MC920: Introdução ao Processamento de Imagem Digital
Trabalho 1

Luciano Zago - 182835

Universidade Estadual de Campinas — 12 de Setembro de 2019

Introdução

O objetivo do trabalho é utilizar a técnica de meios-tons para reduzir a quantidade de cores de uma imagem, de forma a manter uma boa percepção para o usuário. Através de técnicas de pontilhado com difusão de erro, além da variação do modo de varredura da imagem, buscamos melhorar o resultado da técnica de meios-tons.

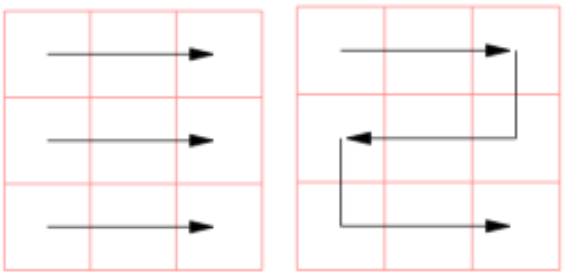
1 Especificação do Problema

Utilizaremos uma imagem colorida RGB de entrada e retornaremos uma imagem pontilhada em 2 níveis para cada camada. Caso o usuário deseje, pode-se retornar uma única camada monocromática com 2 níveis.

Foram utilizados as seguintes abordagens para distribuição de erro e técnicas de pontilhado:

a) Floyd e Steinberg
b) Stevenson e Arce
c) Burkes
d) Sierra
e) Stucki
f) Jarvis, Judice e Ninke

Foram utilizado 2 formas de varredura na imagem, em linha reta e zigue-zague:



(a) Straight (b) ZigZag

Figura 1: Formas de varredura da imagem.

Durante a varredura, o tom do resultado é definido da seguinte forma:

$$dithered[y][x] = \begin{cases} 1 & \text{se } image[y][x] \geq 128 \\ 0 & \text{se não} \end{cases} \quad (1)$$

Depois é calculado a diferença entre o valor do pixel `dithered` e o valor aproximado `image`. Esse erro é distribuido aos pixels adjacentes em `image` segundo a abordagem de distribuição de erro adotada.

2 Implementação

O programa em Python deve ser usado da seguinte forma:

```
./dithering.py mode direction color input output
```

Argumentos:

- `mode` representa a letra referente a abordagem de distribuição de erro, especificadas na seção 1
- `direction` representa a forma de varredura da imagem (straight ou zigzag)
- `color` representa o tipo de cor da saida (rgb ou mono)
- `input` e `output` representam respectivamente os arquivos PNG de entrada e saída

Para rodar todos as opções em todas as imagens de exemplo, existe o arquivo `run.sh` que facilita o processo, e que retorna o tempo de execução.

As matrizes de distribuição de erros estão armazenadas em `np.arrays`, e são retornadas pela função `get_matrix(mode)`.
Caso o modo de varredura for da esquerda para direita, foi necesssário utilizar essa matriz de distribuição de erro espelhada horizontalmente.

Para o tratamento de imagens RGB, foi necessário realizar o `cv2.split` separar as camadas de cor, e realizar o dithering em cada camada separadamente. Após isso, foi necessário juntar novamente as camadas com o `cv2.merge`.

3 Resultados

Os resultados comparando a técnica de pontilhado com os diferentes métodos de difusão de erros e modos de varreduras estão apresentados na Figura 2.

Analisando-se os resultados obtidos, percebe-se que o método de Floyd-Steinberg é o que fornece um resultado mais suave, e melhor para a imagem Monalisa. Utilizando-se o modo de varredura ZigZag permitiu-se uma distribuição mais fiel do erro, pois reduz a impressão de uma possível direcionalidade na imagem resultante.

O método de Stevenson-Arce forneceu o resultado mais distante do original, pois realçou os pontilhados e diminuindo a percepção do usuário em relação à imagem original.

4 Conclusão

Nesse trabalho foi possível utilizar e comparar diferentes métodos de difusão de erro para a técnica de pontilhado, combinados com diferentes modos de varredura da imagem. Verificamos assim, que a técnica de pontilhamento é efetiva para reduzir os níveis de cor de uma imagem, mantendo os detalhes. Porém, existem diferenças importantes entre os diferentes metodos utilizados.
O resultado mais fiél ao original foi o de Floyd-Steinberg com ZigZag, e o menos fiél foi o de Stevenson-Arce com varredura em linha reta.

