

Aplicando técnicas de meios-tons por difusão de erro (trabalho 2)
Introdução ao Processamento de Imagem Digital
Randerson A. Lemos (103897) 2022-1S

1 Introdução

As técnicas de meios-tons se prestam a quantizar as cores de uma imagem (reduzir a quantidade de cores que possui) enquanto procuram manter a imagem resultante, do ponto de vista da percepção visual, o menos alterada possível. Neste trabalho, foram aplicadas as seguintes técnicas de meios-tons por difusão de erro:

Floyd e Steinberg

	f(x,y)	7/16
3/16	5/16	1/16

Stevenson e Arce

		f(x,y)		32/200	
12/200		26/200		30/200	
	12/200		26/200		12/200
5/200		12/200		12/200	
					5/200

Burkes

		f(x,y)	8/32	4/32
2/32	4/32	8/32	4/32	2/32

Sierra

		f(x,y)	5/32	3/32
2/32	4/32	5/32	4/32	2/32
	2/32	3/32	2/32	

Stucki

		f(x,y)	8/42	4/32
2/42	4/42	8/42	4/42	2/42
1/42	2/42	4/42	2/42	1/42

Javis, Judice e Ninke

		f(x,y)	7/48	5/48
3/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48

Detalhes da solução e implementação, assim como de análise e conclusão, serão abordados nas seções subsequentes.

2 Solução

A solução utiliza a linguagem de programação Python e conta com o auxílio do gerenciador de projetos e pacotes Conda. Assumindo que o usuário tenha o Conda instalado em sua máquina, a configuração do projeto pode ser feita pela execução do comando `conda env create -f environment.yml` a partir da pasta do projeto **trab2**. Esse comando cria o ambiente de trabalho **mc920-trab2** e instala os seguintes módulos: opencv, numpy, scipy, pandas, matplotlib. Finalizada a configuração do ambiente de trabalho em questão, o usuário deve executar o comando `source source.sh`¹ para carregar as variáveis de ambiente adequadas e, assim, poder usar os programas do projeto dentro do próprio ambiente de trabalho recém configurado.

Dos arquivos presentes na pasta do projeto **trab2**, destacam-se as pastas **png**, **tex** e o programa **main.py**. A pasta **png** contém imagens no formato png que podem ser utilizadas para a aplicação das técnicas de meios-tonos por difusão de erro. A pasta **tex** contém os arquivos Latex deste relatório. O programa **main.py** contém as implementações das soluções de meios-tonos por difusão de erro utilizadas. As informações pertinentes do programa **main.py** são detalhadas a seguir.

2.1 Main.py

O programa **main.py** é responsável pela aplicação de todas as técnicas de meios-tonos por difusão de erro consideradas neste trabalho. Para ser executado, esse programa precisa receber o parâmetro de entrada **imagem_entrada**:

- ao parâmetro **imagem_entrada** deve-se fornecer o nome do arquivo da imagem a ser utilizada para aplicação das técnicas de meios-tonos por difusão de erro selecionadas.

O programa **main.py** disponibiliza todas as imagens resultantes da aplicação das técnicas de meios-tonos em duas versões: uma versão preto e branco e outra versão colorida. Todas essas imagens são salvas dentro da pasta **out** localizada dentro da pasta do projeto **trab2**. Há também uma imagem de controle, na qual se aplicou uma técnica de meio-tom rudimentar, isto é, apenas considerando o limiar de 128 e não utilizando a difusão de erro. Esta imagem também está salva na pasta **trab2**.

Um exemplo de como executar o programa **main.py** utilizando os recursos contidos dentro do próprio projeto é: `python3 main.py -imagem_entrada=png/baboon.png`.

Sobre a direção de varredura para aplicação das técnicas de meios-tonos optou-se por seguir com a direção que percorre a imagem da esquerda para a direita e de cima para baixo (direção *default*) e com a direção da esquerda para direita com alternância e de cima para baixo (direção *zig-zag*) como ilustrado na imagem da Figura 1.

