

Técnicas de pontilhado

NATHÁLIA HARUMI KUROMIYA *

*Engenharia da Computação - Graduação
E-mail: n175188@dac.unicamp.br

Resumo – O trabalho se concentra em comparar diferentes técnicas de pontilhado. Para o pontilhado ordenado, foi adotado 2 máscaras diferentes: uma pré-definida e a de Bayer. A outra técnica aplicada. Os algoritmos utilizados foram implementados com auxílio das bibliotecas NumPy e OpenCV e todo o trabalho foi implementado em Python 3. Os resultados evidenciam as diferenças desses métodos de pontilhado e como cada um afeta as imagens e a percepção humana.

I. INTRODUÇÃO

A comparação das técnicas de pontilhamento tem a função de auxiliar a utilização delas e a escolha em cada contexto através de suas particularidades e vantagens. Para a elaboração desse trabalho, foi necessário um estudo maior sobre os métodos de realce e métodos de pontilhado. Para esse fim, utilizou-se o livro[1], os slides[2] disponibilizados pelo professor Hélio Pedrini e o site[3].

II. SEÇÕES

Esse trabalho está separado em tais seções:

- Trabalho Proposto
- Entradas e Saídas
- Programas e bibliotecas utilizadas
- Metodologia e Funções implementadas
- Resultados e Discussão
 - Imagens-referências
 - Pontilhado ordenado
 - Pontilhado por difusão de erros de Floyd-Steinberg
- Resultados
- Referências

III. TRABALHO PROPOSTO [4]

O desenvolvimento do trabalho foi dividido em duas partes. A primeira é responsável pela aplicação de pontilhado ordenado. Para isso, os filtros utilizados foram os seguintes:

$$h1 = \begin{bmatrix} 6 & 8 & 4 \\ 1 & 0 & 3 \\ 5 & 2 & 7 \end{bmatrix}$$

$$h2 = \begin{bmatrix} 0 & 12 & 3 & 15 \\ 8 & 4 & 11 & 7 \\ 2 & 14 & 1 & 13 \\ 10 & 6 & 9 & 5 \end{bmatrix}$$

O filtro $h2$ é também conhecido como **matriz de pontilhado ordenado de Bayer**.

A segunda parte do trabalho foi a aplicação da técnica de **pontilhado por difusão de erro de Floyd-Steinberg**.

IV. PROGRAMAS E BIBLIOTECAS UTILIZADAS

A versão de Python utilizada para implementação foi 3.5.4 e as bibliotecas de auxílio foram NumPy 1.11.2 e OpenCV 3.1.0.

V. ENTRADAS E SAÍDAS

As imagens de entradas possuem o formato .PGM (*Portable GrayMap*), enquanto as saídas foram computadas em .PBM (*Portable BitMap*).

VI. METODOLOGIA E FUNÇÕES IMPLEMENTADAS

A técnica de pontilhado ordenado se baseia em utilizar uma máscara como limiar para os pixels da imagem.

Após normalizar os valores dos pixels da imagem entre 0 e o número de itens na máscara +1, ou seja, o número de configurações diferentes que a máscara oferece, cada pixel é comparado com seu correspondente na máscara. Se o valor do pixel é menor do que o valor correspondente na máscara, então o pixel é colocado como 0. Caso contrário, é colocado como 1. Dado um pixel $p(x, y)$, seu correspondente na máscara M_{axb} pode ser encontrado a partir da fórmula:

$$c = M[x \bmod a][y \bmod b]$$

Esse método foi aplicado com as máscaras $h1$ e $h2$, que foram apresentadas na seção III. TRABALHO PROPOSTO.

A técnica de pontilhamento com difusão de erros é aplicada diferente. Para cada pixel, seu valor é transformado para 0 ou 255, dependendo de seu valor mais próximo. Após, a diferença e do valor final e do valor inicial do pixel p é guardado e distribuído para os pixels adjacentes da seguinte forma:

$$e * \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7/16 \\ 3/16 & 5/16 & 1/16 \end{bmatrix}$$

Neste caso, o modo que a imagem é percorrida influencia na imagem resultante, dado que a distribuição dos erros acontecerá de forma distinta.

