

Universidade Federal de Pelotas

Curso: Ciência da Computação

Disciplina: Processamento Digital de Imagens

Professor: Bruno Zatt

Relatório 10: Compressão de Imagens e Análise de Qualidade

Aluno: Yago Martins Pintos

Data: 19/10/2024

1. Introdução

Este relatório descreve a implementação de três funções: `compc_cp`, que realiza a compressão com perdas usando JPEG, `compc_sp`, que realiza a compressão sem perdas usando PNG, e `plot_taxa_vs_distorcao`, que analisa a relação entre a taxa de compressão e a distorção da imagem em termos de PSNR (Peak Signal-to-Noise Ratio).

2. Função CompC_CP

2.1 Descrição da Função

A função `compc_cp` realiza a compressão com perdas de uma imagem em formato JPEG. O processo inclui:

1. Ler a imagem de entrada.
2. Salvar a imagem compactada com a qualidade especificada.
3. Calcular o tamanho da imagem original e da compactada.
4. Calcular a taxa de bits por pixel (bpp) e o PSNR.
5. Exibir e salvar a imagem compactada.

Implementação da Função CompC_CP

matlab

```
function [bpp, PSNR] = compc_cp(imagem, qualidade)
    img_original = imread(imagem);

    nome_arquivo_saida = 'jpg_com_perdas.jpg';
    imwrite(img_original, nome_arquivo_saida, 'Quality',
qualidade);
```

```

img_compactada = imread(nome_arquivo_saida);

info_original = imfinfo(imagem);
info_compactada = imfinfo(nome_arquivo_saida);

tamanho_original = info_original.FileSize;
tamanho_compactado = info_compactada.FileSize;

num_pixels = size(img_original, 1) * size(img_original,
2);

bpp = (tamanho_compactado * 8) / num_pixels;

mse = mean((double(img_original(:)) -
double(img_compactada(:))).^2);
if mse == 0
    PSNR = Inf;
else
    max_pixel_value = 255;
    PSNR = 10 * log10((max_pixel_value^2) / mse);
end

figure;
imshow(img_compactada);
title(['Imagem Compactada com Qualidade: ',
num2str(qualidade)]);

fprintf('Tamanho original: %.2f bytes\n',
tamanho_original);
fprintf('Tamanho compactado: %.2f bytes\n',
tamanho_compactado);
fprintf('Taxa de compressão: %.4f bits por pixel\n', bpp);
fprintf('PSNR: %.4f dB\n', PSNR);
end

```

3. Função CompC_SP

3.1 Descrição da Função

A função `compc_sp` realiza a compressão sem perdas de uma imagem em formato PNG. O processo é semelhante ao da função anterior, mas não envolve perda de dados.

Implementação da Função `CompC_SP`

matlab

```
function bpp = compc_sp(imagem)
    img = imread(imagem);

    nome_arquivo_saida = 'imagem_compactada.png';
    imwrite(img, nome_arquivo_saida, 'png');

    img_compactada = imread(nome_arquivo_saida);

    info_original = imfinfo(imagem);
    info_compactada = imfinfo(nome_arquivo_saida);

    tamanho_original = info_original.FileSize;
    tamanho_compactado = info_compactada.FileSize;

    num_pixels = size(img, 1) * size(img, 2);

    bpp = (tamanho_compactado * 8) / num_pixels;

    if isequal(img, img_compactada)
        disp('As imagens de entrada e gerada são iguais.');
```

else

disp('As imagens de entrada e gerada são diferentes.');

end

fprintf('Tamanho original: %.2f bytes\n',

tamanho_original);

fprintf('Tamanho compactado: %.2f bytes\n',

tamanho_compactado);

fprintf('Taxa de compressão: %.4f bits por pixel\n', bpp);

```
figure;  
imshow(img_compactada);  
title('Imagem Compactada');  
end
```

4. Função Plot_Taxa_VS_Distorcao

4.1 Descrição da Função

A função `plot_taxa_vs_distorcao` analisa como a qualidade da imagem (em termos de PSNR) varia com diferentes níveis de compressão (qualidade JPEG). A função gera um gráfico que mostra essa relação.

Implementação da Função Plot_Taxa_VS_Distorcao

```
matlab  
function plot_taxa_vs_distorcao(imagem)  
  
    qualidades = [0, 20, 40, 60, 80, 100];  
  
    bpp = zeros(size(qualidades));  
    PSNR = zeros(size(qualidades));  
  
    img_original = imread(imagem);  
  
    for i = 1:length(qualidades)  
  
        nome_arquivo_saida =  
sprintf('imagem_compactada_%d.jpg', qualidades(i));  
  
        imwrite(img_original, nome_arquivo_saida, 'Quality',  
qualidades(i));  
  
        img_compactada = imread(nome_arquivo_saida);  
  
        info_compactada = imfinfo(nome_arquivo_saida);  
        tamanho_compactado = info_compactada.FileSize;
```

```

        num_pixels = size(img_original, 1) *
size(img_original, 2);

        bpp(i) = (tamanho_compactado * 8) / num_pixels;

        mse = mean((double(img_original(:)) -
double(img_compactada(:))).^2);
        if mse == 0
            PSNR(i) = Inf;
        else
            max_pixel_value = 255;
            PSNR(i) = 10 * log10((max_pixel_value^2) / mse);
        end
    end

    figure;
    plot(bpp, PSNR, '-o', 'LineWidth', 2);
    xlabel('Bits per Pixel (bpp)');
    ylabel('PSNR (dB)');
    title('Curva Taxa vs Distorção');
    grid on;
end

```

5. Resultados

5.1 Resultados da Compressão com Perdas

Após executar a função `comp_c_cp`, os resultados foram exibidos na tela mostrando o tamanho original e compactado da imagem, juntamente com a taxa de bits por pixel e o valor do PSNR.