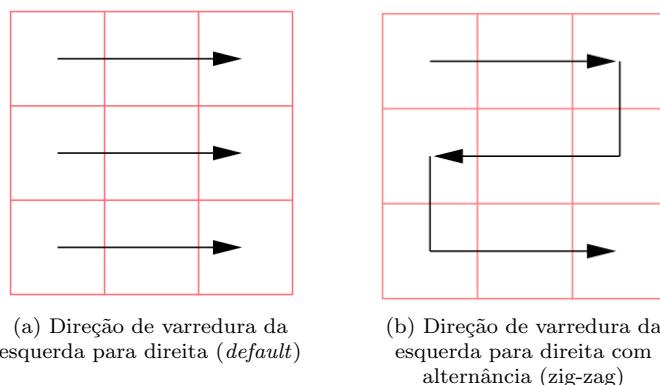


Figura 1: Direção de varredura considerada para aplicação das técnicas de meios-tonos (fonte: imagem extraída do documento pdf com as orientações deste trabalho).

Essas opções de varredura foram consideradas porque a aparência visual das imagens resultantes se mostrou satisfatório, não havendo padrões estranhos que causassem desconforto ao olhar do observador.

Para o processo de aplicação das técnicas de meios-tonos em imagens coloridas optou-se por trabalhar no mapa de cores HSV cujas letras são abreviações de *Hue* (matiz), *Saturation* (saturação) e *Value* (valor). Assim, as imagens que, quando carregadas pelo opencv, eram disponibilizadas no mapa de cores BGR tiveram este mapa de cores convertido para o mapa HSV. Já no mapa de cores HSV, apenas o canal V (Value) da imagem (que é o que contém a intensidade luminosa) foi passado para as funções responsáveis pela aplicação

¹O comando que configura o ambiente de trabalho mc920-trab2 precisa ser executado apenas um vez. Assim sendo, depois que este ambiente está configurado, o usuário precisa apenas executar o comando `source source.sh`

das técnicas de meios-tonos por difusão de erro. Obtido o canal V em meios-tonos, este foi integrado ao esquema HSV original e a nova imagem foi reconvertida para o mapa de cores BGR.

Um dos destaques das implementações realizadas é a classe **Mask**. Essa classe ficou responsável pelo encapsulamento da lógica de recuperação dos pesos das máscaras das técnicas de meios-tonos por difusão de erro bem como das posições nas quais estes pesos devem ser aplicados na imagem original. Dessa maneira, para o usuário, só ficou necessário passar a matriz da máscara da técnica de meio-tom em questão e a posição de referência do posicionamento na máscara do pixel $f(x,y)$ da imagem original. Essa classe foi implementada como um objeto **iterador** do Python de modo que a obtenção dos pesos bem como da posição de aplicação destes na imagem original era facilmente alcançada dentro de uma estrutura de laço de repetição. Isso tornou bastante conveniente a integração da instância da classe **Mask** dentro da função responsável por construir a imagem $g(x, y)$ possuidora dos pixels em meios-tonos. A implementação da classe **Mask** é apresentada a baixo.

```

1  class Mask:
2      def __init__(self, mask):
3          self.name = mask.name
4          self.mask = mask.mask
5          self.ref = mask.ref
6
7
8      def __iter__(self):
9          self._r = 0
10         self._c = -1
11         return self
12
13
14     def __next__(self):
15         mask = self.mask
16         row, col = mask.shape
17         _r = self._r; _c = self._c
18
19         _c += 1
20         if _c == col:
21             _r += 1
22             _c = _c % col
23
24         while True:
25             if _r < row:
26                 val = mask[_r][_c]
27
28                 if val:
29                     self._r = _r
30                     self._c = _c
31                     inc = (_r - self.ref[0], _c - self.ref[1], )
32                     return val, inc # retorna o peso a ser multiplicado pelo erro e o incremento
33                                     # a ser adicionando no pixel (x,y) da imagem f para difusao
34                                     # do erro
35
36                     _c += 1
37                     if _c == col:
38                         _r += 1
39                         _c = _c % col
40
41             else:
42                 break
43
44     raise StopIteration

```

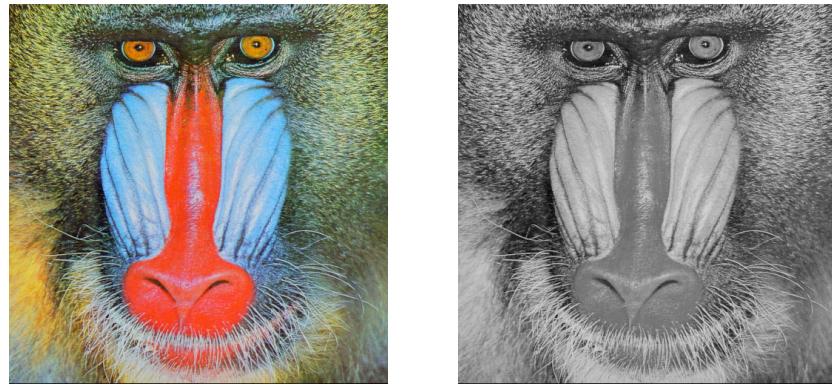
Para demais informações acerca das implementações ou algoritmos utilizados, consultar o código fonte. Lá existem comentários que abrangem esse tipo de informação.

