

Universidade Federal de Pelotas

Curso: Ciência da Computação

Disciplina: Processamento Digital de Imagens

Professor: Bruno Zatt

Relatório 5: Cisalhamento e Transformação Afim em Imagens

Aluno: Yago Martins Pintos

Data: 19/10/2024

1. Introdução

Este relatório descreve duas funções principais: `cisalhamento`, que aplica uma transformação de cisalhamento a uma imagem, e `afim`, que realiza uma transformação afim nas coordenadas de uma imagem.

2. Função Cisalhamento

2.1 Descrição da Função

A função `cisalhamento` aplica uma transformação de cisalhamento vertical ou horizontal a uma imagem em escala de cinza, dependendo dos parâmetros fornecidos. O cisalhamento é uma transformação geométrica que desloca os pontos de uma imagem em uma direção específica, mantendo a integridade dos pixels.

Implementação da Função Cisalhamento

matlab

```
function cisalhada = cisalhamento(imagem, cv, ch)
    I = imread(imagem);
    H = size(I, 1);
    W = size(I, 2);

    cisalhada = zeros(size(I), 'uint8');

    if ch == 0 && cv >= 0
        for v = 1:H
            for w = 1:W
                valor = I(v, w);
                x = v + cv * w;
```

```

        y = w;
        if x > 0 && x <= H && y > 0 && y <= W
            cisalhada(uint16(x), uint16(y)) = valor;
        end
    end
end
figura_nome = 'lena_cisalhamento_vertical.bmp';
elseif ch >= 0 && cv == 0
    for v = 1:H
        for w = 1:W
            valor = I(v, w);
            x = v;
            y = ch * v + w;
            if x > 0 && x <= H && y > 0 && y <= W
                cisalhada(uint16(x), uint16(y)) = valor;
            end
        end
    end
    figura_nome = 'lena_cisalhamento_horizontal.bmp';
else
    for v = 1:H
        for w = 1:W
            valor = I(v, w);
            x = v + cv * w;
            y = ch * v + w;
            if x > 0 && x <= H && y > 0 && y <= W
                cisalhada(uint16(x), uint16(y)) = valor;
            end
        end
    end
    figura_nome =
'lena_cisalhamento_horizontal_vertical.bmp';
end

figure, imshow(uint8(I)); title('Imagem Original');
figure, imshow(cisalhada); title('Imagem Cisalhada');

imwrite(cisalhada, figura_nome);

```

```
    disp(['Imagem cisalhada salva como: ', figura_nome]);  
end
```

2.2 Resultados

A função lê uma imagem e aplica o cisalhamento com base nos parâmetros `cv` (cisalhamento vertical) e `ch` (cisalhamento horizontal). Dependendo dos valores fornecidos, a função gera diferentes versões da imagem cisalhada e as salva com nomes apropriados.

3. Função Transformação Afim

3.1 Descrição da Função

A função `afim` recebe coordenadas de um ponto em uma imagem e aplica uma matriz de transformação afim para calcular as novas coordenadas desse ponto. A transformação afim pode incluir operações como rotação, translação e escalonamento.

Implementação da Função Afim

matlab

```
function [x, y] = afim(v, w, T)  
    theta = 30; % ângulo em graus  
    T = [cosd(theta) -sind(theta) 10;  
         sind(theta) cosd(theta) 5;  
         0           0           1];  
  
    coord_homogeneas = [v; w; 1];  
  
    x = T(1,1) * coord_homogeneas(1) + T(1,2) *  
        coord_homogeneas(2) + T(1,3) * coord_homogeneas(3);  
    y = T(2,1) * coord_homogeneas(1) + T(2,2) *  
        coord_homogeneas(2) + T(2,3) * coord_homogeneas(3);  
  
    fprintf('Novas coordenadas: (%.2f, %.2f)\n', x, y);  
end
```

3.2 Resultados

A função calcula as novas coordenadas após aplicar a transformação afim definida pela matriz `T`. O exemplo dado utiliza um ângulo de rotação de 30 graus e deslocamentos nas direções X e Y.

4. Conclusões

As funções `cisalhamento` e `afim` demonstraram ser eficazes na manipulação geométrica de imagens:

- Cisalhamento: A aplicação do cisalhamento permitiu deslocar pixels em direções específicas, resultando em imagens modificadas que podem ser úteis para análise visual.
- Transformação Afim: A função `afim` ilustra como as coordenadas podem ser transformadas usando matrizes, permitindo operações complexas em imagens.

