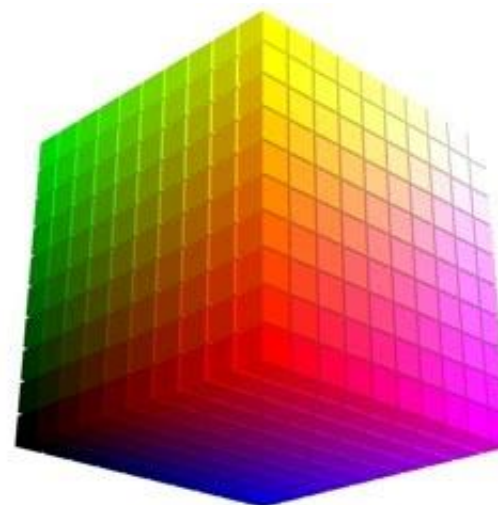




# PROCESSAMENTO DIGITAL DE IMAGENS



Projeto de Ensino - Material didático sobre filtros de imagens

Departamento Engenharias e Computação- DEC

Discente - Luciana Roncarati - Ciência da Computação

# SUMÁRIO

- Definição borda.
- Definição Filtro Prewitt.
- Algoritmo Interface *Processing*.
- Referências Bibliográficas.

# DEFINIÇÃO

**Detecção de bordas:** Partindo da definição de borda como uma fronteira entre duas regiões com níveis de cinza relativamente distintos, os algoritmos utilizados para a detecção de bordas são estruturados de forma a detectar as descontinuidades existentes nas transições.

# DEFINIÇÃO

**Detecção de bordas:** A detecção de bordas é uma técnica fundamental no processamento de imagens que visa identificar as transições abruptas de intensidade nos pixels da imagem. Essas transições representam mudanças significativas nas propriedades visuais da imagem, como mudanças de cor, luminosidade ou textura, e podem indicar a presença de objetos, contornos ou padrões importantes.

# DEFINIÇÃO

O operador realiza uma medição de gradiente espacial 2D em uma imagem. Ele destaca regiões de alta frequência espacial que frequentemente correspondem a bordas. Em seu uso mais comum, a entrada para o operador é uma imagem em escala de cinza, assim como a saída. Os valores de pixel em cada ponto da saída representam a magnitude absoluta estimada do gradiente espacial da imagem de entrada naquele ponto.

# DEFINIÇÃO

O operador gradiente é um dos procedimentos utilizados para detectar essas descontinuidades denominadas como bordas

$$\text{Magnitude do Gradiente} = \text{sqrt}(G_x^2 + G_y^2)$$

# DEFINIÇÃO

$$I = \begin{pmatrix} a_{x-1 \ y-1} & a_{x-1 \ y} & a_{x-1 \ y+1} \\ a_{x \ y-1} & a_{x \ y} & a_{x \ y+1} \\ a_{x+1 \ y-1} & a_{x+1 \ y} & a_{x+1 \ y+1} \end{pmatrix}$$

Considerando-se uma vizinhança de 3 x 3 pixels em torno de um ponto (x,y).

$$I = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

# DEFINIÇÃO

Para realizar a detecção de bordas com o operador Prewitt, primeiro convolvemos a imagem original com os seguintes dois kernels:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \times A$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \times A$$



# DEFINIÇÃO

$$\begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix}$$

Kx:

