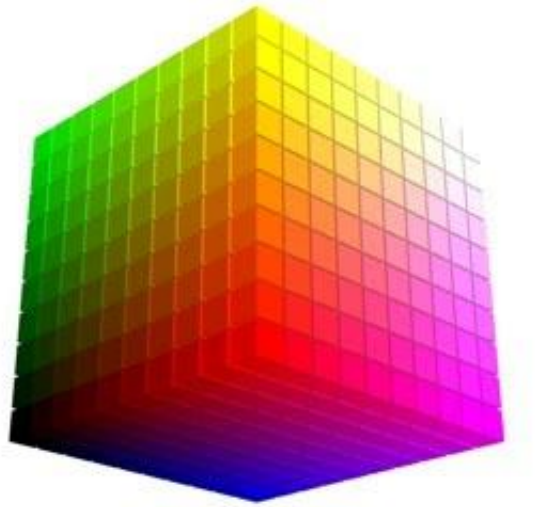
PROCESSAMENTO DIGITAL DE IMAGENS



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 Deriche.
- 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

$$\text{Magnitude do Gradiente} = \sqrt{(Gx^2 + Gy^2)}$$

DEFINIÇÃO

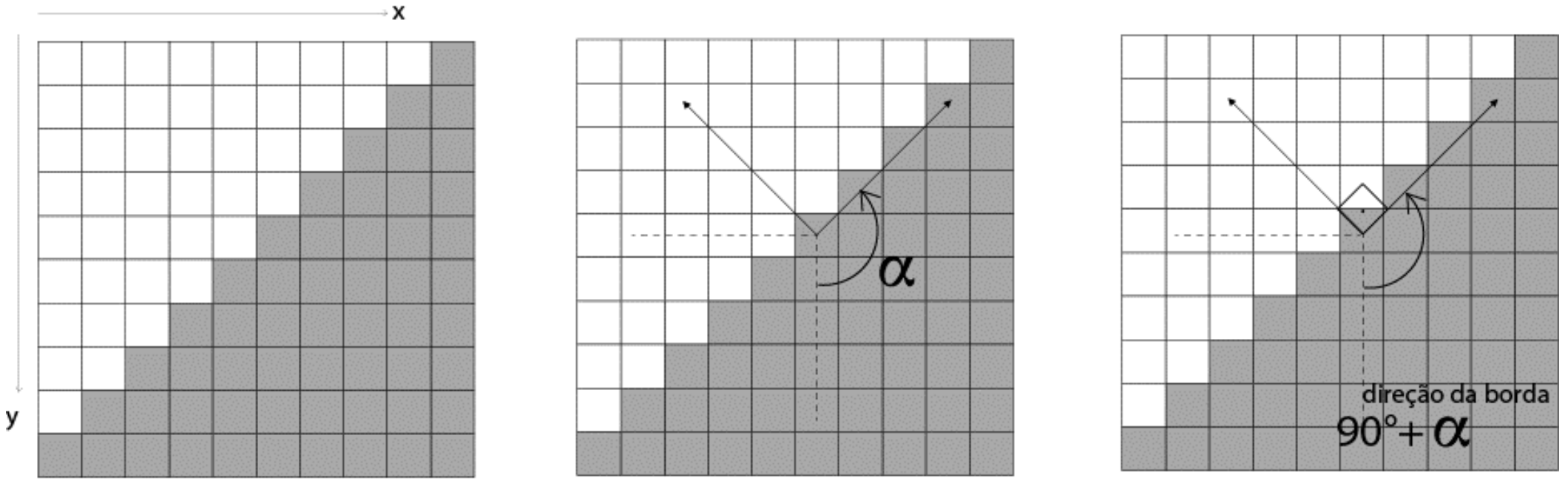


Fig. 1 – definição bordas

DEFINIÇÃO

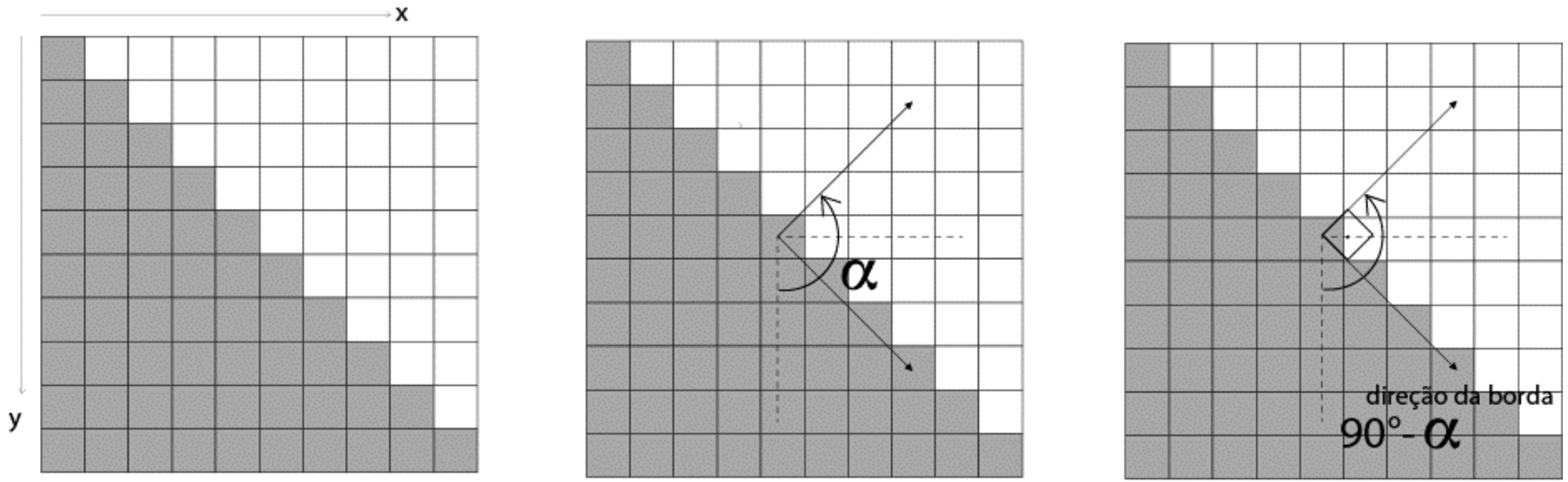


Fig. 2 – definição bordas

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:

$$\alpha = \operatorname{atan} \frac{Gx}{Gy}$$

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

Detecção de bordas: O detector de bordas Deriche é um operador de detecção de bordas desenvolvido por Rachid Deriche em 1987. É um algoritmo de várias etapas usado para obter um resultado ideal de detecção de bordas em uma imagem bidimensional discreta. Este algoritmo é baseado no trabalho de John F. Canny relacionado à detecção de bordas (canny's edge detector) e seus critérios para detecção ideal de bordas:

DEFINIÇÃO

O detector de borda Deriche, como o detector de borda Canny , consiste nas 4 etapas a seguir:

- Suavização: A imagem é suavizada para reduzir o ruído.
- Cálculo do gradiente: A magnitude e a direção do gradiente são calculadas para identificar áreas de mudança de intensidade.

DEFINIÇÃO

- Supressão não máxima: Os pixels são refinados para manter apenas os de maior magnitude ao longo das direções do gradiente.
- Limite de histerese: Pixels são classificados como pertencentes a bordas fortes, fracas ou não bordas, usando limiares superior e inferior, e apenas bordas fortes conectadas são mantidas.

DEFINIÇÃO

A diferença essencial está na implementação das duas primeiras etapas do algoritmo. Ao contrário do detector de bordas Canny, o detector de bordas Deriche usa o filtro Resposta de impulso infinita.

DEFINIÇÃO

Qualidade de detecção – todas as bordas existentes devem ser marcadas e nenhuma detecção falsa deve ocorrer.

Precisão - as bordas marcadas devem estar o mais próximo possível das bordas da imagem real.

Não ambigüidade – uma determinada borda da imagem só deve ser marcada uma vez. Não devem ocorrer respostas múltiplas para uma borda da imagem real.

Por esta razão, este algoritmo é frequentemente referido como detector Canny-Deriche.

ALGORITMO DETECÇÃO DE BORDAS

```
1  PImage img;
2  void setup() {
3    size(400, 530);
4    img = loadImage("Sunflowers_in_July.jpg"); // Carrega a imagem
5    img.resize(width, height); // Redimensiona a imagem para o tamanho do canvas
6    image(img, 0, 0); // Exibe a imagem original
7    loadPixels(); // Carrega os pixels da imagem
8
9    PImage grayImg = createImage(width, height, RGB); // Cria uma nova imagem em tons de cinza
10   // Transforma a imagem em tons de cinza
11   for (int x = 0; x < width; x++) {
12     for (int y = 0; y < height; y++) {
13       int loc = x + y * width;
14       color c = img.pixels[loc];
15       float grayValue = (red(c) + green(c) + blue(c)) / 3.0; // Calcula a média dos valores de RGB
16       grayImg.pixels[loc] = color(grayValue, grayValue, grayValue);
17     }
18   }
19
20   grayImg.updatePixels(); // Atualiza os pixels da imagem em tons de cinza
21
22   // Aplica o filtro Deriche para detecção de bordas na imagem em tons de cinza
23   float[][] deriche = {
24     {-1, -1, -1},
25     {-1,  8, -1},
26     {-1, -1, -1}
27   };
28
```

Fig. 3 – interface Processing

ALGORITMO DETECÇÃO DE BORDAS

```
28
29  PImage newImg = createImage(width, height, RGB); // Cria uma nova imagem para armazenar o resultado
30
31  for (int x = 1; x < width - 1; x++) {
32    for (int y = 1; y < height - 1; y++) {
33      float sumR = 0;
34      float sumG = 0;
35      float sumB = 0;
36      int offset = (y * width + x);
37
38      for (int i = -1; i <= 1; i++) {
39        for (int j = -1; j <= 1; j++) {
40          int grayPix = grayImg.pixels[offset + j + i * width];
41          float factor = deriche[j + 1][i + 1];
42          sumR += red(grayPix) * factor;
43          sumG += green(grayPix) * factor;
44          sumB += blue(grayPix) * factor;
45        }
46      }
47
48      sumR = constrain(sumR, 0, 255);
49      sumG = constrain(sumG, 0, 255);
50      sumB = constrain(sumB, 0, 255);
51
52      newImg.pixels[offset] = color(sumR, sumG, sumB); // Armazena o pixel na nova imagem
53    }
54  }
55
```

Fig. 4 – interface Processing

ALGORITMO DETECÇÃO DE BORDAS

```
56
57    newImg.updatePixels(); // Atualiza os pixels da nova imagem
58    newImg.save("Sunflowers_in_July_DERICHE.jpg"); // Salva a nova imagem
59  }
60
```

Fig. 5 – interface Processing

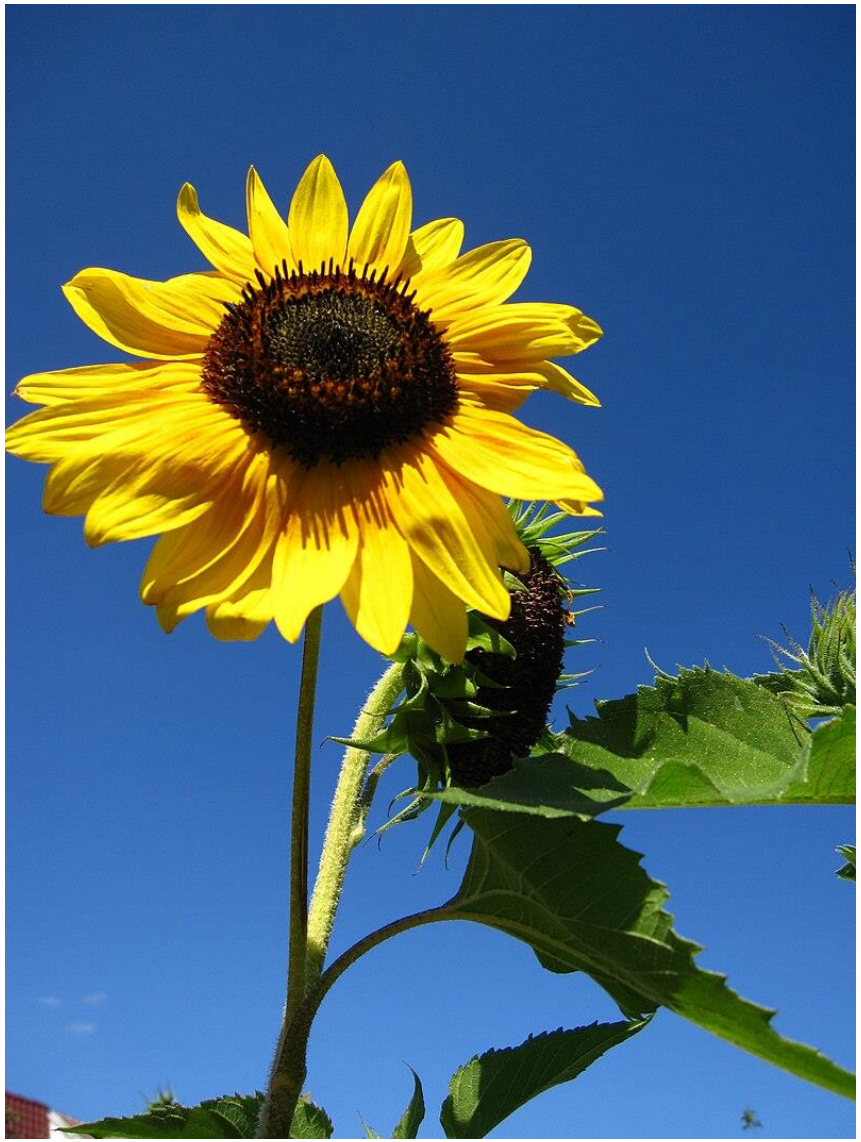


Fig. 6 – imagem de entrada

Fig. 7– 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