

# Pré-processamento e Quantização em Codificação com Perdas

Rodrigo Ferreira Guimarães

Departamento de Ciência de Computação e Faculdade de Tecnologia

Universidade de Brasília, Brasília

Email: rodrigofegui@aluno.unb.br

Matrícula: 14/0170740

## I. INTRODUÇÃO

Este trabalho visa fixar os conceitos relativos a realce no domínio espacial, filtragem no domínio da frequência e de morfologia matemática. Para atingir tal objetivo foram consideradas três imagens-base e dois problemas: **a)** aplicar a DCT em duas dimensões, seguida pela quantização e pela DCT inversa; **b)** similar ao anterior, com a diferença de que deve ser realizada o DPCM antes dos passos já descritos. Para ambos os problemas foram realizadas análises sobre a variância dos blocos requantizados e sobre a PSNR entre as imagens.

30 de novembro, 2015

## II. EMBASAMENTO TEÓRICO

Para que haja um correto entendimento sobre o desenvolvimento deste projeto é importante abordar alguns aspectos relevantes, como as definições: de uma imagem; codificação de imagens; da DCT e sua inversa; do DPCM; da PSNR, dentre outros assuntos.

### A. Imagem e Pixels

Uma imagem pode ser definida como uma função bidimensional do tipo  $f(x, y)$ , onde  $x$  e  $y$  são as coordenadas espaciais e a amplitude de  $f$  em qualquer ponto de coordenadas  $(x, y)$  é denominado de intensidade da imagem naquele ponto. Quando  $x$ ,  $y$  e  $f$  são valores *finitos* e *discretos*, a imagem é denominada imagem digital, tendo esta significância aos computadores digitais. Um dado elemento com coordenadas  $(x, y)$  e intensidade  $f$  é denominado de *pixel* (picture element ou, em português, elemento de imagem), dessa forma, entende-se que uma imagem é constituída por um ou mais *pixels*.

Para a manipulação dos *pixels* é necessário saber as relações básicas entre eles, como, por exemplo, a vizinhança. Os conceitos a serem apresentados consideram uma imagem em nível de cinza.

Cada *pixel*  $p$  pode possuir três tipos de vizinhança, semelhantes a Rosa-dos-Ventos da Figura 1: *vizinhança de 4*, *vizinhança diagonal* e *vizinhança de 8*; onde para o primeiro, denotado por  $N_4(p)$ , são considerados os quatro vizinhos horizontais e verticais, seguindo as orientações *N-S-L-O* da Rosa-dos-Ventos; para o segundo, denotado por  $N_D(p)$ , são considerados os vizinhos das diagonais, seguindo as orientações *NE-SE-SO-NO*; enquanto que o terceiro tipo, denotado por  $N_8(p)$ , é a junção dos dois anteriores.

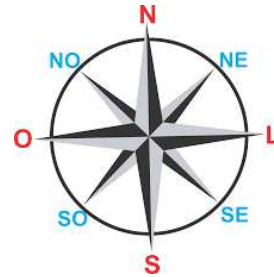


Figura 1: Sinalização dos sentido cardeais da Rosa-dos-Ventos. Fonte [6].

### B. Codificação de Imagens

Ao considerar que sinais possuem dados redundantes e que imagens são sinais, digitais, então tem-se que as imagens possuem dados redundantes. O ramo do processamento de imagens que lida com estes dados redundantes é a codificação, pois objetiva diminuir a quantidade de dados necessários para a reconstrução da imagem sem a perda de qualidade aliada a um fator de codificação, havendo duas formas de se trabalhar: com ou sem perdas (*lossy* ou *lossless*, respectivamente). Para este projeto é aplicada uma parte da codificação com perdas.

A principal característica da codificação com perdas é o fato de que a imagem reconstruída não é idêntica à original, mas possui somente as informações úteis aos olhos humanos.