Figuras:

- Figura 1: Código da função `cisalhamento`.

```
function cisalhada = cisalhamento(imagem, cv, ch)

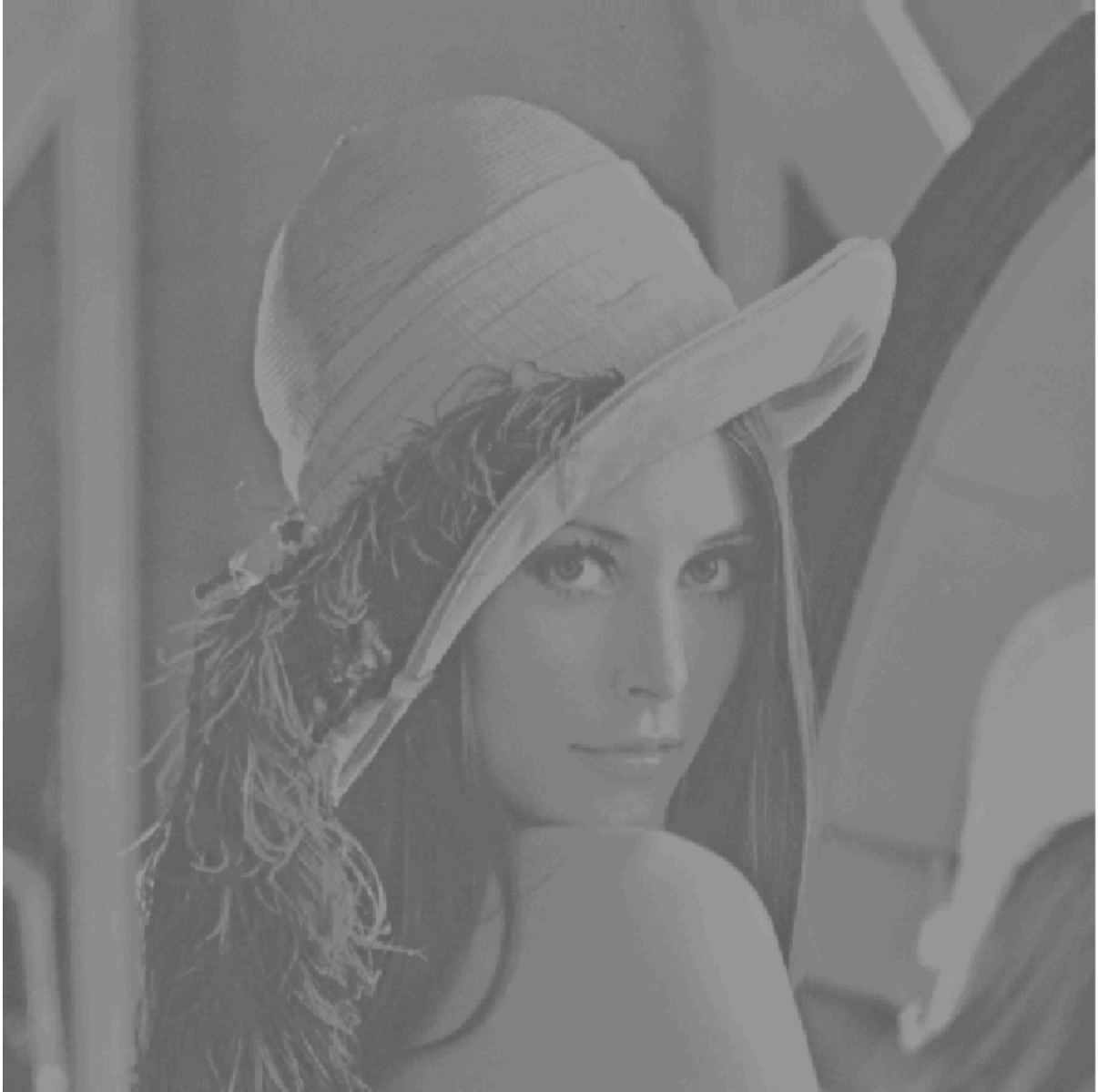
%
%
%   I = imread(imagem);
%   H = size(I, 1);
%   W = size(I, 2);
%
%   cisalhada = zeros(size(I), 'uint8');
%
%   if ch == 0 && cv >= 0
%       for v = 1:H
%           for w = 1:W
%               valor = I(v, w);
%               x = v + cv * w;
%               y = w;
%               if x > 0 && x <= H && y > 0 && y <= W
%                   cisalhada(uint16(x), uint16(y)) = valor;
%               end
%           end
%       end
%   end
```