3 Resultado

Os resultados levantados são provenientes da aplicação do programa **main.py** sobre as imagens apresentadas na Figura 2.

Para comparação com as imagens resultantes das técnicas de meios-tonos por difusão de erro, gerou-se imagens denominadas controle a partir da aplicação da técnica de meio-tom que utiliza apenas o limiar de 128 para tomada de decisão se a um pixel será atribuído o valor zero (0) ou o valor (1). As imagens controle da imagem Baboon nas versões colorida e em escala de cinza são apresentadas na Figura 3.

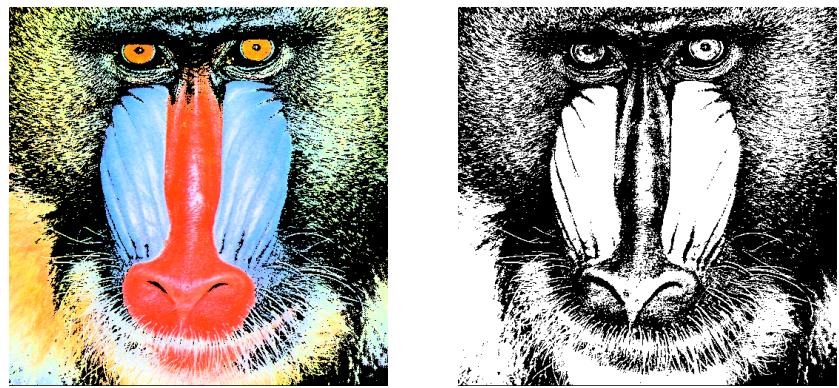
Na sequência, teremos a apresentação das imagens resultantes da aplicação das técnicas de meios-tonos por difusão de erro sobre a imagem Baboon na sua versão colorida e em escala de cinza.



(a) Versão colorida da imagem Baboon
(original)

(b) Versão em escala de cinza da
imagem Baboon (convertida)

Figura 2: Imagens utilizadas para aplicação das técnicas de meios-tons por difusão de erro.



(a) Imagem controle na versão colorida

(b) Imagem controle na versão em preto
e branco

Figura 3: Imagens controle para comparação com as imagens resultantes da aplicação da técnicas de meios-tons por difusão de erro.

3.1 Técnicas de meios-tons sobre images em escala de cinza (varredura *default*)

Aqui serão apresentados as imagens resultantes da aplicação das técnicas de meios-tons sobre a imagem de entrada em escala de cinza.

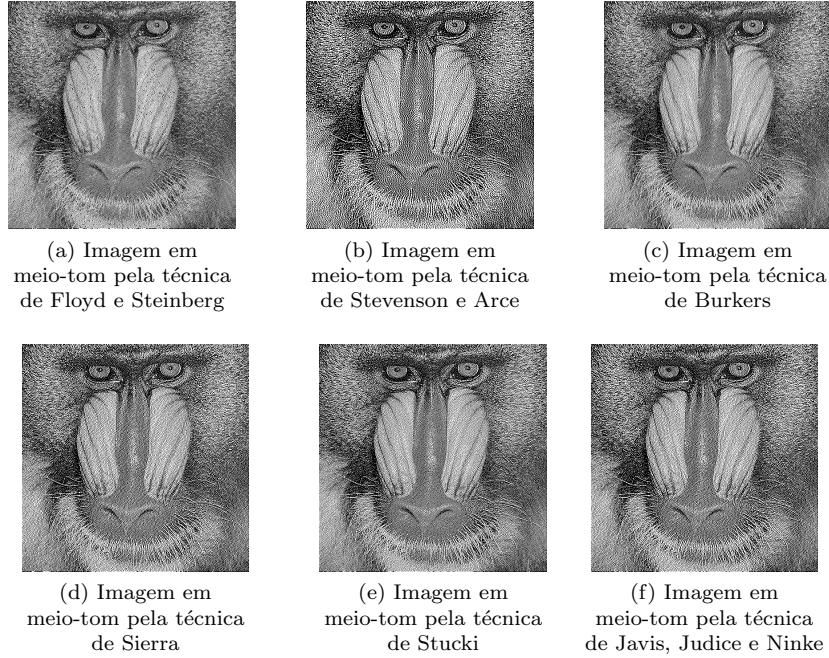


Figura 4: Imagens resultantes em escala de cinza da aplicação das técnicas de meios-tons por difusão de erro em questão usando a varredura *default*.