Os codificadores com perdas possuem um esquema padrão quanto aos passos a serem seguidos, de forma geral, sendo eles: **a)** pré-processamento, consiste na remoção de redundâncias e transformadas para o domínio da frequência, por exemplo, sendo um processo reversível; **b)** quantizador, transforma o alfabeto de símbolos para um outro de conjunto mais limitado, sendo um processo irreversível; **c)** codificador entrópico, consiste em codificar os símbolos numa sequência de *bits* que possa ser enviada ao decodificador.

### C. DCT e sua Inversa

A DCT (*Discrete Cosine Transform* ou, em português, Transformada Discreta de Cosseno) é uma ferramenta importante para o processamento de imagens, proporciona alta eficiência no cálculo e excelente compressão.

É esperado que os dados reais da entrada tenham uma transição, quase, contínua para a sua vizinha. Dessa forma, a DCT

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right), \text{ para } 0 \leq i, j \leq n-1$$

$$\text{onde } C_f = \begin{cases} \frac{1}{\sqrt{2}}, & f = 0 \\ 1, & f > 0 \end{cases}$$

Figura 2: Equação da DCT bidimensional.

funciona na base que os primeiros elementos tenham vizinhos de pouca variação e os últimos tenham grande variância, ou seja, os primeiros estão nas baixas frequências e os últimos nas altas frequências. Assim, ao realizar uma combinação linear da base com os dados de entradas, é esperado que os coeficientes de maior magnitude sejam os primeiro, enquanto que os últimos coeficientes sejam praticamente nulos, podendo ser desprezíveis.

Ao ser aplicado em imagens, é utilizada a DCT bidimensional, sendo calculada a partir da equação da Figura 2.

Como a DCT leva do domínio espacial ao da frequência é preciso ter uma outra ferramenta que faça o “caminho” da volta, por isso que existe a DCT inversa.

Vale ressaltar que a DCT, aplicada a este projeto, manipula blocos de  $8 \times 8$  pixels.

#### D. DPCM

Sabendo que uma imagem é um sinal analógico amostrado, o DPCM (*Differential Pulse-Code Modulation* ou, em português, Diferencial de Modulação por Codificação de Pulso) realiza a quantização de um sinal de erro, comumente denominado erro de predição. Tal erro é obtido pela diferença entre o sinal amostrado e o sinal estimado pelo preditor linear.

Dessa forma, objetiva-se minimizar o erro médio quadrático de predição, uma vez que o erro é assumido desprezível e que o valor predito é uma combinação linear de  $m$  pixels anteriores.

#### E. PSNR

A PSNR (*Peak Signal-to-Noise Ratio* ou, em português, Relação Sinal-Ruído de Pico), aplica a imagens, representa a medida quantitativa da qualidade de reconstrução, focada a codificação de imagens. Para determinar seu valor, que é expresso em escala logarítmica, cuja unidade é o decibel, é importante determinar o *erro quadrático médio* (*MSE*), seguinte esta equação:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} |I(i, j) - K(i, j)|^2,$$

onde  $I$  e  $K$  são as imagens. Com o *MSE* definido é possível, então, definir a *PSNR* em si, sendo ela determinada através desta equação ou qualquer manipulação algébrica da mesma:

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right),$$

onde  $MAX_I$  é o valor máximo possível para as imagens, que neste projeto corresponde a 255.

#### F. Variância

A variância ( $Var(x)$ ) a variável aleatória é a medida da sua dispersão estatística ao indicar a diferença dos valores reais em relação ao valor esperado. Sendo definida pela equação:

$$Var(x) = \frac{1}{N-1} \sum_{i=1}^N |X_i - E(x)|^2,$$

onde  $E(x)$  é o valor esperado, também denominado de *esperança*, enquanto que  $X_i$  é o valor real das amostras.

### III. METODOLOGIA

Com as definições feitas é possível prosseguir para o desenvolvimento do projeto. Dessa forma, esta seção visa apresentar os passos seguidos para a realização das atividades, desenvolvidas em *MatLab* [2]. Como o projeto é constituído por dois problemas, estes serão abordados separadamente.

#### A. Simples Codificação com Perdas

Como informado na Seção I, este problema consiste em aplica a DCT numa imagem, seguida pela quantização em níveis pré-estabelecidos e pela DCT inversa. Os níveis de quantização considerados foram: 1, 10, 20, 50 e 100.

