

**Universidade Estadual de Campinas
Instituto de Computação**

Introdução ao Processamento Digital de Imagem (MC920 / MO443)

Professor: Hélio Pedrini

Aluno: Gabriel Pedroso Mariani

Ra: 197470

Trabalho 2

1. O problema

Nosso objetivo neste trabalho é reduzir o número de cores utilizados para a representação de uma imagem, queremos representar as imagens com apenas duas cores por canal 0 ou 255.

Existem diversas técnicas para esta tarefa. A técnica mais básica seria limiarização, mas o resultado não é muito satisfatório. A técnica escolhida nessa tarefa foi a de halftoning com difusão de erro. Aplicamos essa técnica com varredura esquerda-direita e alternada para imagens rgb e tons de cinza.

2. Modo de uso do Script

Esse Script faz uso da biblioteca OpenCv, NumPy e para rodar basta definir 3 parâmetros.

- i (Requerido)**- caminho para a imagem à qual deseja se aplicar o halftoning.
- m (Requerido)**- qual máscara deseja usar 1 ao 6 (1 = Floyd e Steinberg, 2 = Stevenson e Arce, 3= Burkes, 4 = Sierra, 5 = Stucki, 6 = Jarvis, Judice e Ninke).
- d (Requerido)** - caminho para as imagens destino (formato necessario).
- g** - 1 para se escalada de cinza (colorido é default).
- z** - 1 para varredura alternada (varredura esquerda-direita default)

Ex. python trabalho_2.py -i baboon.png -m 3 -d Baboon_Burkes.png

3. Modo de implementação

Primeiramente foi implementado usando apenas loops, mas como o tempo de execução estava muito alto foi usado o NumPy para otimiza-lo.

Sobre o método otimizado, primeiramente adicionou-se um padding preto (0,0,0) ao redor de toda a imagem base, com um tamanho baseado na máscara.

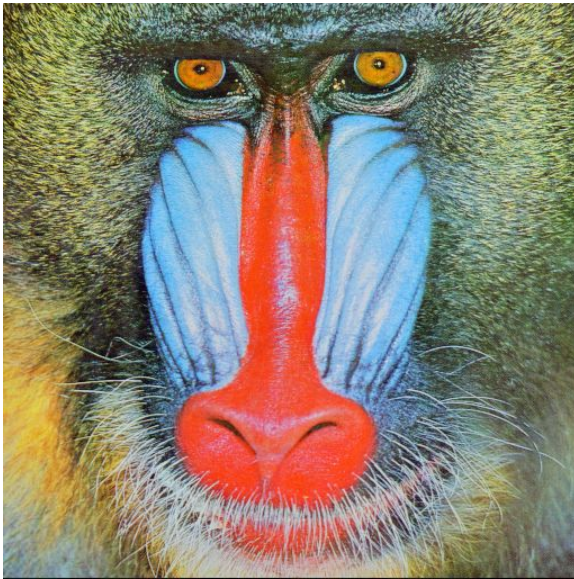
Após isso começou a iteração pelas linhas e colunas. Dentro da iteração primeiro recortou o frame desejado, aplicou-se o threshold de 128 na imagem destino e calculou-se o erro. Então com o frame, com a máscara e com erro, pode-se propagar o erro pela imagem de uma maneira vetorizada.

Todas as Máscaras estão inseridas dentro do código, mas em uma implementação mais completa, poderiam ser lidas de um arquivo.