Para as duas técnicas, a imagem binária final é invertida, dado que a extensão .PGM (extensão das imagens de entrada) reconhece as cores 0 como branco e 1 como preto, ao contrário da extensão .PBM.

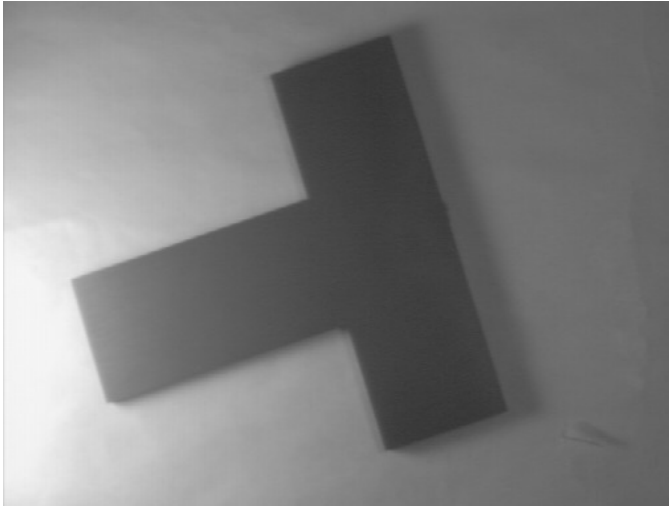
VII. RESULTADOS E DISCUSSÃO

A. Imagens-referências

Para comparação das técnicas utilizadas, usaremos a Figura 1 como base para todos os experimentos.



(a) Lena

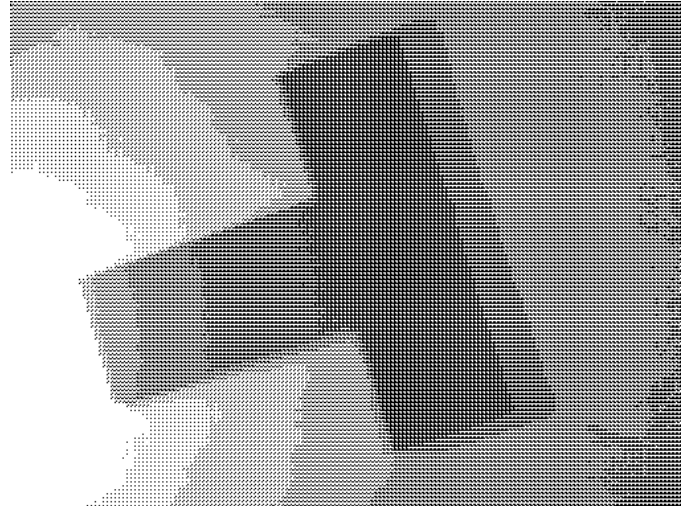


(b) Wedge

Figura 1. Figuras originais que servirão de base para comparações.



(a) Lena



(b) Wedge

Figura 2. Figura com $h1$ aplicado

B. Pontilhado ordenado

O pontilhado ordenado se deu por meio da aplicação das duas matrizes $h1$ e $h2$, e geraram os resultados apresentados nas figuras 2 e 3.

A vantagem do uso dessas matrizes é que o valor de cada pixel só depende de ele mesmo. Dessa forma, é possível trabalhar com a imagem em blocos. Por outro lado, como podemos perceber, a informação fica dependente de blocos do tamanho da matriz, o que traz um efeito hachurado para a imagem. Também é restringida a uma paleta de cores (ou escalas de cinza) paleta uniforme.