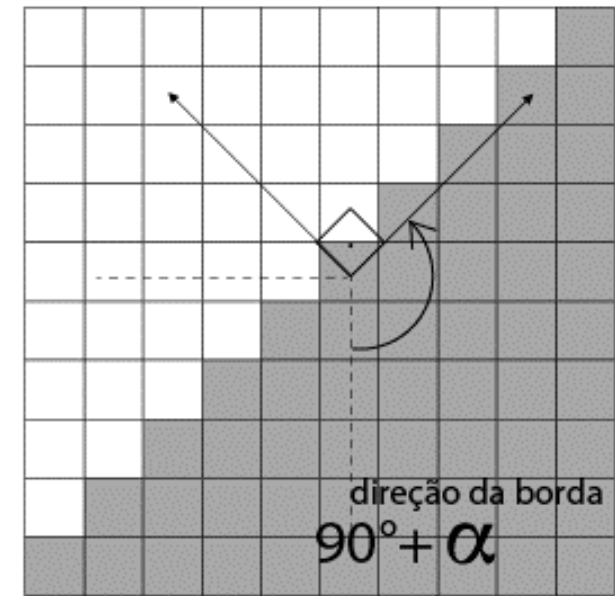
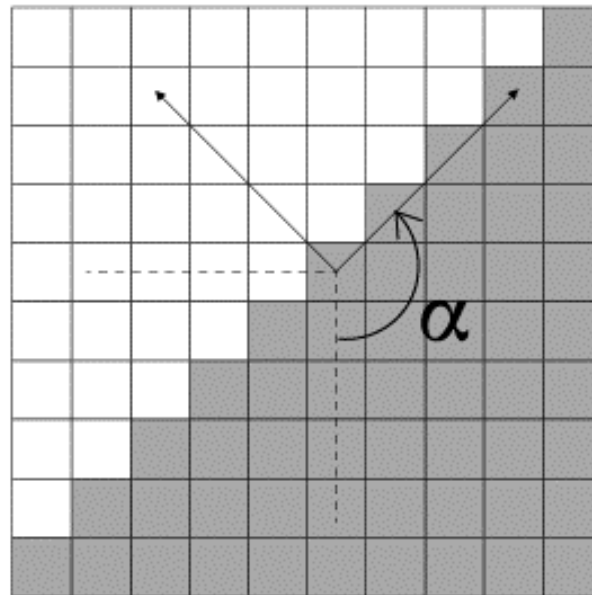
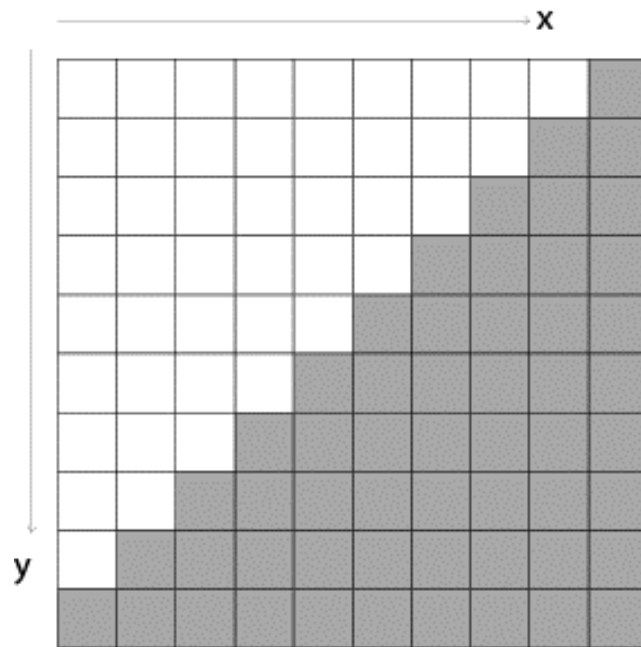


Fig. 1 – definição bordas

# DEFINIÇÃO

$$\begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$$

Ky:

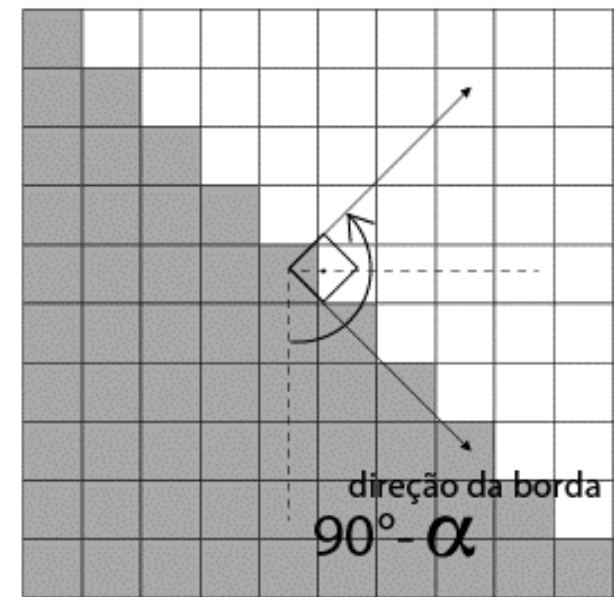
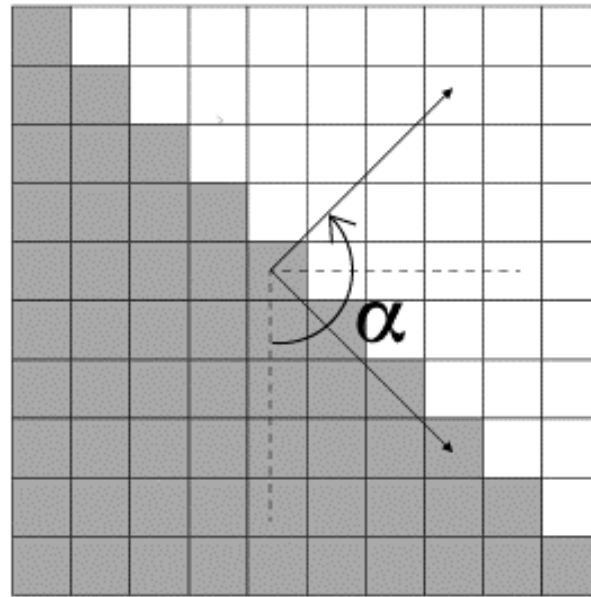
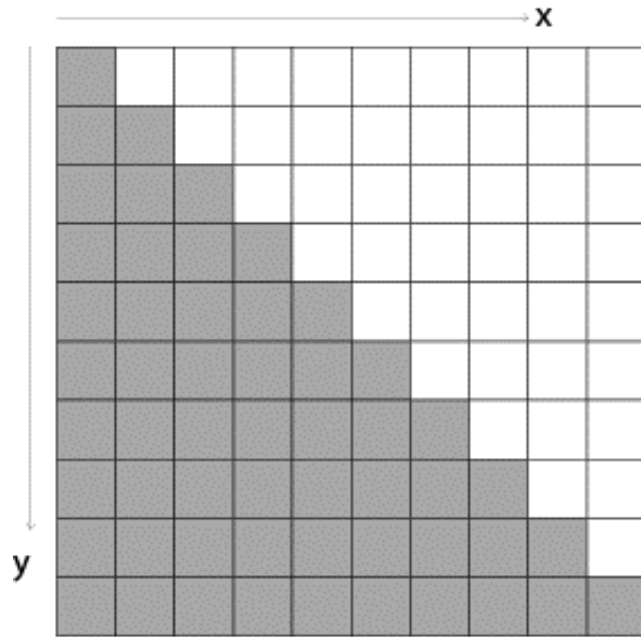


