

Universidade Federal de Pelotas

Curso: Ciência da Computação

Disciplina: Processamento Digital de Imagens

Professor: Bruno Zatt

Relatório 4: Realce e Suavização de Imagens em Escala de Cinza

Aluno: Yago Martins Pintos

Data: 19/10/2024

1. Introdução

Este relatório aborda duas funções principais: `realce`, que aplica um filtro Laplaciano para realçar características em uma imagem, e `suaviza`, que utiliza uma máscara média para suavizar a imagem.

2. Função Realce

2.1 Descrição da Função

A função `realce` é projetada para aplicar um filtro Laplaciano a uma imagem em escala de cinza, destacando bordas e detalhes. O filtro Laplaciano é uma técnica de detecção de bordas que enfatiza áreas onde há mudanças abruptas na intensidade da imagem.

Implementação da Função Realce

matlab

```
function imagem_realcada = realce(imagem)
    % Carregar a imagem em escala de cinza
    img = imread(imagem);

    if size(img, 3) == 3
        img = rgb2gray(img);
    end

    laplacian_mask = [0 -1 0; -1 4 -1; 0 -1 0];

    enhanced_img = conv2(double(img), laplacian_mask, 'same');

    enhanced_img = uint8(mat2gray(enhanced_img) * 255);
```

```

[~, name, ext] = fileparts(imagem);
output_filename = strcat(name, '_realçada', ext);

imwrite(enhanced_img, output_filename);
disp(['Imagem realçada salva como: ', output_filename]);
end

```

2.2 Resultados

A função lê uma imagem, aplica o filtro Laplaciano e salva a nova imagem realçada com um nome modificado. O uso do `conv2` permite realizar a convolução da imagem com a máscara Laplaciana, resultando em uma imagem que destaca bordas.

3. Função Suaviza

3.1 Descrição da Função

A função `suaviza` aplica um filtro médio à imagem, suavizando as variações bruscas de intensidade e reduzindo o ruído.

Implementação da Função Suaviza

```

matlab
function imagem_suavizada = suaviza(imagem)
    img = imread(imagem);

    if size(img, 3) == 3
        img_gray = rgb2gray(img);
    else
        img_gray = img;
    end

    img_gray = double(img_gray);

    mascara = (1/9) * [1 1 1;
                       1 1 1;
                       1 1 1];

```

```

[linhas, colunas] = size(img_gray);
[m_linhas, m_colunas] = size(mascara);

imagem_suavizada = zeros(linhas, colunas);

offset_linha = floor(m_linhas / 2);
offset_coluna = floor(m_colunas / 2);

for i = 1:linhas
    for j = 1:colunas
        soma = 0;
        for m = 1:m_linhas
            for n = 1:m_colunas
                linha_nova = i + (m - offset_linha);
                coluna_nova = j + (n - offset_coluna);

                if linha_nova > 0 && linha_nova <= linhas
                    && ...
                        coluna_nova > 0 && coluna_nova <=
colunas
                        soma = soma + img_gray(linha_nova,
coluna_nova) * mascara(m, n);
                        end
                    end
                end
                imagem_suavizada(i, j) = soma;
            end
        end
    end

imagem_suavizada = uint8(imagem_suavizada);

nome_arquivo = 'imagem_suavizada.png';
imwrite(imagem_suavizada, nome_arquivo);

disp(['Imagem suavizada salva como: ', nome_arquivo]);
end

```

3.2 Resultados

A função lê uma imagem em escala de cinza (ou converte uma imagem colorida), aplica uma máscara média para suavização e salva a nova imagem suavizada. O uso da convolução manual permite um controle detalhado sobre o processo de suavização.

4. Conclusões

- Realce: A aplicação do filtro Laplaciano destacou as bordas e detalhes na imagem original, resultando em uma nova imagem que facilita a visualização das características importantes, entretanto se perdeu muito de outras características
- Suavização: A aplicação do filtro médio reduziu o ruído e suavizou as transições na intensidade da imagem, tornando-a mais uniforme.