3.2 Técnicas de meios-tons sobre images em escala de cinza (varredura zig-zag)

Aqui serão apresentados as imagens resultantes da aplicação das técnicas de meios-tons sobre a imagem de entrada em escala de cinza.

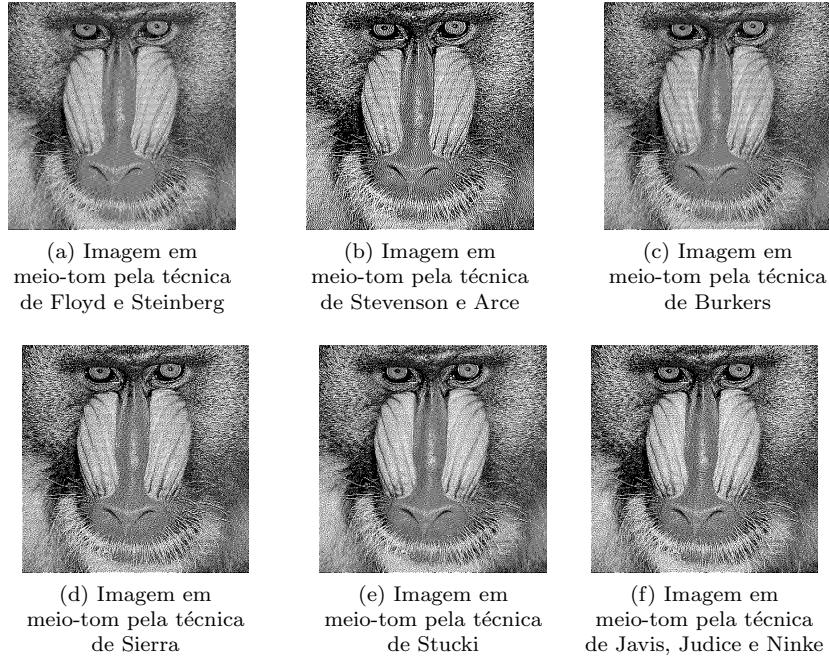


Figura 5: Imagens resultantes em escala de cinza da aplicação das técnicas de meios-tons por difusão de erro em questão usando a varredura zig-zag.

3.3 Técnicas de meios-tons sobre images coloridas (varredura default)

Temos a seguir apresentadas as imagens resultantes da aplicação das técnicas de meios-tons sobre a imagem de entrada colorida.

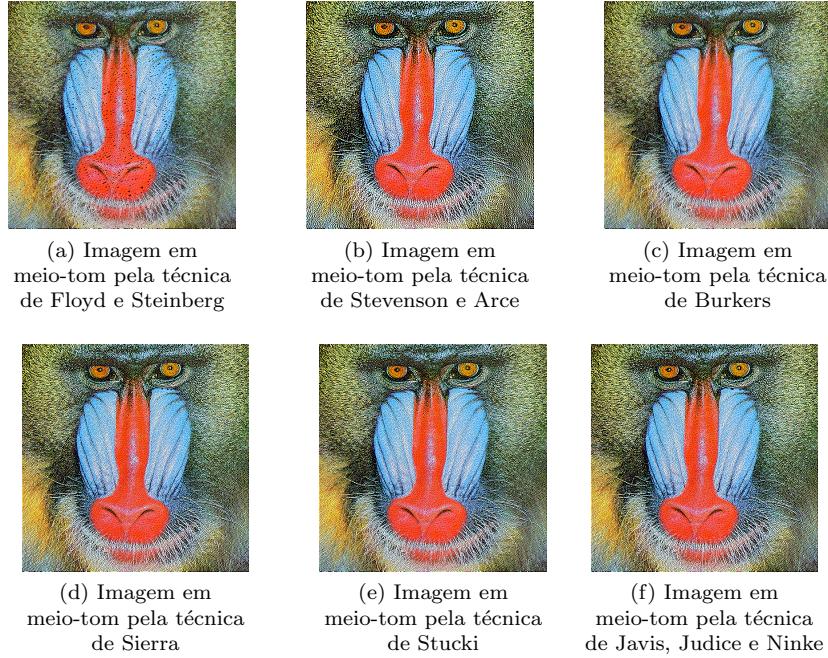


Figura 6: Imagens resultantes coloridas da aplicação das técnicas de meios-tons por difusão de erro em questão usando a varredura *default*.