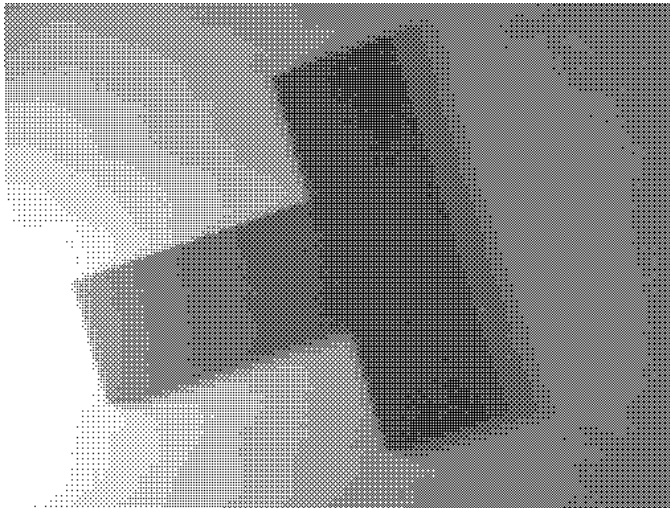
Esse método é uma grande vantagem em contextos de

animação e frames, pois reduz a diferença entre os frames de uma animação.

Já considerando a diferença a aplicação das matrizes $h1$ e $h2$, temos as figuras 2 e 3. A figura 2 se baseia numa matriz 3×3 , que é capaz de produzir 10 padrões diferentes. Esse fato implica em uma escala de cinza menor do que a vista na figura 3, em que a matriz 4×4 produz 17 padrões diferentes. Isso faz com que a figura 3 com a matriz de Bayer seja mais atrativa visualmente e contenha mais detalhes do que a figura 2 com a matriz 3×3 .



(a) Lena

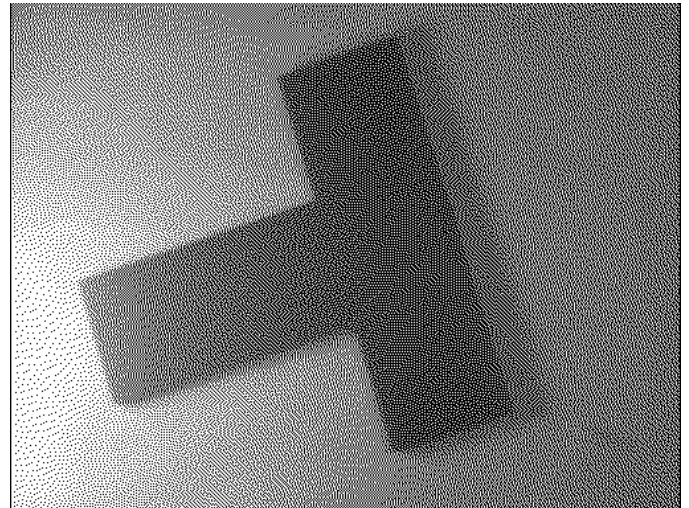


(b) Wedge

Figura 3. Figura com $h2$ aplicado (Bayer)



(a)



(b)

Figura 4. Figura com difusão de erros

C. Pontilhado por difusão de erros de Floyd-Steinberg

A aplicação do método de pontilhado por difusão de erros resultou nas imagens apresentadas na Figura 4.

Diferente do método de pontilhamento ordenado, cada pixel influencia nos próximos pixels a serem analisados, dado que há a distribuição de erros. Assim, uma modificação em um único pixel pode causar um efeito em grande escala.

A vantagem desse método, como podemos ver, é que ele produz resultados com menos conteúdo de baixa-frequência. Isso faz com que a imagem contenha mais detalhes e seja mais atraente para a visão humana, ou seja, traz mais qualidade visual ao resultado. Com esse método, também é possível

explorar paletas de cores otimizadas e não uniformes.

VIII. RESULTADOS

A implementação da comparação dos métodos de pontilhado foi satisfatória, dado que trouxe os resultados visuais esperados e evidenciou as diferenças entre o pontilhado ordenado e a difusão de erros.

+-----+

REFERÊNCIAS

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson, 2017. 1
- [2] H. Pedrini, "Aula: Realce." [Online]. Available: https://www.ic.unicamp.br/~helio/disciplinas/MC920/aula_realce.pdf 1

- [3] “Ordered dither and error diffusion dither.” [Online]. Available: <http://collaboration.cmc.ec.gc.ca/science/rpn/biblio/ddj/Website/articles/CUJ/1998/9812/riemersm/sidebar1.htm> 1
- [4] H. Pedrini, *trabalho 2*. [Online]. Available: <http://www.ic.unicamp.br/~helio/disciplinas/MC920/trabalho2.pdf> 1