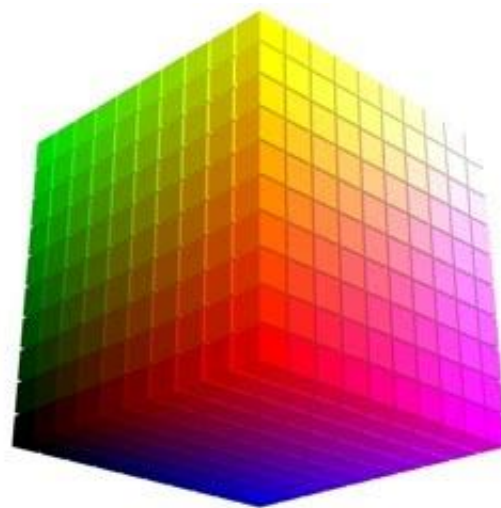




PROCESSAMENTO DIGITAL DE IMAGENS



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 Canny.
- 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

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

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

A detecção de bordas compreende uma série de métodos matemáticos projetados para identificar as bordas em uma imagem digital, definidas como as regiões onde ocorre uma mudança abrupta no brilho da imagem ou, de maneira mais formal, onde ocorrem descontinuidades. A detecção de bordas desempenha um papel essencial no campo do processamento de imagens, visão computacional e visão de máquina, especialmente no contexto da detecção e extração de características.

DEFINIÇÃO

Canny (1986) propôs um método para detecção de bordas que visa otimizar a localização dos pontos de borda na presença de ruído. Inicialmente, a imagem é suavizada usando um filtro Gaussiano. Em seguida, a magnitude e a direção do gradiente são calculadas utilizando aproximações baseadas em diferenças finitas para as derivadas parciais, de maneira similar aos métodos de gradiente apresentados anteriormente.

DEFINIÇÃO

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

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

DEFINIÇÃO

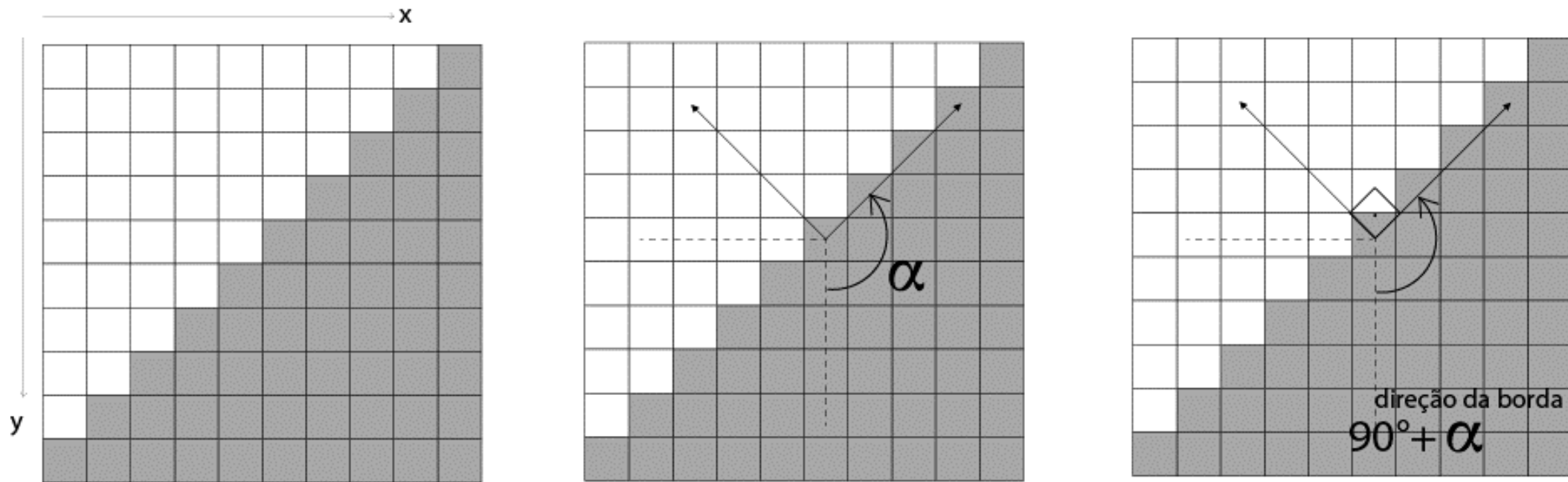


Fig. 1 – definição bordas

DEFINIÇÃO

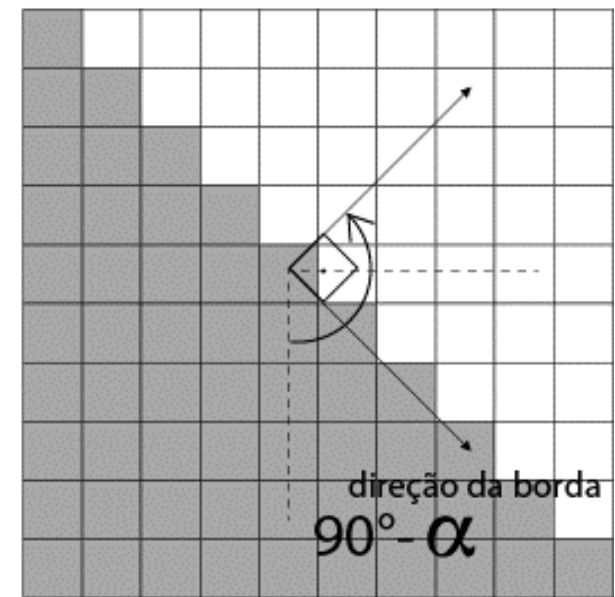
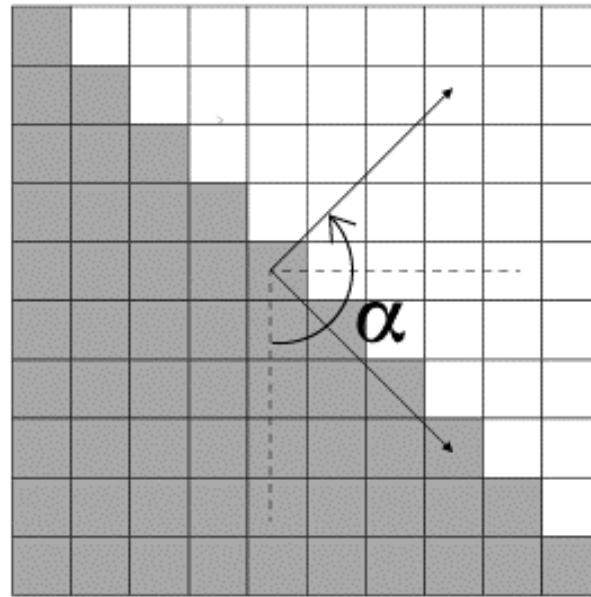
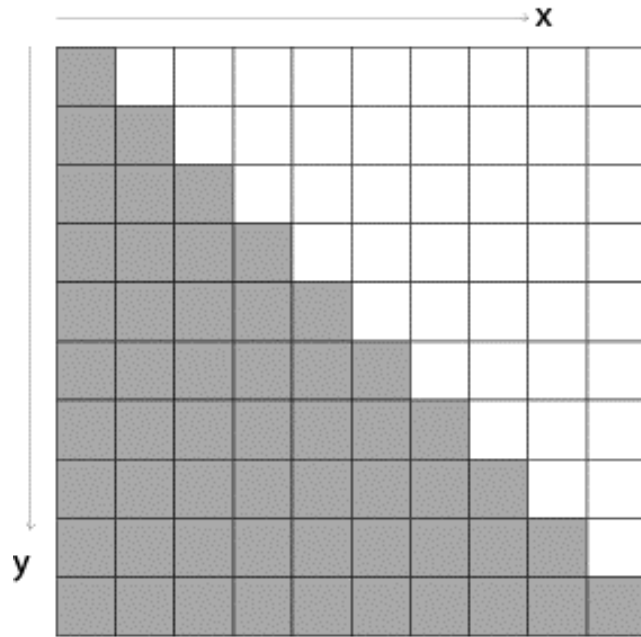


Fig. 2 – definição bordas

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 uma borda. A direção do gradiente da borda, perpendicular à direção da borda, é encontrada com a seguinte fórmula:

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

DEFINIÇÃO

Após o cálculo do gradiente, a borda é localizada selecionando apenas os pontos cuja magnitude é localmente máxima na direção do gradiente. Essa operação, chamada de supressão não-máxima, reduz a espessura das bordas. No entanto, a borda ainda pode conter certos fragmentos espúrios causados pela presença de ruído ou textura fina.

DEFINIÇÃO

Depois de calculada uma medida da resistência da borda (normalmente a magnitude do gradiente), o próximo estágio é aplicar um limite, para decidir se as bordas estão presentes ou não em um ponto da imagem. Quanto menor o limite, mais bordas serão detectadas e o resultado será cada vez mais suscetível a ruído e à detecção de bordas de características irrelevantes na imagem.

DEFINIÇÃO

Por outro lado, um limite alto pode perder bordas sutis ou resultar em bordas fragmentadas. Se a borda for aplicada apenas à imagem de magnitude do gradiente, as bordas resultantes serão geralmente espessas e algum tipo de pós-processamento de afinamento da borda será necessário.

DEFINIÇÃO

Uma possível abordagem para remover esses fragmentos é utilizar um limiar durante a etapa de supressão não-máxima. A escolha do valor desse limiar é uma tarefa complexa, pois pode resultar em bordas falsas se o limiar for muito baixo ou, por outro lado, na perda de fragmentos da borda se o limiar for muito alto. Para resolver esse problema, o operador de Canny utiliza dois limiares diferentes, T_1 e T_2 , com $T_2 > T_1$. Essa etapa é conhecida como limiarização com histerese.

DEFINIÇÃO

Os pontos da borda que possuem gradiente maior que T_2 são mantidos como pontos da borda. Qualquer outro ponto conectado a esses pontos da borda é considerado como pertencente à borda se a magnitude de seu gradiente estiver acima de T_1 . Isso evita que as bordas fiquem fragmentadas em múltiplos segmentos. A escolha dos limiares T_1 e T_2 é feita com base em uma estimativa da relação sinal-ruído.

ALGORITMO DETECÇÃO DE BORDAS

```
1 PImage originalImage;
2 PImage bordasImage;
3
4 void setup() {
5     size(640, 480);
6     originalImage = loadImage("maquina_color.png");
7
8
9     bordasImage = aplicarFiltroCanny(originalImage); // Aplica o filtro de Canny
10    bordasImage.save("maquina_color_canny.jpg");// Salva a imagem com as bordas detectadas
11
12    // Exibe a imagem original e a imagem com as bordas detectadas
13    image(originalImage, 0, 0);
14    image(bordasImage, width/2, 0);
15    noLoop();
16 }
17
18 PImage aplicarFiltroCanny(PImage img) {
19     img.loadPixels();
20     PImage bordas = createImage(img.width, img.height, RGB);
21     bordas.loadPixels();
22
23     float[][] kernelX = {
24         {-1, 0, 1},
25         {-2, 0, 2},
26         {-1, 0, 1}
27     };
28 }
```

Fig. 3 – interface Processing

ALGORITMO DETECÇÃO DE BORDAS

```
30 float[][] kernelY = {
31     {-1, -2, -1},
32     {0, 0, 0},
33     {1, 2, 1}
34 };
35
36 float limiarBaixo = 10;
37 float limiarAlto = 100;
38
39 for (int y = 1; y < img.height - 1; y++) {
40     for (int x = 1; x < img.width - 1; x++) {
41         float valorX = 0;
42         float valorY = 0;
43
44         for (int j = -1; j <= 1; j++) {
45             for (int i = -1; i <= 1; i++) {
46                 int xx = constrain(x + i, 0, img.width - 1);
47                 int yy = constrain(y + j, 0, img.height - 1);
48                 int loc = xx + yy * img.width;
49                 float gray = brightness(img.pixels[loc]);
50                 valorX += kernelX[j + 1][i + 1] * gray;
51                 valorY += kernelY[j + 1][i + 1] * gray;
52             }
53         }
54         float magnitude = sqrt(valorX * valorX + valorY * valorY);
55
56         if (magnitude > limiarAlto) {
57             bordas.pixels[x + y * bordas.width] = color(255);
```

Fig. 4 – interface Processing

ALGORITMO DETECÇÃO DE BORDAS

```
57         bordas.pixels[x + y * bordas.width] = color(255);
58     } else if (magnitude > limiarBaixo) {
59         bordas.pixels[x + y * bordas.width] = color(1);
60     } else {
61         bordas.pixels[x + y * bordas.width] = color(0);
62     }
63 }
64 }
65 }
66
67 bordas.updatePixels();
68 return bordas;
69 }
```

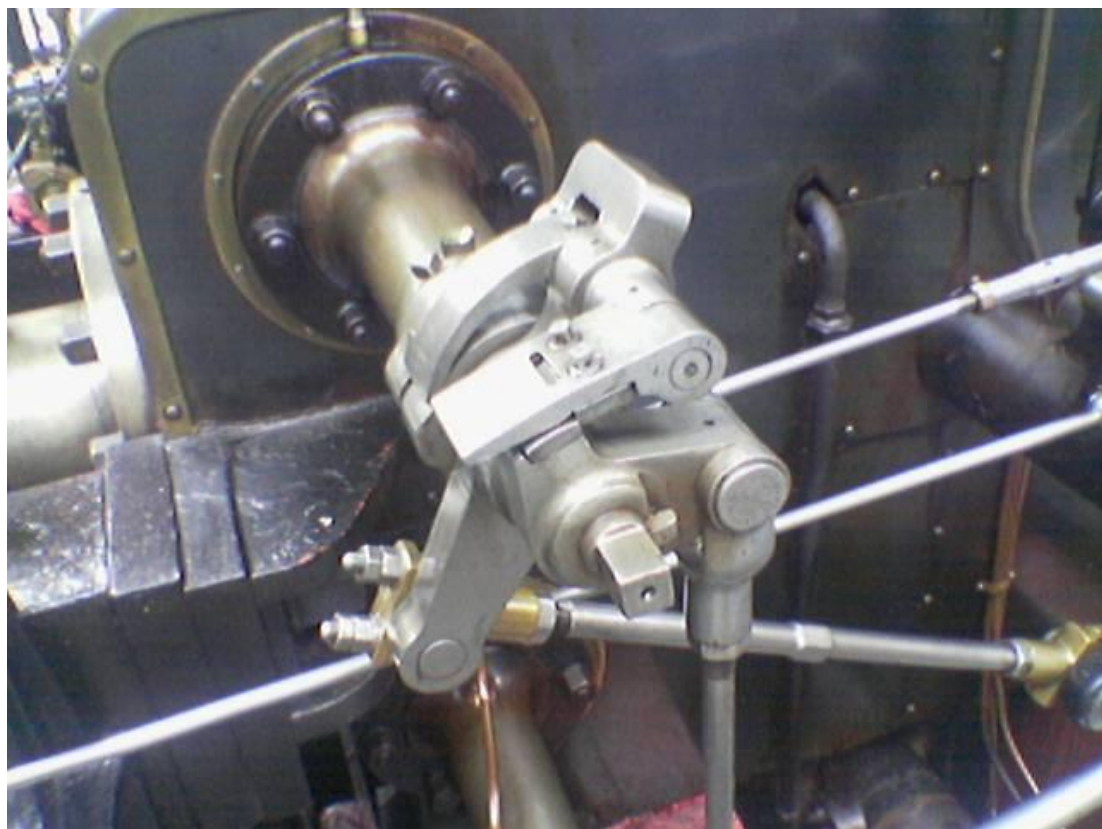


Fig. 6 – imagem de entrada

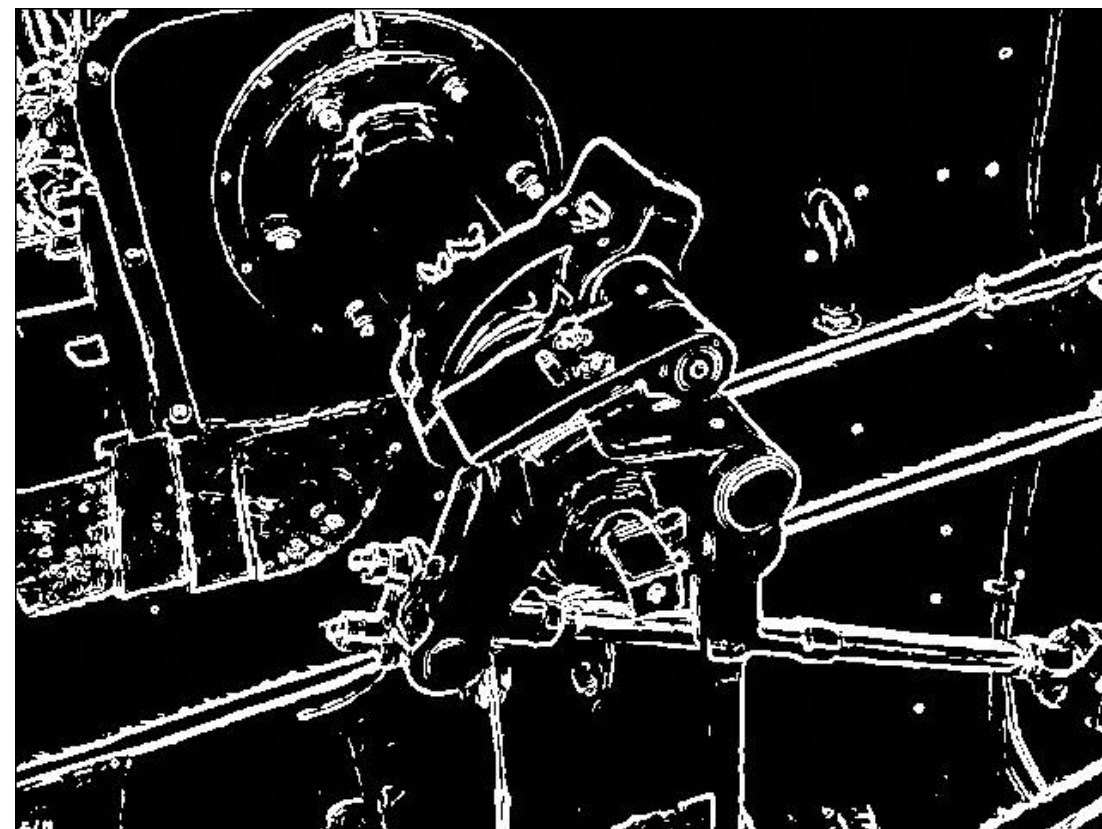


Fig. 7– imagem de saída

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>
- <https://dsp.stackexchange.com/questions/34103/subpixel-what-is-it>