Projeto 1: Aplicação de Técnicas de meio tons

.

MC920 - Introduçãao ao Processamento Digital de Imagem (MC920 / MO443) 2S2019
Professor: Hélio Pedrini

Giovani Nascimento Pereira
giovani.x.pereira@gmail.com
168609

I. Especificação do Problema

As técnicas de meio-tons são empregadas para reduzir a quantidade de cores utilizadas para exibir uma imagem, mas procurando manter uma boa percepção visual por parte do usuário [1].

Essa técnica foi empregada fazendo normalizando os valores de cada pixel e então distribuindo o erro (diferença entre o valor original do pixel e seu valor normalizado) de acordo com uma máscara especificada. Neste trabalho foram implementadas 6 abordagens de distribuições diferentes para esses erros:

- Burkes
- Sierra
- Stevenson and Arce
- Stucki
- Floyd-Steinberg
- Jarvas Judice and Ninke

II. Entrada de dados e execução do programa

O script que faz a aplicação das técnicas de meio-tons foi desenvolvido em Python e tem como arquivo principal o **main.py**. Além disso, arquivos auxiliares foram criados para facilitar a divisão de responsabilidades dentre cada arquivo e deixar mais simples a compreensão de cada um individualmente.

Dentro da pasta *Half-tones* temos o código que executa cada método de meio tom aplicado pelo script separadamente, e o arquivo **arguments.py** cuida dos parâmetros passados na execução do programa (foi feito usando a biblioteca *argparse*).

A entrada esperada do script é uma imagem no formato PNG (*imagem*), de dimensão qualquer. E para explicitar qual método de meio-tom será aplicado a imagem deve-se explicitar através de flags na execução do programa, como por exemplo *-fs* para Floyd Steinberg ou *-b* para Burke ou ainda *-all* para todos os métodos implementados.

```
$ python3 main.py [imagem] [flags]
```

Uma documentação pode ser exibida pelo programa com o comando:

```
$ python3 main.py --help
```

Para exemplificação dos resultados foram adotadas as imagens disponibilizadas pelo professor em https://www.ic.unicamp.br/~helio/imagens_coloridas/.

III. Código e Decisões tomadas

A. Pré processamento

A imagem que será trabalhada foi carregada usando **imageio**, ele já converte o arquivo PNG em uma matriz das dimensões da imagem onde cada posição da matriz representa um pixel e seu nível de cor. Contudo, o tipo de cada item dessa matriz é **uint8** que aceita números representados por até 8 bits (de 0 a 255).

Contudo, esse intervalo de valores não é suficiente para fazer as operações matemáticas necessárias, nem mesmo aceita números negativos, e poderia causar overflow nos valores, o que traria resultados incorretos no processamento.

Para solucionar esse problema, a matriz foi convertida em **int64** durante o processamento, e depois novamente convertida em **uint8** para salvar como arquivo PNG.

B. Direção da distribuição de erro

A implementação das distribuições de erro seguiram um padrão *zigue-zague* ao longo das matrizes das imagens. Movendo em x de 0 a x_{max} e depois de x_{max} a 0, alternado, incrementando y. Isso, para cada banda de cor. Essa implementação está representada na imagem 1. Esse modelo segue para tentar reduzir efeitos de padrões indesejados ou direcionalidade que poderiam surgir na imagem de saída usando apenas uma movimentação unidirecional no eixo x [1].

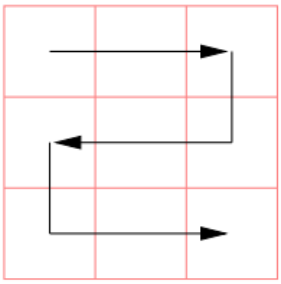

Figura 1. Esquema da direção que o algoritmo de distribuição de erro percorre a matriz da imagem

C. Implementação das técnicas de meio tom

Dado o algoritmo que percorre a matriz da imagem, a cada pixel era aplicado o algoritmo de distribuição de erro solicitado.

Quando um ponto de distribuição de erro se encontrava em uma posição xyz que não pertencia à matriz (casos de borda), eles foram puramente desconsiderados.

Cada técnica de meio-tom foi implementada em um arquivo separado com o mesmo nome da técnica empregada.

IV. Resultados

A. Imagens geradas

Executando o script para todas as distribuições implementadas e usando a imagem *baboon.png* como entrada, foram geradas as imagens da figura 2.

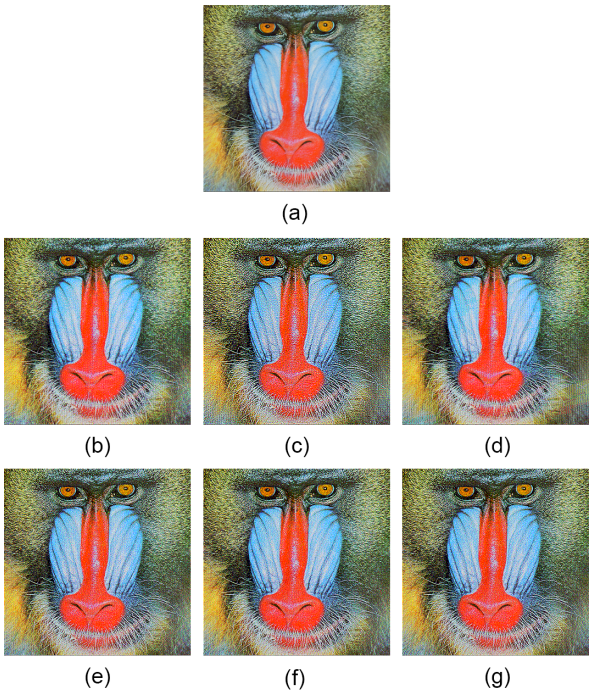

Figura 2. Imagens geradas através do método de meios tons. (a) Imagem original, (b) Floyd-Steinberg, (c) Stevenson and Arce, (d) Burkes, (e) Sierra, (f) Stucki, (g) Jarvas Judice and Ninke

V. Análise

A. Considerações gerais

Durante a análise dos resultados das imagens obtidas, ficou evidente que o visualizador de imagens, bem como a dimensão com que a imagem era exibida tinha um impacto muito grande na percepção dos resultados de aplicação as técnicas de meio-tons [2].