Figuras:

- Figura 1: Código da função `realce`.

```
function imagem_realcada = realce(imagem)

%Crie uma função
%Matlab Octave que:

%Receba uma imagens em escala de cinzas;

%Realce a imagem aplicando uma convolução da imagem com a máscara
%do Laplaciano apresentada abaixo;

%Crie uma nova imagem realçada;

%Retorne como resultado o nome do arquivo gerado com a nova
imagem.

% Carregar a imagem em escala de cinza

    img = imread(imagem);

    if size(img, 3) == 3
        img = rgb2gray(img);
    end
```

```

laplacian_mask = [0 -1 0; -1 4 -1; 0 -1 0];

enhanced_img = conv2(double(img), laplacian_mask, 'same');

enhanced_img = uint8(mat2gray(enhanced_img) * 255);

[~, name, ext] = fileparts(imagem);
output_filename = strcat(name, '_realçada', ext);

imwrite(enhanced_img, output_filename);

disp(['Imagem realçada salva como: ', output_filename]);
end

```

- Figura 2: Código da função suaviza.

```

• function imagem_suavizada = suaviza(imagem)
•
•
•     img = imread(imagem);
•
•
•
•
•     if size(img, 3) == 3
•
•         img_gray = rgb2gray(img);
•     else
•         img_gray = img;
•     end
•
•
•     img_gray = double(img_gray);
•
•

```



```

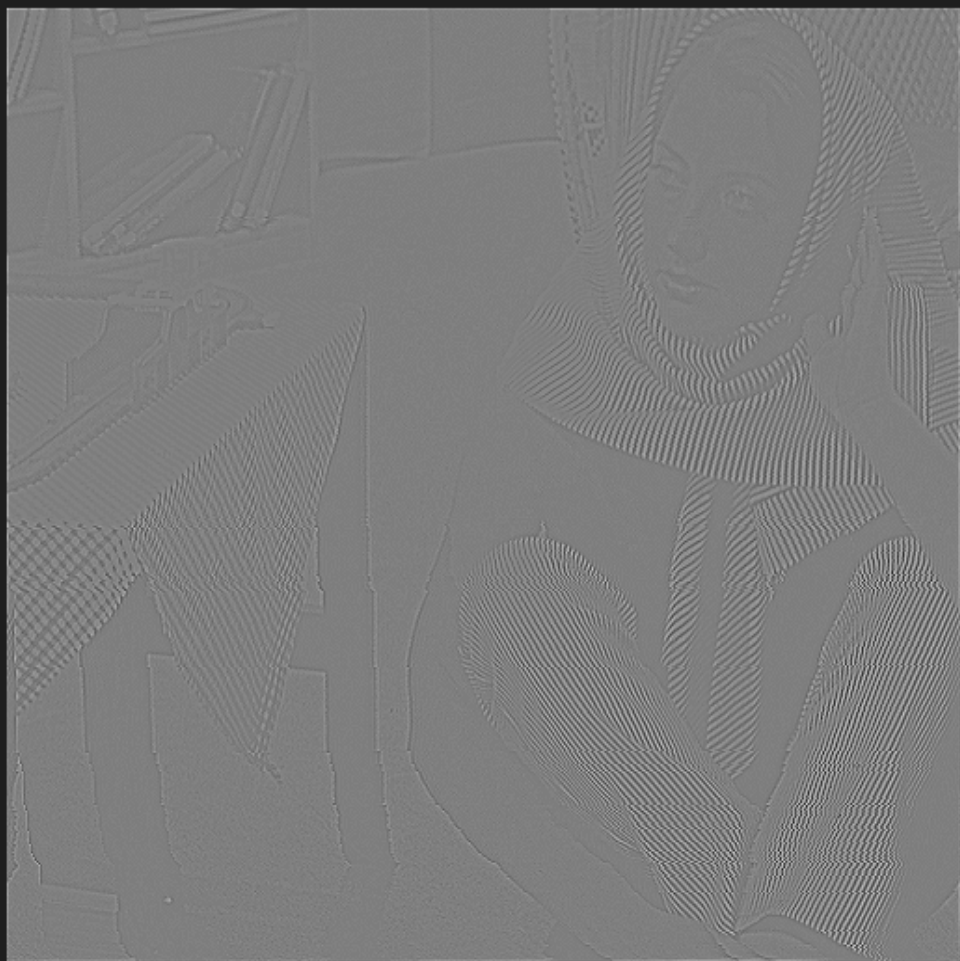
•         imagem_suavizada(i, j) = soma;
•     end
• end
•
•     imagem_suavizada = uint8(imagem_suavizada);
•
•     nome_arquivo = 'imagem_suavizada.png';
•     imwrite(imagem_suavizada, nome_arquivo);
•
•     disp(['Imagem suavizada salva como: ', nome_arquivo]);
• end

```

-
- Figura 3: Imagem original antes do realce e suavização.



-
- Figura 4: Imagem realçada.



-
- Figura 5: Imagem suavizada.



•