac{\partial f}{\partial x} = 1/4 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

0	0	255	
0	0	255	
0	0	255	
...			

para o segundo filtro, ocorre o mesmo, na vertical



## Normalização de Valores de Saída

Aplicando o e será obtido um valor extremo igual a 255 quando a imagem for

$$\frac{\partial f}{\partial x} = 1/4 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

0	0	255	
0	0	255	
0	0	255	
...			

para o segundo filtro, ocorre o mesmo, na vertical



## Normalização de Valores de Saída

Considerando que estes seriam os valores máximos para  $g_x$  e  $g_y$ , quando se aplica a equação para obter a magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

teríamos valores na faixa de 0 até 360, assim, para ficar na faixa de 0 até 255, bastaria dividir o resultado por 1.4142

Entretanto, quando se obtém um valor máximo para  $G_x$ , se obtém zero para  $G_y$  e vice-versa, logo, o valor máximo não é 360



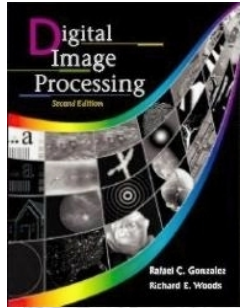
## Normalização de Valores de Saída

É possível adotar 360 como o valor máximo nesta operação, e visualizar o resultado, porém, com uma imprecisão.

O mais correto seria calcular todos os valores de magnitude. Em seguida, obter os valores mínimo e máximo ( $M_{\min}$  e  $M_{\max}$ ) e aplicar a normalização

$$normalizado = \frac{M - M_{\min}}{M_{\max} - M_{\min}} . 255$$

Deste modo, se garante que os valores de magnitudes ficarão totalmente distribuídos na faixa de 0 até 255



# Exercícios

1) Aplicar o filtro ao lado por toda a imagem e, em seguida, repetir este processo novamente, equivale a aplicar qual filtro uma única vez ?

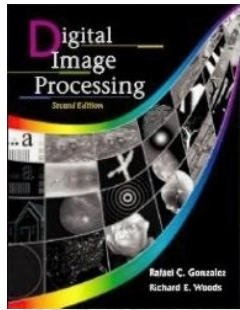
1	1	1
---	---	---

2) implementar a detecção de bordas por Sobel

- Exibir bordas na imagem de saída (magnitudes)
- Quando se passar o mouse sobre a imagem de saída, apresentar as informações de magnitude e direção em caixas de texto

(usar a Figura [aula05-a.bmp](#))

**usar a normalização  
do slide anterior**



3) Implemente a compressão de escala dinâmica  $S = cr^\gamma$  com  $c$  e  $\gamma$  (gama) fornecidos pelo usuário. Testar com os valores do slide seguinte, usado na aula 5.1 (usar a Figura [aula05-b.bmp](#))

4) Aplicar a Limiarização (usuário define  $T$ )





## Realce de Imagens

Imagem original

$r = \text{tom da imagem}$



$$S = cr^\gamma$$
$$c = 1.0$$
$$\gamma = 0.6$$

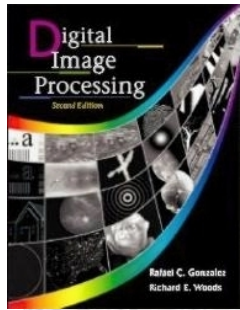


$$S = cr^\gamma$$
$$c = 1.0$$
$$\gamma = 0.4$$

$$S = cr^\gamma$$
$$c = 1.0$$
$$\gamma = 0.3$$



```
//--- Detecção de bordas utilizando o operador de Sobel -----  
void __fastcall TForm1::GxeGy1Click(TObject *Sender)  
{  
    double GX, GY;  
    int x, y, z, i, j;  
    double mag;  
    inicia_processamento();  
    for (i=1; i<larg-1; i++)  
        for (j=1; j<altu-1; j++)  
        {  
            GX = round(( - ImE1[i-1][j-1] - 2*ImE1[i][j-1] - ImE1[i+1][j-1]  
                        + ImE1[i-1][j+1] + 2*ImE1[i][j+1] + ImE1[i+1][j+1] ) / 4.0);  
  
            GY = round(( - ImE1[i-1][j-1] + ImE1[i+1][j-1]  
                        - 2*ImE1[i-1][j] + 2 * ImE1[i+1][j]  
                        - ImE1[i-1][j+1] + ImE1[i+1][j+1]) / 4.0);  
  
            mag = sqrt(GX*GX + GY*GY);  
            ImS[i][j] = round(mag * 2.0 / 3.0);  
  
            if (GX == 0) x=1; else x=0;  
            if (GY == 0) y=1; else y=0;
```



```

z = y * 2 + x;
switch (z)
{
    case 0 : { dir[i][j]=(int)(atan2(GY,GX)*180/pi);
               if (dir[i][j] < 0) dir[i][j] = dir[i][j] + 360;
               break;
            }
    case 1 : if (GY>0) dir[i][j]=90; else dir[i][j]=270; break;
    case 2 : if (GX>0) dir[i][j]=0; else dir[i][j]=180; break;
    default : dir[i][j]=-1;
}
}

```

```

direcoes_calculadas = true; // direções calculadas ok.
i=0; for (j=0; j<altu; j++) { ImS[i][j] = 0; dir[i][j] = -1; }
i=1; for (j=0; j<altu; j++) { ImS[i][j] = 0; dir[i][j] = -1; }
i=larg-2; for (j=0; j<altu; j++) { ImS[i][j] = 0; dir[i][j] = -1; }
i=larg-1; for (j=0; j<altu; j++) { ImS[i][j] = 0; dir[i][j] = -1; }
j=0; for (i=0; i<larg; i++) { ImS[i][j] = 0; dir[i][j] = -1; }
j=1; for (i=0; i<larg; i++) { ImS[i][j] = 0; dir[i][j] = -1; }
j=altu-2; for (i=0; i<larg; i++) { ImS[i][j] = 0; dir[i][j] = -1; }
j=altu-1; for (i=0; i<larg; i++) { ImS[i][j] = 0; dir[i][j] = -1; }
termina_processamento();
}

```