Fig. 2 – definição bordas

# DEFINIÇÃO

Se a magnitude calculada é maior do que o menor valor de entrada (definido de acordo com a natureza e qualidade da imagem que esta sendo processada), o pixel é considerado ser parte de um borda. A direção do gradiente da borda, perpendicular a direção da borda, é encontrada com a seguinte fórmula:

$$\alpha = \text{atan} \frac{G_x}{G_y}$$

# ALGORITMO DETECÇÃO DE BORDAS

```
1 PImage img;
2 PImage bordasPrewitt;
3
4 void setup() {
5     size(640, 480);
6     img = loadImage("Bikesgray.jpg"); // Carregue sua própria imagem aqui
7     img.resize(width, height); // Redimensione a imagem para o tamanho da janela
8     image(img, 0, 0);
9     loadPixels();
10
11     // Crie duas máscaras de Prewitt para detecção de bordas
12     float[][] PrewittlX = {
13         {1, 0, -1},
14         {1, 0, -1},
15         {1, 0, -1}
16     };
17
18     float[][] PrewittlY = {
19         {1, 1, 1},
20         {0, 0, 0},
21         {-1, -1, -1}
22     };
23
24     // Criar uma nova imagem para armazenar as bordas detectadas
25     bordasPrewitt = createImage(width, height, RGB); // Removido 'PImage' daqui
26 }
```

Fig. 3 – interface Processing

# ALGORITMO DETECÇÃO DE BORDAS

```
26
27 // Aplica o filtro de Prewitt
28 for (int y = 1; y < height - 1; y++) {
29     for (int x = 1; x < width - 1; x++) {
30         float sumX = 0;
31         float sumY = 0;
32
33         // Aplique as máscaras de Prewitt para cada pixel
34         for (int i = -1; i <= 1; i++) {
35             for (int j = -1; j <= 1; j++) {
36                 color c = img.get(x + i, y + j);
37                 float gray = brightness(c);
38                 sumX += PrewittlX[i + 1][j + 1] * gray;
39                 sumY += PrewittlY[i + 1][j + 1] * gray;
40             }
41         }
42
43         // Combine os gradientes x e y
44         float gradient = sqrt(sumX * sumX + sumY * sumY);
45         int c = color(gradient);
46         bordasPrewitt.pixels[y * width + x] = c; // Alterado 'pixels' para 'bordasPrewitt.pixels'
47     }
48 }
49 bordasPrewitt.updatePixels(); // Adicionado para garantir que os pixels sejam atualizados
50 bordasPrewitt.save("imagem_bordas_Prewitt.jpg");
51 }
```

Fig. 4 – interface Processing



Fig. 5 – imagem de entrada



Fig. 6 – imagem de saída

# REFERÊNCIAS BIBLIOGRÁFICAS

- NUNES L. S, Fátima - Introdução ao processamento de imagens médicas para auxílio ao diagnóstico - uma visão prática, capítulo 2.
- GONZALEZ C, Rafael. e WOODS, Richard - Processamento digital de imagens - 3. Ed. Pearson Prentice hall, São paulo, 2010.
- <https://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm>