

# Universidade Federal de Pelotas

Curso: Ciência da Computação

Disciplina: Processamento Digital de Imagens

Professor: Bruno Zatt

Relatório 2: Cálculo de Distâncias e Detecção de Fronteiras em Imagens Binárias

Aluno: Yago Martins Pintos

Data: 19/10/2024

## 1. Introdução

Este relatório abrange duas funções principais: `distancias`, que calcula diferentes métricas de distância entre pontos em uma imagem binária, e `fronteira`, que identifica as bordas internas e externas em uma imagem binária. Ambas as funções são essenciais para aplicações em visão computacional e análise de imagens.

## 2. Discussões

### 2.1 Função Distâncias

A função `distancias` calcula três tipos de distâncias entre dois pontos em uma imagem binária:

- Distância D4 (Manhattan): Soma das diferenças absolutas das coordenadas.
- Distância D8 (Chebyshev): Máxima diferença absoluta entre as coordenadas.
- Distância DE (Euclidiana): Distância direta entre os pontos.

Além disso, a função verifica a conectividade entre os pontos na imagem e calcula a distância do menor caminho (DM) se eles estiverem conectados.

### Implementação da Função Distâncias

matlab

```
function [D4, D8, DE, DM] = distancias(arquivo, Xp, Yp, Xq, Yq)
    img = imread(arquivo);

    if size(img, 3) == 3
        error('A imagem fornecida não está em escala de cinza.');
```

```

imagemBinaria = im2bw(img, 0.5);

P = [Xp, Yp];
Q = [Xq, Yq];

D4 = abs(P(1) - Q(1)) + abs(P(2) - Q(2));
D8 = max(abs(P(1) - Q(1)), abs(P(2) - Q(2)));
DE = sqrt((P(1) - Q(1))^2 + (P(2) - Q(2))^2);

fprintf('Distância D4 entre P e Q: %.2f\n', D4);
fprintf('Distância D8 entre P e Q: %.2f\n', D8);
fprintf('Distância DE (Euclidiana) entre P e Q: %.2f\n',
DE);

connectivity = 8;
labeledImage = bwlabel(imagemBinaria, connectivity);

labelP = labeledImage(P(1), P(2));
labelQ = labeledImage(Q(1), Q(2));

if labelP == labelQ && labelP ~= 0
    estaConectado = true;
    distanceMap = bwdistgeodesic(imagemBinaria, P(2),
P(1), 'quasi-euclidean');
    DM = distanceMap(Q(1), Q(2));
    fprintf('P e Q estão conectados. Distância DM (Menor
Caminho) entre P e Q: %.2f\n', DM);
else
    estaConectado = false;
    DM = inf;
    fprintf('P e Q não estão conectados.\n');
end

novoNomeImagem = strcat('binarized_', arquivo);
imwrite(imagemBinaria, novoNomeImagem);

fprintf('Imagem binarizada salva como: %s\n',
novoNomeImagem);

```

end

## 2.2 Função Fronteira

A função `fronteira` identifica as bordas internas e externas de uma imagem binária. Utilizando a análise dos pixels vizinhos, a função determina se um pixel é parte da fronteira interna ou externa.

## Implementação da Função Fronteira

matlab

```
function novoArquivo = medidas_distancia(nomeImagem, Xp, Yp, Xq, Yq)

    % Carrega a imagem em escala de cinza

    img = imread(nomeImagem);

    % Verifica se a imagem é em escala de cinza

    if size(img, 3) == 3

        error('A imagem fornecida não está em escala de cinza.');
```

```

D8 = max(abs(Xp - Xq), abs(Yp - Yq));

DE = sqrt((Xp - Xq)^2 + (Yp - Yq)^2);

fprintf('Distância D4 entre P e Q: %.2f\n', D4);

fprintf('Distância D8 entre P e Q: %.2f\n', D8);

fprintf('Distância DE (Euclidiana) entre P e Q: %.2f\n', DE);

% Determina se P e Q têm conectividade e calcula a distância DM
if imagemBinaria(Xp, Yp) == 1 && imagemBinaria(Xq, Yq) == 1

    DM = calcula_distancia_menor(imagemBinaria, Xp, Yp, Xq, Yq);

    fprintf('P e Q estão conectados. Distância DM entre P e Q:
%.2f\n', DM);

else

    DM = Inf;

    fprintf('P e Q não estão conectados.\n');

end

% Salva a imagem binarizada

novoArquivo = strcat('binarized_', nomeImagem);

imwrite(imagemBinaria, novoArquivo);

fprintf('Imagem binarizada salva como: %s\n', novoArquivo);

```

```

end

function DM = calcula_distancia_menor(imagem, Xp, Yp, Xq, Yq)

    % Inicializa variáveis

    [linhas, colunas] = size(imagem);

    visitados = zeros(linhas, colunas);

    fila = [Xp, Yp];

    distancia = 0;

    while ~isempty(fila)

        novaFila = [];

        for i = 1:size(fila, 1)

            x = fila(i, 1);

            y = fila(i, 2);

            visitados(x, y) = 1;

            % Verifica se chegou ao destino

            if x == Xq && y == Yq

                DM = distancia;

                return;

            end
        end
    end
end

```

```

        % Adiciona vizinhos não visitados

        for dx = -1:1

            for dy = -1:1

                if abs(dx) + abs(dy) == 1 % Movimento ortogonal

                    nx = x + dx;

                    ny = y + dy;

                    if nx > 0 && nx <= linhas && ny > 0 && ny <=
colunas && ...

                                imagem(nx, ny) == 1 && visitados(nx, ny) ==
0

                                    novaFila = [novaFila; nx, ny];

                                end

                            end

                        end

                    end

                end

            end

        end

        fila = novaFila; % Atualiza a fila

        distancia = distancia + 1; % Aumenta a distância

    end

    DM = Inf; % Se não encontrou caminho

```

```
end
```

### 3. Resultados

#### Resultados da Função Distâncias

A função foi testada com uma imagem em escala de cinza. Os resultados obtidos foram:

- Distância D4: Valor calculado corretamente com base nas coordenadas dos pontos.
- Distância D8: Valor calculado corretamente refletindo a máxima diferença.
- Distância DE: Valor Euclidiano calculado com precisão.
- Conectividade: A função corretamente identificou se os pontos estavam conectados ou não.

#### Resultados da Função Fronteira

A função foi testada com uma imagem binária. Os resultados obtidos foram:

- Fronteiras Internas: A função corretamente identificou os pixels que pertencem à fronteira interna (tipo 0).
- Fronteiras Externas: A função também foi capaz de identificar as fronteiras externas (tipo 1), mostrando os pixels que estão na borda do objeto.

### 4. Conclusões

As funções `distancias` e `fronteira` demonstraram ser eficazes na análise de imagens binárias. A primeira permite calcular distâncias entre pontos e verificar conectividade, enquanto a segunda identifica bordas internas e externas. Além disso, o `fprintf`

#### Figuras:

- Figura 1: Imagem original antes da binarização.



- 
- Figura 2:retorno gerado pela função `distancias`.

```

distancias at line 9 column 22
>> distancias('barbara_gray.bmp',5,5,15,10)
warning: function name 'medidas_distancia' does not agree wit
Users\yagom\projects\PDI\Tarefa2\distancias.m'
Distância D4 entre P e Q: 15.00
Distância D8 entre P e Q: 10.00
Distância DE (Euclidiana) entre P e Q: 11.18
P e Q estão conectados. Distância DM entre P e Q: 15.00
Imagem binarizada salva como: binarized_barbara_gray.bmp
ans = binarized_barbara_gray.bmp
>> distancias('barbara_gray.bmp',5,10,15,25)
Distância D4 entre P e Q: 25.00
Distância D8 entre P e Q: 15.00
Distância DE (Euclidiana) entre P e Q: 18.03
P e Q não estão conectados.
Imagem binarizada salva como: binarized_barbara_gray.bmp

```





- Figura 3: Imagem com as fronteiras detectadas pela função `fronteira`