
RELATÓRIO SOBRE O PRIMEIRO TRABALHO DE PROCESSAMENTO DIGITAL DE SINAIS E IMAGENS

June 24, 2019

Kevin Levrone Rodrigues Machado Silva
Christian Takashi Nakata
Universidade Estadual de Maringá
Departamento de Informática

Contents

0.1	Objetivo e Descrição do problema	2
0.2	Ferramentas e recursos utilizados	2
0.3	Funcionamento do algoritmo	3
0.3.1	Método A	3
0.3.2	Parte B (Algoritmo do Corte Mediano)	4
0.4	Conclusão	5

0.1 OBJETIVO E DESCRIÇÃO DO PROBLEMA

De acordo com Gonzalez [1], uma imagem pode ser contínua em relação às coordenadas x e y e também em relação à amplitude. Para que uma imagem contínua seja convertida para o formato digital, é preciso realizar a amostragem e a quantização da mesma. A primeira consiste na digitalização dos valores de coordenada x e y , enquanto a segunda digitaliza os valores de amplitude.

O objetivo deste trabalho é elaborar dois métodos de quantização de imagens coloridas, representadas no sistema de cor RGB. O primeiro envolve a quantização de uma imagem no referido sistema de cor para n cores resultantes, através da seleção de $n = a * b * c$ pontos uniformemente espaçados. Cada um destes pontos do cubo será representativo de uma cor resultante. Cada cor na imagem de entrada será comparada com uma das cores finais e será associada à cor mais próxima por algum método de cálculo de distância. Nesse trabalho, o cálculo utilizado foi o cálculo de distância euclidiana, cujo os valores entre os pontos são calculadas pela trigonometria espacial de onde elas se encontram.

O algoritmo do Corte Mediano consiste em realizar cortes no cubo de cores conforme a mediana de números de determinados pixels e, em seguida, realizar cortes no cubo de cores para obter uma nova forma de quantizar as cores na imagem. Nesse trabalho, foi implementado uma função recursiva que para toda chamada determina o intervalo de cor do RGB com a maior diferença entre o maior valor e menor valor encontrados. Em seguida obtém o valor da mediana desse intervalo e corta o intervalo referente o valor da mediana.

0.2 FERRAMENTAS E RECURSOS UTILIZADOS

Para o desenvolvimento deste trabalho, foram utilizadas as seguintes ferramentas:

- A linguagem de programação Python (Versão 3);
- A biblioteca Numpy, para carregamento e manipulação dos dados das imagens;
- A biblioteca PIL, para trabalhar com a entrada e saída das imagens e para a conversão das informações em arrays do Numpy;
- Uma imagem de um gato, cujas dimensões são: $733 * 490$.



Figure 1: Imagem original de um gato

0.3 FUNCIONAMENTO DO ALGORITMO

0.3.1 Método A

Para a parte a) do trabalho, definiu os valores de a , b e c ; para criar novos intervalos de R , G e B . Para cada valor de a , b e c ; os intervalos R , G e B ; eram fatiados pela função *numpy.linspace*. Como exemplo, para uma entrada $a = 3$, o intervalo R era fatiado em 3 valores e tornava uma lista como: $[0, 127, 255]$. Após criar os novos intervalos de RGB , elaborou todas as combinações de cores com os novos valores destes intervalos. Em seguida, com a função *numpy.linalg.norm* foi calculado a distância euclidiana normalizada de cada pixel da imagem original até as novas cores geradas pela combinação dos novos intervalos, com o propósito de substituir o pixel original da imagem com a nova cor gerada de menor distância. Para casos testados, onde $N = 8$ e $N = 16$, teve os seguintes resultados.

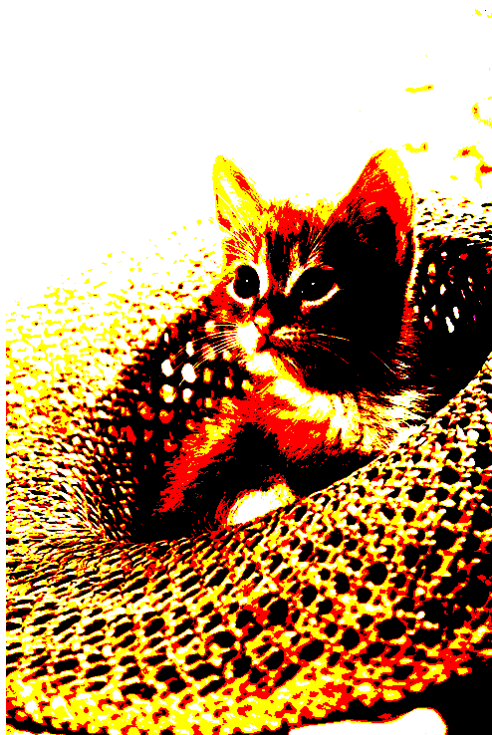


Figure 2: Imagem quantizada com 8 cores, no método A

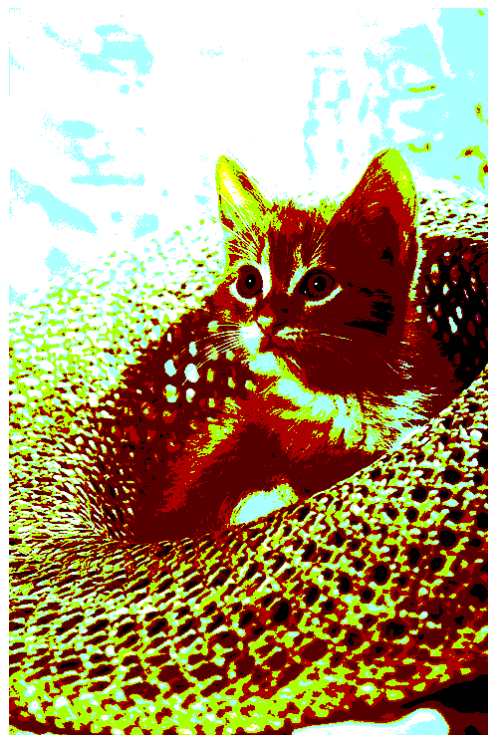


Figure 3: Imagem quantizada com 16 cores, no método A

0.3.2 Parte B (Algoritmo do Corte Mediano)

Para a parte b) do trabalho, implementou o algoritmo do Corte Mediano como uma função recursiva da seguinte maneira:

1. Verifique o intervalo de intensidade de cada cor que a imagem tem;
2. Determine o intervalo com a maior diferença entre o maior valor e o menor valor encontrados nesse intervalo;
3. Busque a mediana desse intervalo;
4. Divida esse intervalo no valor da mediana obtida;
5. Repita recursivamente o processo.

Sendo assim, gerou n-cubos (ou listas) que contém os valores dos pixels da imagem de entrada. Finalmente, para cada pixel da imagem da entrada que corresponder algum valor desses cubos, serão quantizados pelo valor da mediana de todos os pixels de cada cubo. Para casos testados, onde $N = 8$ e $N = 16$, teve os seguintes resultados.



Figure 4: Imagem quantizada com 8 cores, no método B



Figure 5: Imagem quantizada com 16 cores, no método B

Os resultados parecem ser semelhantes, no entanto a figura 6 ficou mais nítida em relação a figura 5.

0.4 CONCLUSÃO

Finalmente, após gerar as imagens para os métodos A e B, estas imagens foram submetidas em função CPSNR (Color Peak Signal-to-Noise Ratio) para comparar os resultados. A função CPSNR são fórmulas matemáticas que comparam a imagem original com a modificada. Quanto maior for o valor resultante, isso implica que a imagem modificada está próxima da original. Os resultados obtidos foram:

Table 1: Tabela dos valores obtidos pela função CPSNR

	Metódo A	Método B
8 cores	27.799184378697362	29.493339156560165
16 cores	27.858765581454765	31.012558595546672

Pode concluir que quanto maior for o número de cores para quantizar, a imagem gerada é mais idêntica ao original. Referente aos métodos, pode dizer que o Método B é mais eficiente para quantizar.

Bibliography

- [1] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.