

# Processamento e Análise de Imagens Digitais

## Exercício 01 - Amostragem e Quantização

**André Washington Moraes de Freitas - 407343**

Doutorando em Engenharia de Teleinformática

`andre.was12@gmail.com`

**Artur Rodrigues Rocha Neto - 431951**

Mestrando em Engenharia de Teleinformática

`artur.rodrigues26@gmail.com`

13 de Março de 2019

## 1 Introdução

Nesse exercício, são explorados conceitos práticos associados a amostragem e quantização de imagens digitais [1]. Os resultados foram gerados a partir de *scripts*<sup>12</sup> Python usando as bibliotecas OpenCV [2] e SciPy [3]. Para fins de visualização, as imagens redimensionadas foram esticadas usando filtro do vizinho mais próximo para possibilitar a comparação com as imagens em resolução original.

Iremos mostrar o efeito da amostragem e da quantização numa imagem de *benchmark* bem conhecida (`lenna.jpg`), mostrada na Figura 1 ao lado do seu histograma de intensidades.

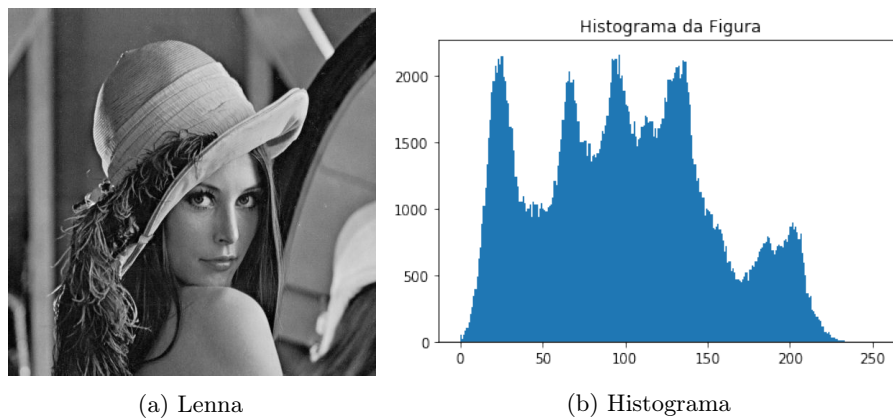


Figura 1: Imagem usada para testes

<sup>1</sup><https://github.com/andre-washintgon/pdi2019/tree/master/Trab1>

<sup>2</sup><https://github.com/keizerzilla/pdi-pos/blob/master/T01.py>

## 2 Metodologia e Resultados

Uma imagem  $f$  pode ser uma função contínua com relação às coordenadas  $x$  e  $y$ , e também com relação à amplitude (valor das coordenadas ou intensidade). Uma imagem digital trata-se de uma imagem que foi amostrada e quantizada a partir de uma imagem  $f$  contínua.

Digitalizar uma imagem  $f$  com relação às coordenadas é chamado de amostragem ou *sampling*. Digitalizar com relação à amplitude é chamado quantização ou *quantization*. Tomando como base o vector obtido a partir da imagem, podemos aplicar a técnica da amostragem de forma direta, tomando amostras igualmente espaçadas do vetor e descartando aquelas amostras que não forma selecionadas. Ao final obteremos um vetor reduzido de amostras para montar a imagem.

A Figura 2 mostra o efeito visual de diferentes tamanhos de amostragem em uma mesma imagem.

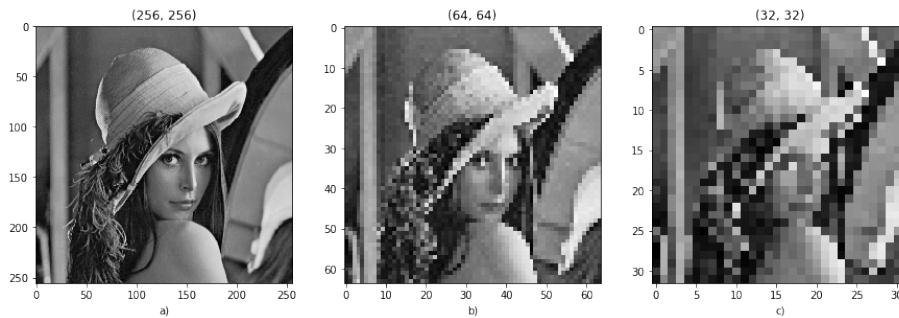


Figura 2: Efeito visual da amostragem

A quantização está relacionada ao nível de intensidade atribuído a cada pixel, sendo que neste caso a imagem `lenna.jpg` tem 256 níveis de cinza (8bits). Podemos fazer algumas modificações nesta imagem reduzindo este número de níveis, sendo por tanto necessário uma decisão para associar cada *pixel* a um novo valor de intensidade dentro deste reduzido conjunto de níveis.

No experimento anterior, a imagem original que foi originalmente amostrada com 512 *pixels* de resolução (pelo sensor optico ou processamento inicial) e depois manipulada para que sua nova resolução fosse mais baixa, ou seja, foi aplicado uma subamostragem. A imagem resultante claramente é pior que a original devido à quantidade de informação que foi removida. Quanto mais informação removida, menor será a qualidade da imagem.

Modificaremos agora os níveis de cinza para demonstrar o efeito da quantização na imagem original de 512 *pixels* de resolução. Definimos uma função que normaliza uma matriz dentro dos níveis definidos, fazendo o papel de quantizador. Ou seja, dada uma imagem de entrada, decidimos para cada *pixel* qual tom de cinza ele pertence por arredondamento. O primeiro teste é para dois tons, ou seja, branco e preto. Sendo assim, qualquer pixel que seja diferente de branco, será preto. Essa técnica é chamada de binarização (Figura 3). Também foram comparados 4 níveis de quantização (Figura 4).

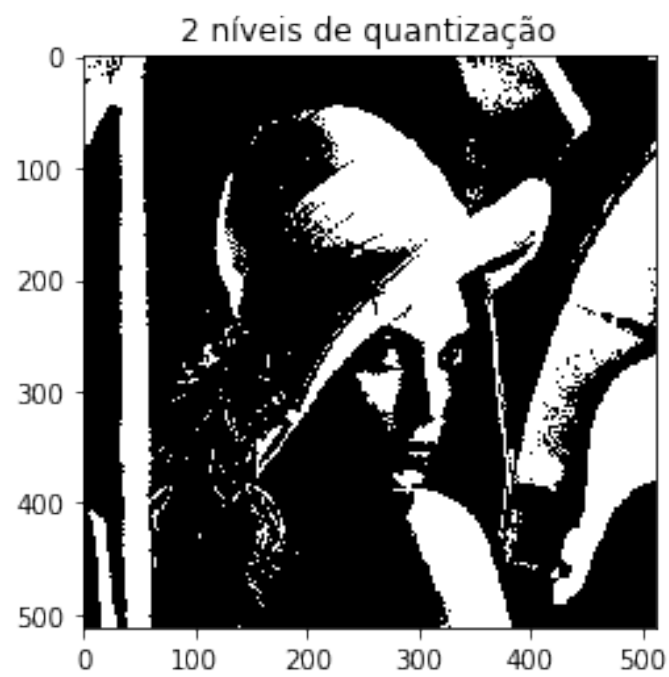


Figura 3: Binarização



Figura 4: Diferentes níveis de quantização

### 3 Conclusão

Por fim, a Figura 5 uma visão comparativa entre os efeitos de amostragem e quantização na qualidade de uma imagem digital.



Figura 5: Resultados nas combinações de transformações de amostragem (redimensão) e quantização (binarização) efetuadas na imagem `lenna.jpg`

### Referências

- [1] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [3] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; acessado 11 de março de 2019].