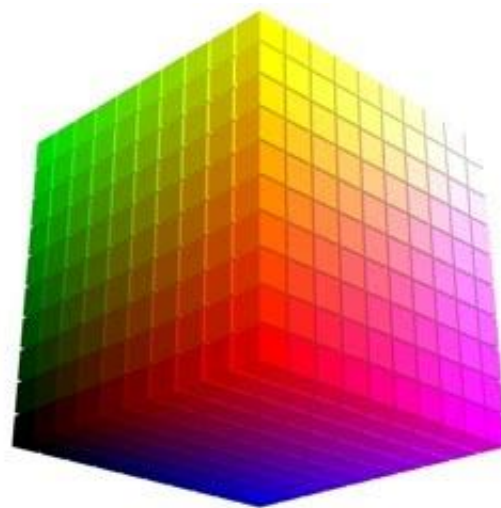




PROCESSAMENTO DIGITAL DE IMAGENS



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 Robert Cross.
- 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 **Roberts Cross** é usado em processamento de imagem e visão computacional para detecção de bordas. Foi um dos primeiros detectores de borda propostos por Lawrence Roberts em 1963. Como um operador diferencial, a ideia por trás do operador Roberts Cross é aproximar o gradiente de uma imagem através de diferenciação discreta, o que é alcançado calculando a soma dos quadrados das diferenças entre pixels diagonalmente adjacentes.

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 Roberts, primeiro convolvemos a imagem original com os seguintes dois kernels:

K_x:

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

K_y:

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

DEFINIÇÃO

Kx:

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

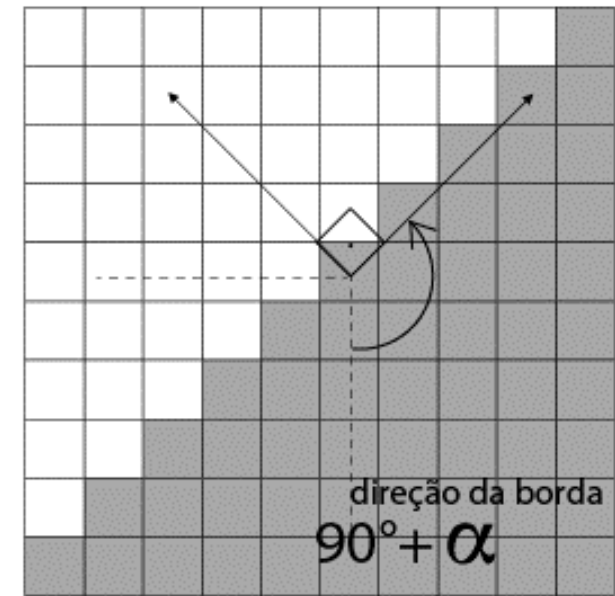
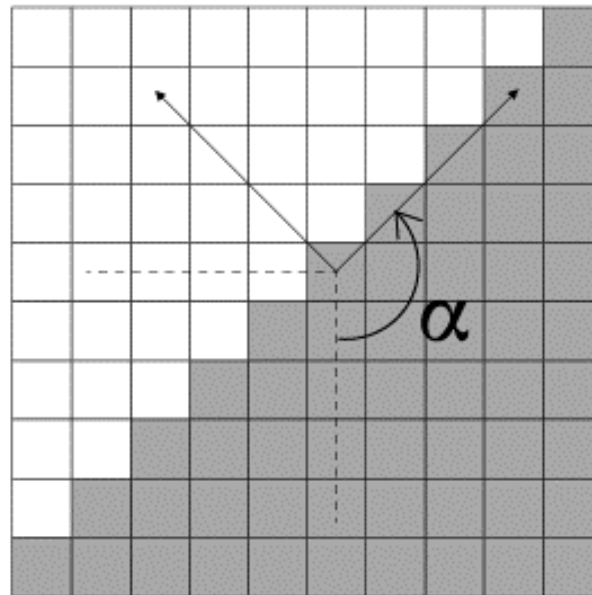
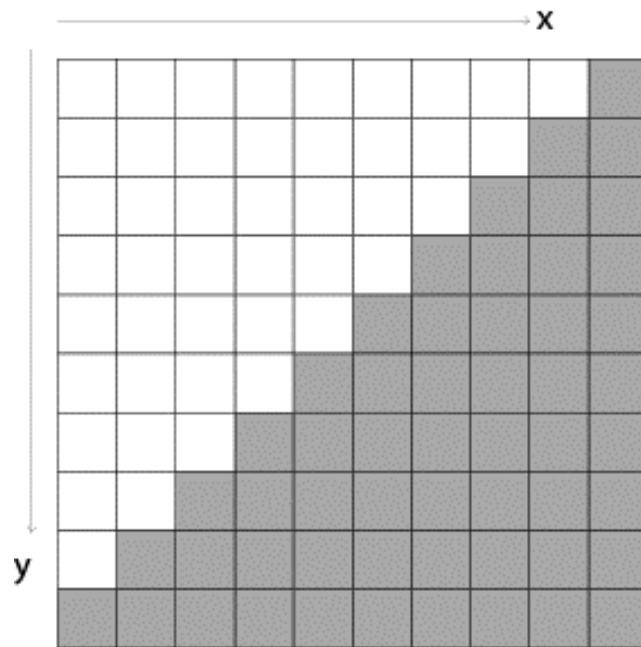


Fig. 1 – definição bordas

DEFINIÇÃO

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

Ky:

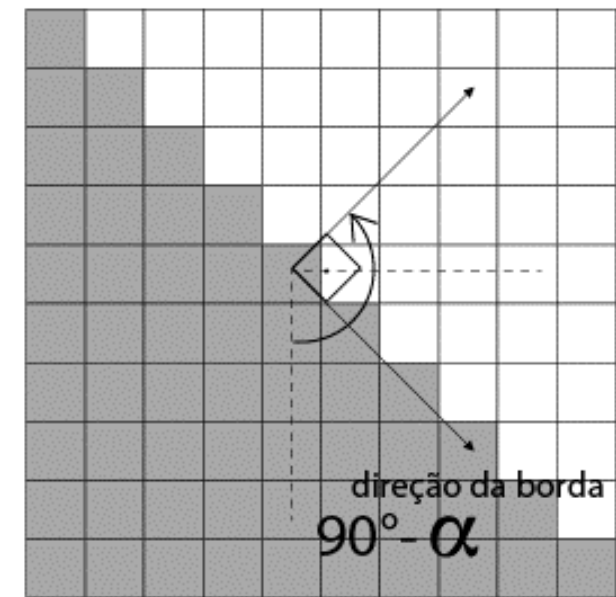
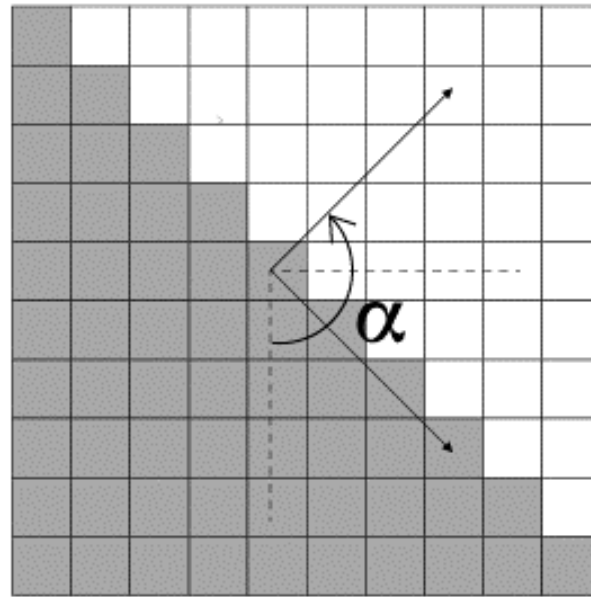
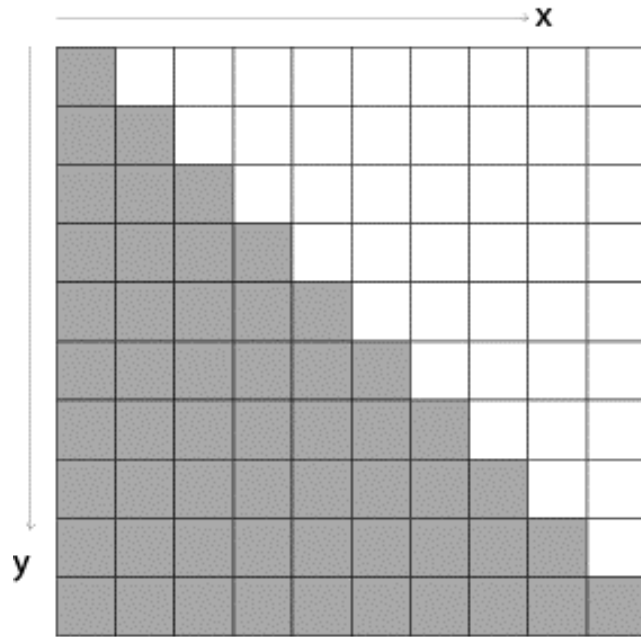


Fig. 2 – definição bordas

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
3 void setup() {
4   size(640, 480);
5   img = loadImage("Bikesgray.jpg"); // Carregue sua própria imagem aqui
6   img.resize(width, height); // Redimensione a imagem para o tamanho da janela
7   image(img, 0, 0);
8   loadPixels();
9
10  // Máscaras de Robert Cross para detecção de bordas
11  float[][] robertsX = {
12    {1, 0},
13    {0, -1}
14  };
15
16  float[][] robertsY = {
17    {0, 1},
18    {-1, 0}
19  };
20
21  // Criar uma nova imagem para armazenar as bordas detectadas
22  PImage bordas = createImage(width, height, RGB);
23
24  // Aplicando o filtro de Robert Cross
25  for (int y = 0; y < height - 1; y++) {
26    for (int x = 0; x < width - 1; x++) {
```

Fig. 3 – interface Processing

ALGORITMO DETECÇÃO DE BORDAS

```
27     float sumX = 0;
28     float sumY = 0;
29
30     // Aplicando as máscaras de Robert Cross para cada pixel
31     for (int i = 0; i <= 1; i++) {
32         for (int j = 0; j <= 1; j++) {
33             color c = img.get(x + i, y + j);
34             float gray = brightness(c);
35             sumX += robertsX[i][j] * gray;
36             sumY += robertsY[i][j] * gray;
37         }
38     }
39     // Calculando o gradiente
40     float gradient = sqrt(sumX * sumX + sumY * sumY);
41     int c = color(gray);
42     bordas.pixels[y * width + x] = c;
43 }
44 }
45 bordas.updatePixels();
46 image(bordas, 0, 0); // Exibe a imagem com as bordas detectadas
47
48 // Salva a nova imagem com as bordas detectadas
49 bordas.save("imagem_borda_Cross.jpg");
50 }
```

Fig. 4 – interface Processing



Fig. 5 – imagem de entrada

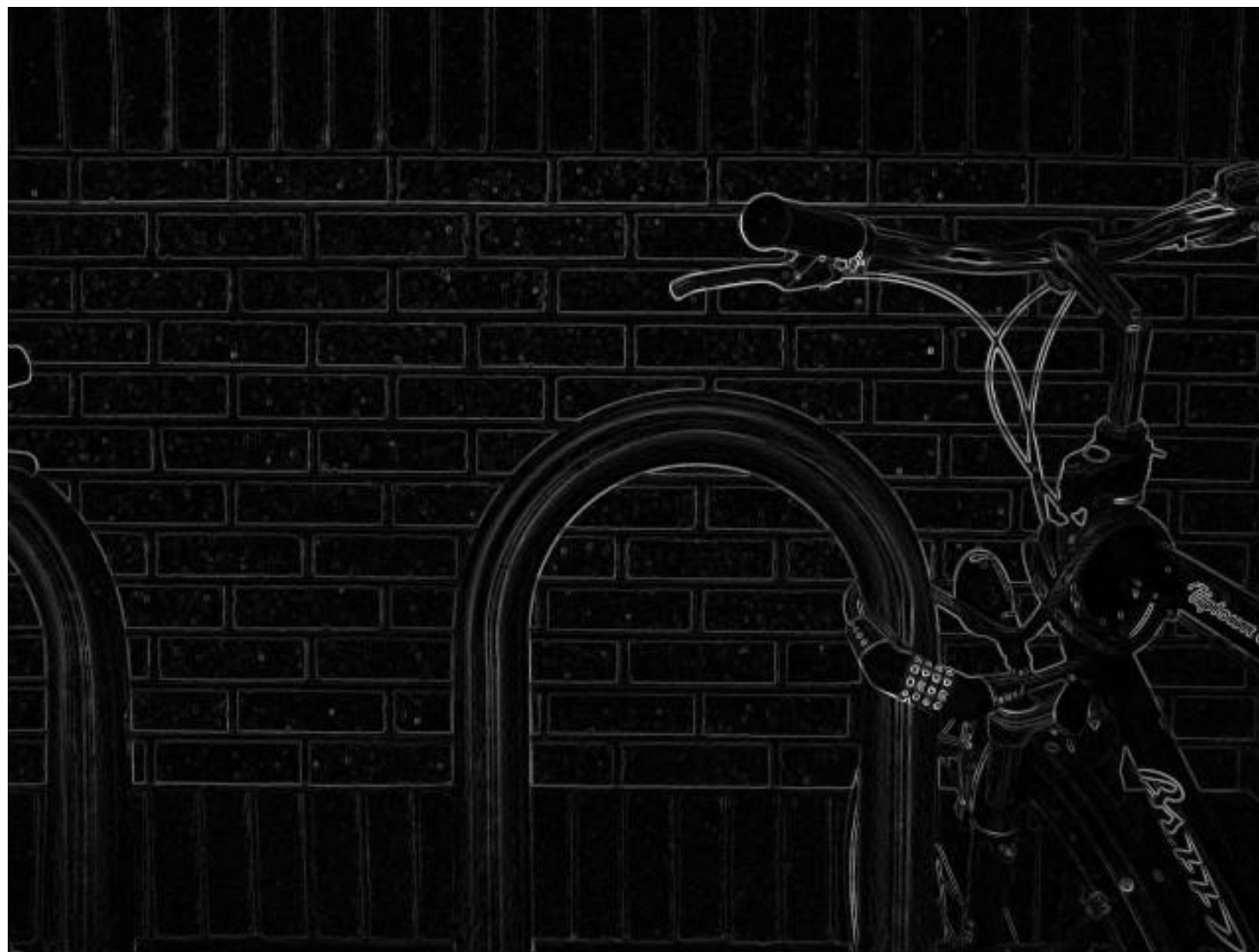


Fig. 6 – imagem de saída

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>