

Projeto 2 de Fundamentos de Compressão de Sinais

Quantização

I. PROJETO

O projeto é **individual** e consiste em gerar quatro codificadores de imagem:

- 1) Codificador de Imagens em escala de cinza para 2 cores (preto e branco) (C1)
- 2) Codificador de Imagens em escala de cinza para 2 cores (preto e branco) com *Dithering* (C2)
- 3) Codificador de Imagens coloridas baseado em mapa de cores (C1Map)
- 4) Codificador de Imagens coloridas baseado em mapa de cores com *Dithering* (C1Map2)

O programa pode ser implementado em qualquer linguagem de programação. É recomendado utilizar uma linguagem de programação que o aluno tenha experiência no desenvolvimento, de forma que o aluno possa desenvolver os demais projetos utilizando a mesma linguagem. Porém, é sugerido que o projeto seja implementado em linguagem C/C++, Python ou Matlab.

Os codificadores são descritos detalhadamente a seguir. Em todos os casos, vamos medir a taxa como o *número de bits por pixel (bpp)* e a distorção como a PSNR - *peak signal-to-noise ratio*. A PSNR entre duas imagens é definida a partir do erro médio quadrático (*mean squared error* - MSE):

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i, j) - R(i, j))^2$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{(2^b - 1)^2}{MSE} \right)$$

onde m e n corresponde ao tamanho da imagem, b corresponde ao bitdepth da imagem, I corresponde a imagem de teste e R corresponde a imagem de referência.

II. CODIFICADOR 1 - C1

Este codificador de imagens deve aceitar como entrada apenas imagens em escala de cinza e quantizar cada pixel individualmente para a cor mais próxima (preto ou branco).

III. CODIFICADOR 2 - C2

A ideia agora é utilizar um dos algoritmos de *Dithering* para tentar melhorar a qualidade visual da imagem. Nesta técnica, **adicionamos** ruído de forma **intencional** para melhorar a qualidade subjetiva da imagem. O aluno pode utilizar o algoritmo de Floyd-Steinberg, Atkinson, ou algum outro (veja a referência: <https://surma.dev/things/ditherpunk/>).

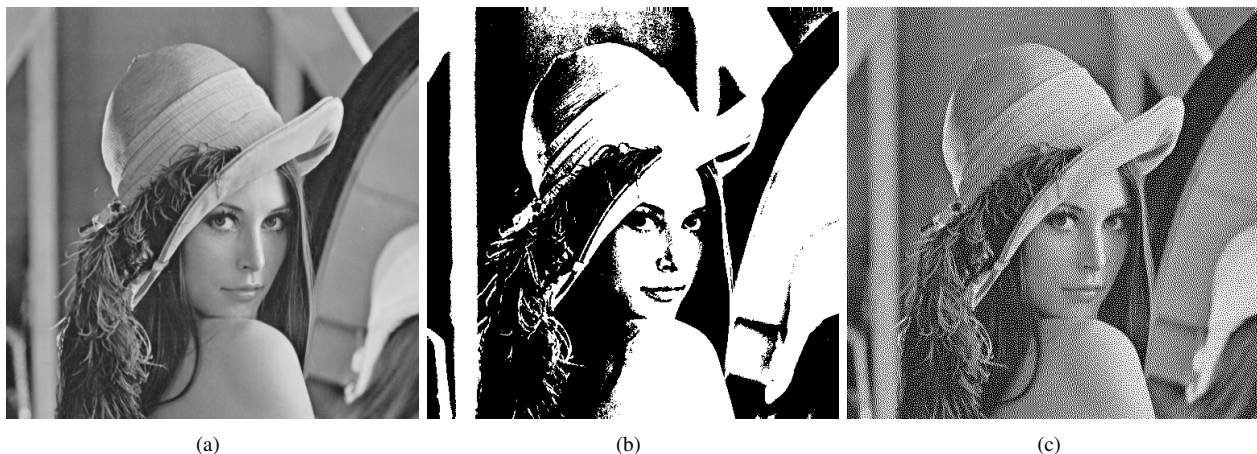


Fig. 1. Exemplo do codificador: (a) Imagem Original (em escala de cinza, com 256 níveis); (b) Quantizada para 2 níveis; e (c) Quantizada para 2 níveis com dithering.