Na máquina onde o projeto foi desenolvido os resultados se apresentavam como esperado, com canais de cor monocromáticos, quando os pixels eram obervados usando o espaço de cor sRGB. Outros espaços de cor como P3 ou Adobe RGB mostravam resultados diferentes para as mesmas imagens analisadas, que não correspondiam ao esperado. E isso dependia da ferramenta de visualização utilizada.

Dessa forma, toda a análise das cores resultantes pelos processos de meio-tons foram feitas observando através do espaço de cor sRGB.

B. Pontilhado

Se oservarmos uma imagem individualmente é possível notar que seus pixels, um a um, são formados exclusivamente por intensidades 0 e 255. A imagem 3 mostra uma seção 4x4 pixels da imagem 2 (a), *baboon* aplicada a distribuição de Floyd Steinberg.

255 255 0	255 255 0	0 0 255	0 255 0
255 255 255	0 0 0	255 255 0	255 0 0
0 255 255	255 255 255	0 0 0	0 255 0
0 255 0	255 0 255	255 255 0	0 0 255

Figura 3. Seção 4x4 da imagem 2 (a), *baboon* aplicada a distribuição de Floyd Steinberg, com os valores das intensidades dos canais RGB em order representados dentro de cada abstração dos pixels.

Esse era o resultado esperado, dessa forma, cada canal de cor, individualmente está monocromático. Mas mesmo assim, a percepção da imagem como um todo não é alterada. Uma pessoa com visão comum ainda é capaz de identificar o conteúdo da imagem, mesmo que ela tenha passado pela transformação de meios-tons.

C. Comparação dos Métodos de distribuição de erro

Usando como referência a imagem *monalisa.png* disponibilizada pelo professor para a confecção do trabalho, foram executados os algoritmos de meio tons e extraídos alguns exemplos para análise. Os resultados são exibidos na figura 4. A imagem original pode ser observada no *Apêndice A* figura 5.

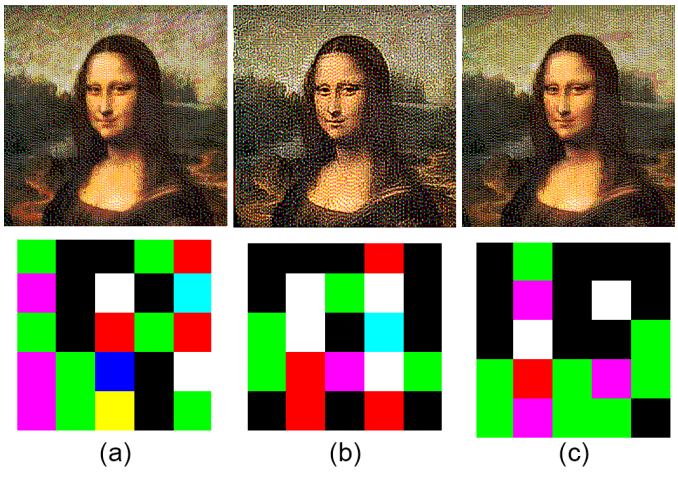

Figura 4. Comparação entre as imagens resultantes e um corte 5x5 do canto superior esquerdo das imagens geradas. (a) Floyd-Steinberg, (b) Stevenson and Arce, (c) Burkes.

VI. Conclusão

Os resultados encontrados para as técnicas de meio-tons foram satisfatórios. Todas as imagens (seja em escala de cinza ou em cor) eram reconhecíveis após a aplicação de qualquer uma das técnicas de meio tom, e ao analisar os pixels das imagens resultantes, para cada banda de cor, a imagem havia se tornado monocromática (ou com nível 0 ou 255).

Cada método de meio tom tem um resultado diferente, sendo que cada um pode ser mais recomendado dado o tipo de imagem que está sendo processada.

Referências

[1] Helio Pedrini. Trabalho 1. Introdução ao Processamento Digital de Imagem (MC920 / MO443), 2019.

[2] SRGB. disponível em https://en.wikipedia.org/wiki/SRGB. Acesso 11 de setembro de 2019.

Apêndice A - Imagens

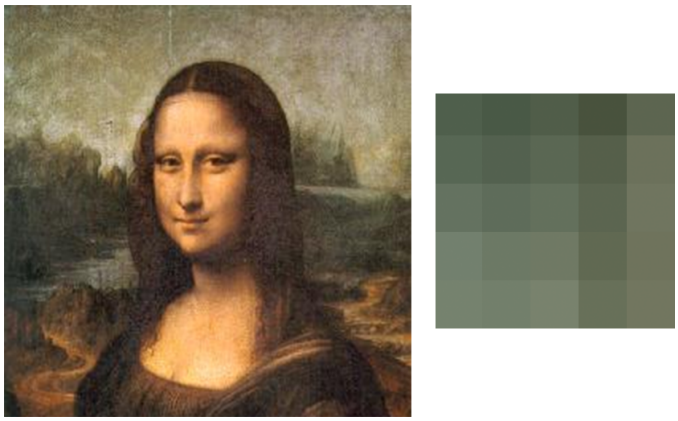

Figura 5. Imagem de teste *monalisa.png* original, com corte 5x5 do canto superior esquerdo da imagens.