Primeiramente foi realizado um *padding* nas imagens, pois, como informado na Seção II-C, a DCT manipula blocos de  $8 \times 8$  pixels, logo é preciso ajustar as imagens a esta proporção e para este projeto foi *padding* de zeros.

Em seguida a imagem foi manipulada de bloco em bloco, sendo aplicada esta sequência de passos: **a)** DCT sobre o bloco, seguida pela divisão pelo passo de quantização e pelo arredondamento para  $-\infty$  (função *floor*); **b)** foi calculada a média das variâncias das colunas do bloco transformado; **c)** o bloco foi decodificado através da DCT inversa, seguido pela multiplicação pelo passo de quantização.

Ao finalizar as manipulações nos blocos o *padding* foi removido, a variância média para toda a imagem foi determinada, além da determinação da PSNR da imagem reconstruída com a imagem original.

#### B. DPCM antes da Codificação

Tem um procedimento totalmente análogo ao descrito na Seção III-A, o diferencial está na inserção do DPCM.

Considerando que o bloco corrente está na posição  $(x, y)$  e que a predição realizada considera o bloco na posição  $(x, y-1)$ , então para o primeiro bloco a predição é inviável. Mas para os seguintes, a predição sobre cada *pixel* é realizada sobre o *pixel* correspondente do bloco base da predição.

A predição é a diferença do *pixel* corrente com o *pixel* correspondente no bloco já decodificado. Após a predição ocorre o procedimento já descrito e há o laço das ações até que o bloco corrente seja o último bloco da linha de blocos.

### IV. RESULTADOS

Como dito na Seção III há dois problemas, dessa forma esta seção também será subdividida. Todos os códigos estão disponíveis *online* [5], assim como todas as imagens-resultado obtidas com a execução dos *scripts*. Além disso, só serão expostos os resultados sobre uma imagem-base.



Figura 3: Imagem-base para o processamento, popularmente conhecida como Lena. Fonte: fornecida pelo professor



Figura 4: Imagem reconstruída a partir do passo de quantização 1.



Figura 5: Imagem reconstruída a partir do passo de quantização 50.

#### A. Simples Codificação com Perdas

Foi escolhido mostrar os resultados sobre a Figura 3, onde para esta tem-se: Figura 4 que demonstra a imagem reconstruída a partir do passo de quantização 1, que obteve um PSNR de 49.08db; enquanto que para a Figura 5 foi aplicado o passo de quantização 50, que obteve um PSNR de 18.83db.

A partir das reconstruções foi possível construir a Figura 6, que demonstra a variação da PSNR em função da Variância.

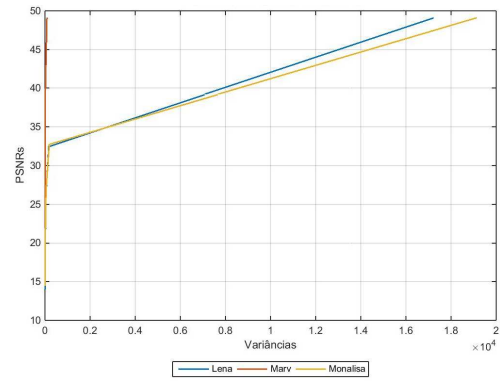


Figura 6: Variação das PSNR dado as Variâncias.

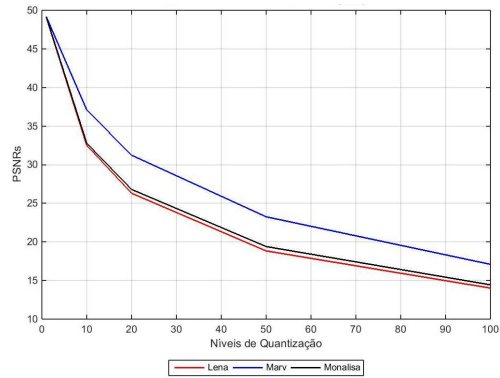


Figura 7: Variação das PSNR das três imagens-base com suas respectivas reconstruídas, dado um nível de quantização.