3.4 Técnicas de meios-tons sobre images coloridas (varredura zig-zag)

Temos a seguir apresentadas as imagens resultantes da aplicação das técnicas de meios-tons sobre a imagem de entrada colorida.

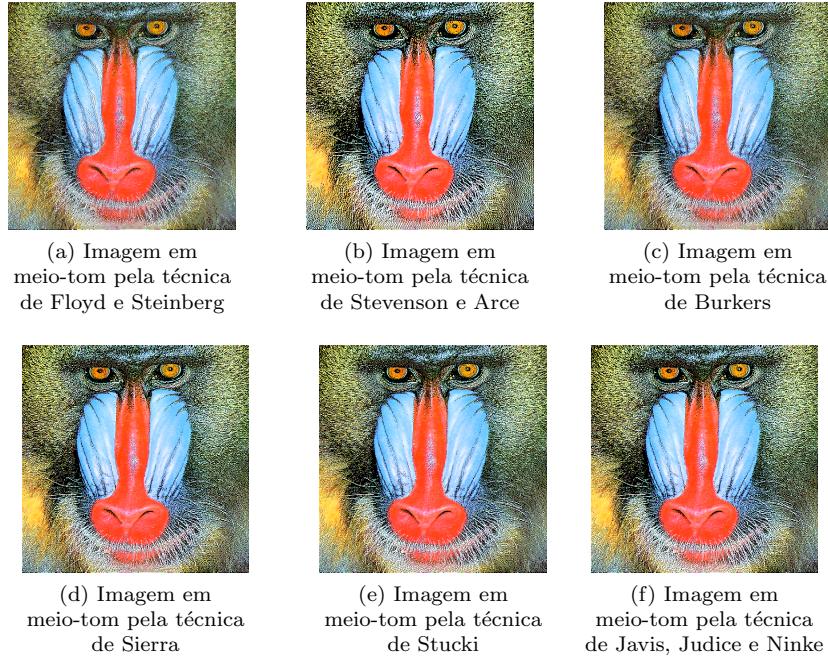


Figura 7: Imagens resultantes coloridas da aplicação das técnicas de meios-tons por difusão de erro em questão usando a varredura *zig-zag*.

4 Dicussão e Conclusão

Do apresentado, podemos concluir que as técnicas de meios-tons foram aplicadas satisfatoriamente tanto sobre imagens coloridas quanto sobre imagens em escala de cinza. Para a imagem utilizada e dentro das dimensões consideradas, as técnicas de meios-tons tiveram resultados visuais semelhantes e bastante satisfatórios. Agora, dentro de uma análise mais minuciosa (ampliando-se as imagens) a técnica de meio-tom de Floyd e Steinberg apresentou uma imagem resultante com algumas falhas e vazios pretos na imagem resultante que não foram apresentados pelas imagens resultantes da aplicação das outras técnicas de meios-tons. Desse modo, quanto ao criterio visual, pode-se afirmar que a técnica de meio-tom de Floyd Steinberg apresentou os resultados menos interessantes com relação a harmonia visual da imagem resultante. Por outro lado, a aplicação da técnica de meio-tom de Floyd Steinberg é a menos custosa computacionalmente, dado que dispõe da menor mascara.

Sobre as duas varreduras utilizadas, verifica-se que o resultante de ambas foi visualmente similar para a imagem utilizada. A diferença mais notável, foi que as imagens resultantes da aplicação das técnicas de meios-tons pela varredura em zig-zag ficaram com maior contraste que as imagens resultantes da varredura *default*. Talvez para aplicações mais sofisticadas em que a qualidade das imagens em meios-tons seja crítica, a utilização de padrões de varredura mais complexos seja justificado, o que não é o caso dentro do contexto deste trabalho.