5.2 Resultados da Compressão sem Perdas

A função `comp_c_sp` foi testada para verificar se as imagens originais e compactadas eram idênticas e exibiu os tamanhos correspondentes.

5.3 Análise Taxa vs Distorção

A função `plot_taxa_vs_distorcao` gerou um gráfico que ilustra como o PSNR varia com diferentes níveis de compressão JPEG.

6. Conclusões

- Compressão com Perdas: A função `compc_cp` permitiu observar como a qualidade do JPEG afeta o tamanho do arquivo e a qualidade visual.
- Compressão sem Perdas: A função `compc_sp` confirmou que as imagens podem ser armazenadas sem perda de informações.
- Análise Taxa vs Distorção: A relação entre taxa de bits por pixel e PSNR foi visualizada graficamente, permitindo uma melhor compreensão do impacto da compressão na qualidade das imagens.

Imagem Compactada com Qualidade: 50



Tamanho original: 263224.00 bytes
Tamanho compactado: 35437.00 bytes
Taxa de compressão: 1.0815 bits por pixel
PSNR: 30.9569 dB
ans = 1.0815
Tamanho compactado: 185744.00 bytes

Taxa de compressão: 5.6685 bits por pixel
ans = 5.6685

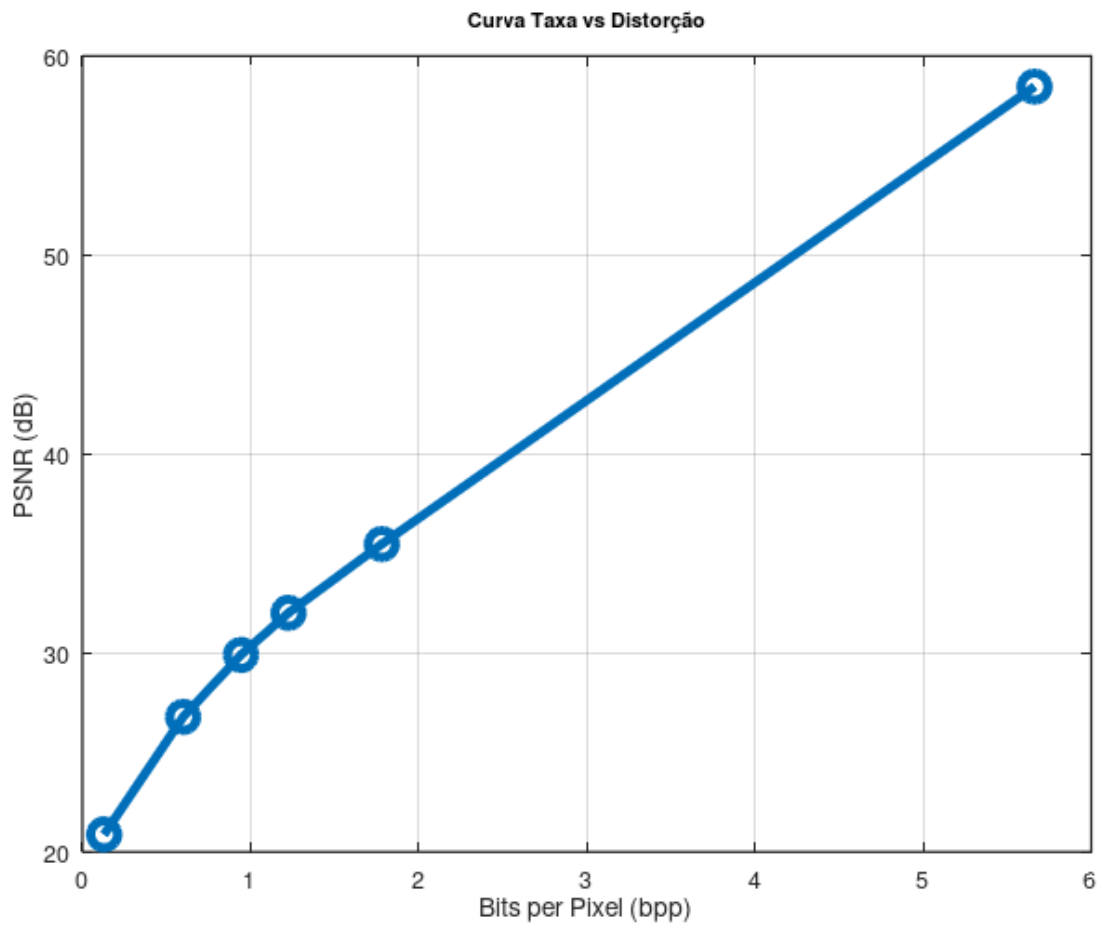


Imagem Compactada