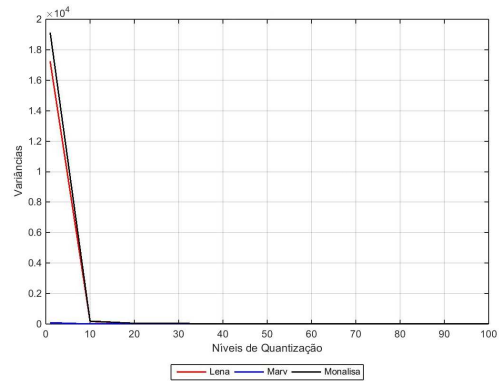


Figura 8: Variação das Variâncias das três imagens-base com suas respectivas reconstruídas, dado um nível de quantização.

Além disso, há as Figuras 7 e 8 demonstram a variação da PSNR e da Variância em função do passo de quantização, respectivamente.

#### B. DPCM antes da Codificação

Foi escolhido mostrar os resultados sobre a Figura 3, onde para esta tem-se: Figura 9 que demonstra a imagem reconstruída a partir do passo de quantização 1, que obteve um PSNR de 10.47db; enquanto que para a Figura 10 foi aplicado o passo de quantização 50, que obteve um PSNR de 10.40db.



Figura 9: Imagem reconstruída a partir do passo de quantização 1, sendo aplicado o DPCM antes da codificação.



Figura 10: Imagem reconstruída a partir do passo de quantização 50, sendo aplicado o DPCM antes da codificação.

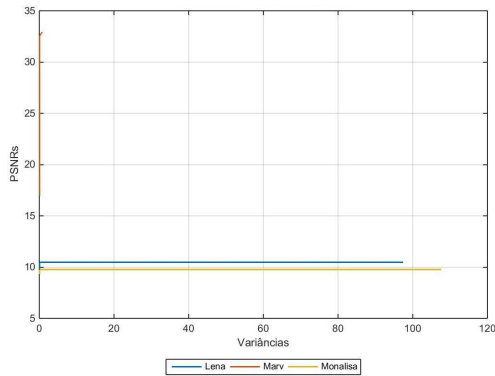


Figura 11: Variação das PSNR dado as Variâncias, com DPCM antes da codificação.

A partir das reconstruções foi possível construir a Figura 11, que demonstra a variação da PSNR em função da Variância. Além disso, há as Figuras 12 e 13 demonstram a variação da PSNR e da Variância em função do passo de quantização, respectivamente.

## V. CONCLUSÃO

Com o desenvolvimento do projeto foi possível sanar algumas dúvidas perante os conteúdos teóricos visto em sala, de-

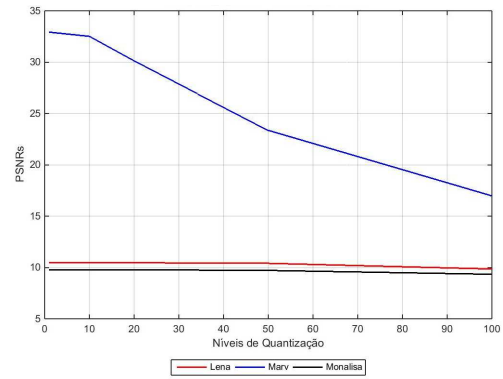


Figura 12: Variação das PSNR das três imagens-base com suas respectivas reconstruídas, dado um nível de quantização, com DPCM antes da codificação.

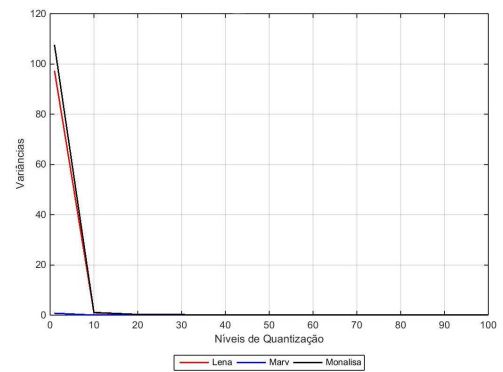


Figura 13: Variação das Variâncias das três imagens-base com suas respectivas reconstruídas, dado um nível de quantização, com DPCM antes da codificação.

monstrando que o objetivo principal foi alcançado. Foram utilizadas as ferramentas disponibilizadas pelo *MatLab*, que possuem grande poder de atuação no processamento de imagens, o que facilitou a fixação dos conceitos, mas é esperado que futuramente seja feita uma aplicação em *OpenCV* [3] (*Open Source Computer Vision Library* ou, em português, Biblioteca de Visão Computacional em Código Aberto).

Ao analisar o primeiro problema foi observado que as imagens reconstruídas a partir de um passo de quantização maior eram tanto visualmente quanto pelo PSNR menos parecidas com as imagens originais, reforçando a ideia de que deve ter uma ponderação em relação ao passo de quantização a ser utilizado, caso se tenha o objetivo de manter as características, pelo menos visuais, das imagens originais.

Ao analisar o segundo problema foi observado que houve um acréscimo de linhas escuras verticais, que seguem o contorno de algumas formas, na imagens. Além disso, há uma grande diferença entre as imagens requantizadas com a original, tanto visualmente como através dos baixos valores encontrados para as PSNRs. Dessa forma, tem-se que as imagens reconstruídas sem o DPCM são mais semelhantes às originais do que as que utilizaram.



## REFERÊNCIAS

- [1] Gonzalez, Rafael C. e Woods, Richard E., *Digital Image Processing*, 3<sup>o</sup> ed, Pearson Ed. - ISBN: 9780131687288.
- [2] MathWorks. *MATLAB and Simulink for Technical Computing*. Disponível em: [https : //www.mathworks.com/index.html](https://www.mathworks.com/index.html), acessado em 2015.
- [3] Documentation, OpenCV. *Welcome to opencv documentation*. Disponível em: [http : //docs.opencv.org/index.html](http://docs.opencv.org/index.html), acessado em 2015.
- [4] Espinoza, B. *Material didático utilizado em aula*.
- [5] 4Shared. *Trabalho 3\_IPI*. Disponível em: [https : //www.4shared.com/zip/5QlXoE2kce/Trabalho\\_3\\_IPI.html](https://www.4shared.com/zip/5QlXoE2kce/Trabalho_3_IPI.html).
- [6] Tríade da Aprovação. *Níveis de Conhecimento: Por onde começar e até onde você deve estudar cada assunto?*. Disponível em: [http : //triadedaaprovacao.com/niveis – de – conhecimento – por – onde – comecar – e – ate – onde – voce – deve – estudar – cada – assunto/](http://triadedaaprovacao.com/niveis-de-conhecimento-por-onde-comecar-e-ate-onde-voce-deve-estudar-cada-assunto/), acessado em 2015.
- [7] Falcção, Alexandre Xavier. *Processamento de Imagens usando Grafos*. Disponível em: [http : //www.ic.unicamp.br/ afalcao/mo815 – grafos/aula6.pdf](http://www.ic.unicamp.br/~afalcao/mo815-grafos/aula6.pdf), acessado em 2015.
- [8] ICMC. *Transformada Discreta de Cosseno: uma aplicação da Álgebra Linear na codificação de imagens do formato JPEG*. Disponível em: [http : //www.icmc.usp.br/ frasson/jpeg/jpeg.html](http://www.icmc.usp.br/~frasson/jpeg/jpeg.html), acessado em 2015.
- [9] UniCamp. *Transformada de Fourier de Sinais Contínuos*. Disponível em: [http : //www.dt.fee.unicamp.br/ peres/ea614/113/pdf/LSScap10.pdf](http://www.dt.fee.unicamp.br/~peres/ea614/113/pdf/LSScap10.pdf), acessado em 2015.
- [10] UniCamp. *Codificação de Sinais Analógicos*. Disponível em: [http : //www.decom.fee.unicamp.br/ baldini/EE881/Cap4.pdf](http://www.decom.fee.unicamp.br/~baldini/EE881/Cap4.pdf), 3acessado em 2015.