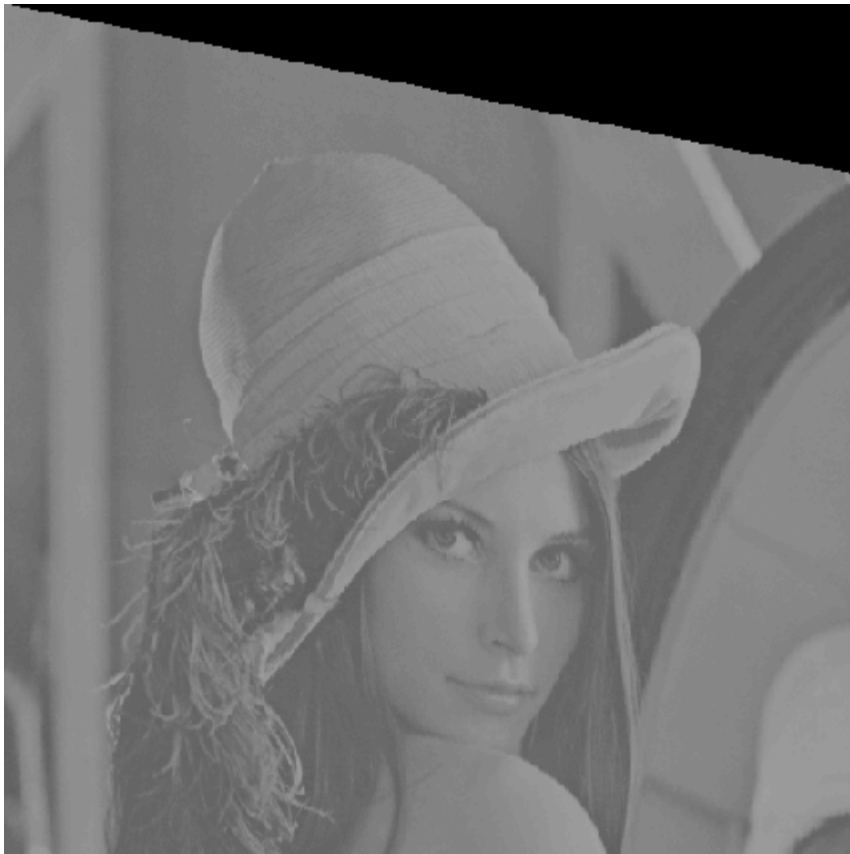
```

•         end
•         figura_nome = 'lena_cisalhamento_vertical.bmp';
•     elseif ch >= 0 && cv == 0
•         for v = 1:H
•             for w = 1:W
•                 valor = I(v, w);
•                 x = v;
•                 y = ch * v + w;
•                 if x > 0 && x <= H && y > 0 && y <= W
•                     cisalhada(uint16(x), uint16(y)) = valor;
•                 end
•             end
•         end
•         figura_nome = 'lena_cisalhamento_horizontal.bmp';
•     else
•         for v = 1:H
•             for w = 1:W
•                 valor = I(v, w);
•                 x = v + cv * w;
•                 y = ch * v + w;
•                 if x > 0 && x <= H && y > 0 && y <= W
•                     cisalhada(uint16(x), uint16(y)) = valor;
•                 end
•             end
•         end
•         figura_nome =
•         'lena_cisalhamento_horizontal_vertical.bmp';
•     end
•
•
•
•     figure, imshow(uint8(I)); title('Imagem Original');
•     figure, imshow(cisalhada); title('Imagem Cisalhada');
•
•
•     imwrite(cisalhada, figura_nome);
•     disp(['Imagem cisalhada salva como: ', figura_nome]);
• end

```


- teste3 = cisalhamento('lena_contraste.bmp', 0.25, 0.1);
-





b

c