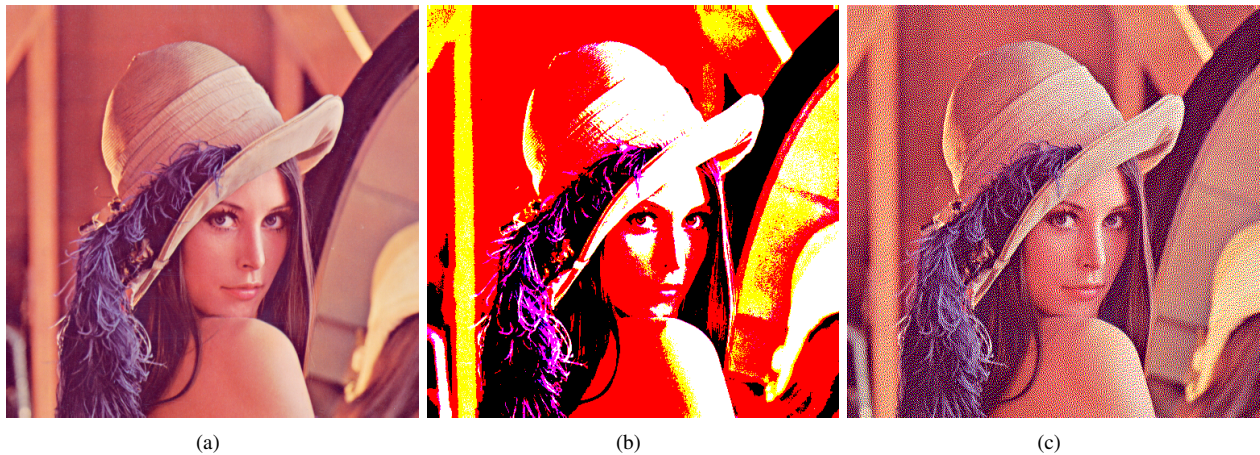


Fig. 2. Exemplo do codificador: (a) Imagem Original (com 16 milhões de cores); (b) Quantizada para 8 cores; e (c) Quantizada para 8 cores com dithering. No exemplo foram utilizadas as 8 cores das pontas do cubo RGB, e não as 8 cores otimizadas com o algoritmo LBG.

IV. CODIFICADOR 3 - CIMAP

A ideia aqui é utilizar quantização vetorial para quantizar as cores da imagem. Uma imagem colorida tem três canais de cor, R , G e B , compostos por três matrizes de tamanho $m \times n$. A ideia é quantizar as cores da imagem com Quantização Vetorial - reduzindo, efetivamente, o número de cores da imagem. Efetivamente, estamos fazendo um quantizador vetorial com $L = 3$, e aplicando nas cores da imagem. O único parâmetro é o tamanho do codebook, e o programa deve funcionar para $M = [8, 16, 32, 64, 128, 256]$ cores. Para encontrar o codebook utilizar o algoritmo LBG. Na comparação, compare a distorção (PSNR) do seu codificador com uma imagem em JPEG com tamanho similar, plotando um gráfico *psnr versus bpp* (você pode usar o GIMP, ou o Matlab, ou qualquer outro codificador JPEG).

V. CODIFICADOR 4 - CIMAP2

A ideia aqui é aplicar a técnica de dithering ao codificador 3. Novamente, pode ser utilizado qualquer algoritmo. Compare a PSNR do Codificador 3 com a PSNR do codificador 4 (note que ambos tem a mesma taxa).

VI. RELATÓRIO

Deve ser entregue um pequeno relatório, em formato de artigo IEEE, para relatar algoritmo desenvolvido, os experimentos feitos, seus resultados e a conclusão.

- Os algoritmos desenvolvidos deve ser *brevemente* descritos.
- O experimento deve conter testes com os arquivos do professor (disponíveis no Teams).
- Os resultados devem conter comparações entre os diferentes codificadores desenvolvidos e o codificador comercial JPEG (sugestão: utilizar o GIMP para codificar com o JPEG com um tamanho específico).
- A conclusão deve conter observações sobre os resultados, incluindo as comparações, e observações sobre o que pode ser feito para melhorar o algoritmo desenvolvido.

O relatório, o código-fonte, e os arquivos utilizados nos experimentos devem ser entregues, via Teams, em um único arquivo .ZIP (ZIP, não rar, pkzip, ARJ, etc...).