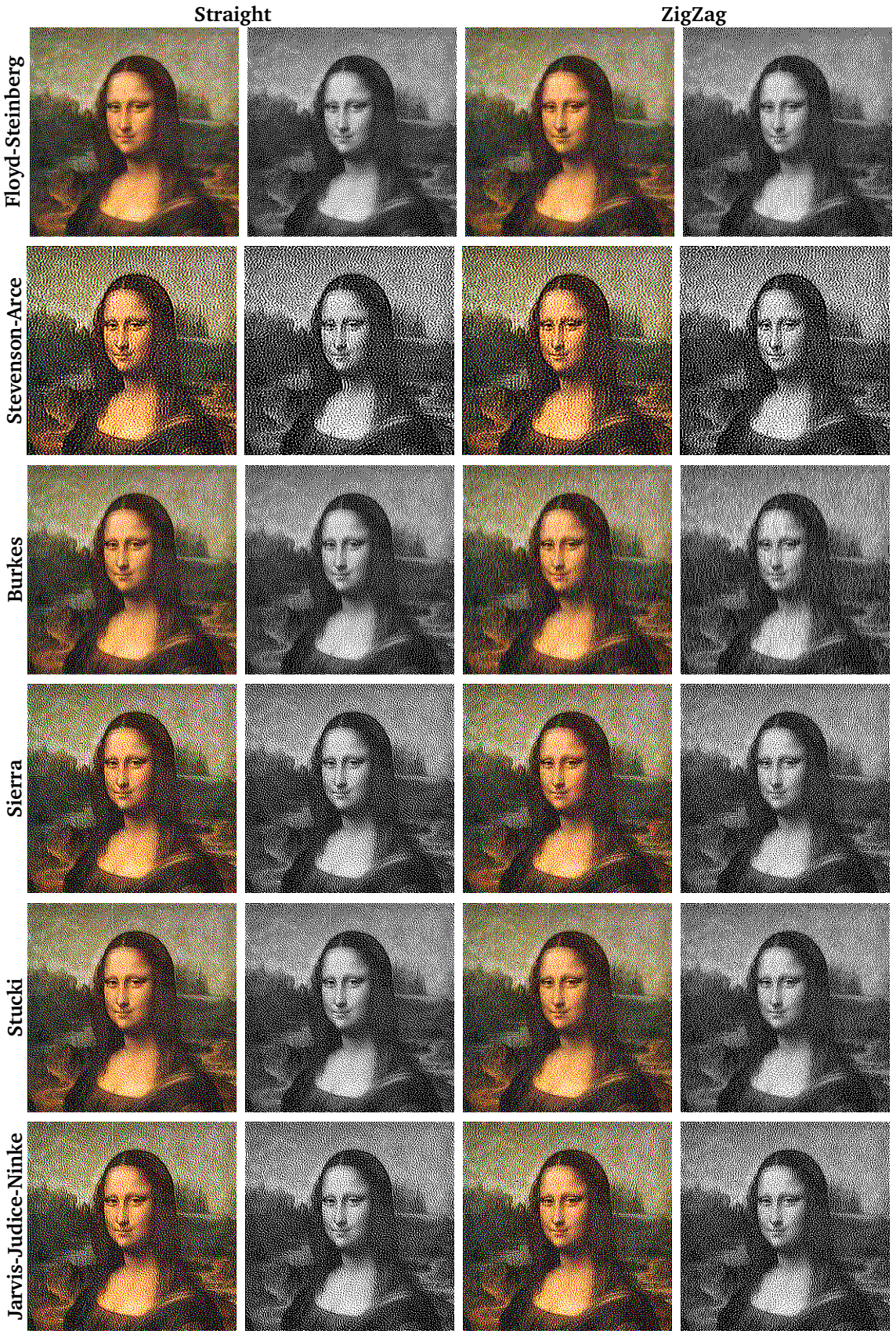


Figura 2: Imagem Monalisa após aplicado a técnica de pontilhado com diferentes métodos de difusão de erro e varreduras