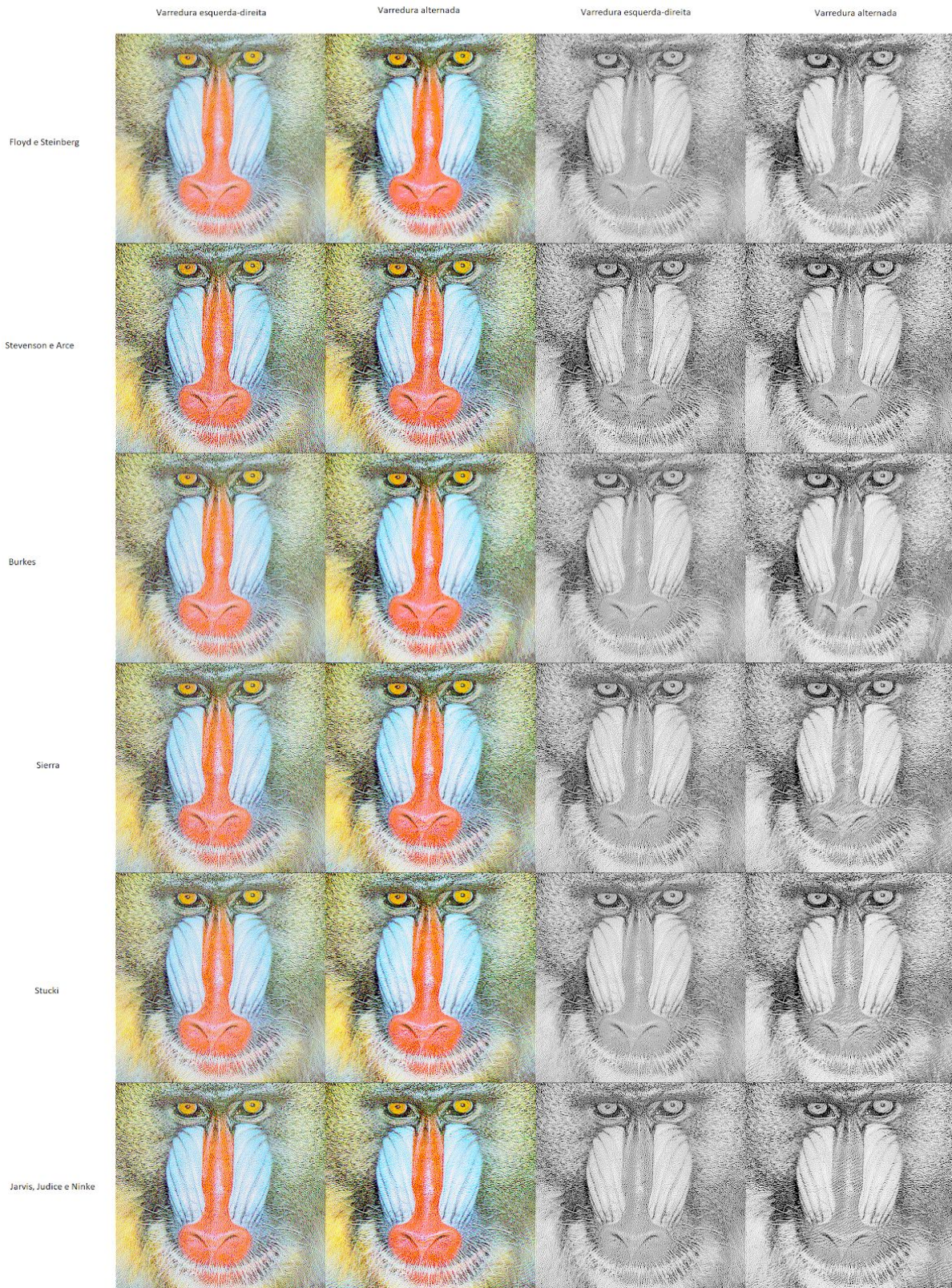
A forma vetorizada do código é bem curta, mas não é tão simples de ser implementada.

4. Resultados

a. Imagens originais usadas



b. Halftoning aplicado



Varredura esquerda-direita

Varredura alternada

Varredura esquerda-direita

Varredura alternada

Floyd e Steinberg

Stevenson e Arce

Burkes

Sierra

Stucki

Jarvis, Judice e Ninke



5. Conclusões

Analisando os resultados, pode-se perceber que a diferença entre cada técnica é pequena, tanto em relação ao kernel quanto em relação à varredura. Analisando o tempo de execução as diferenças não são pequenas quando se fala sobre o tamanho do Kernel.

A técnica de varredura alternada, reduziu a chance de aparições de artefatos, sem que haja prejuízo de tempo de execução, portanto mesmo que a diferença de resultado seja pequena é uma técnica altamente recomendada nesses casos.

6. Considerações finais

Em uma análise pessoal, Burkes foi a que apresentou melhores resultados, mas o custo benefício de Floyd e Steinberg é melhor devido ao tempo de execução.

O trabalho possibilitou um conhecimento mais incisivo sobre técnicas de halftoning, introduziu a manipulação de canais de cores e também possibilitou o aprofundamento em formas de vetorização do processamento de imagem.