

Meios-tones

Introdução ao Processamento de Imagem Digital

Percy Maldonado Quispe
203968

1 Introdução

A técnica de meios tons é conhecida para reduzir a quantidade de cores. Esta consiste em pontilhar a imagem em preto e branco para substituir os níveis de cinza, no caso que seja uma imagem colorida, a técnica se aplica em cada canal de cor. Neste caso, faz-se uso do método de pontilhado com difusão de erros, estes erros vistos em classes, como os que se mostram na tabela 1.

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

(a) Floyd e Steinberg

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

(b) Stevenson e Arce

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

(c) Burkes

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

(d) Sierra

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

(e) Stucki

		$f(x, y)$	7/48	5/48
2/48	5/48	7/48	5/48	3/48
1/48	3/48	5/48	3/48	1/48

(f) Jarvis, Judice e Ninke

Table 1: Formas de distribuição de erro em diferentes técnicas de meios-tones com difusão de erro.

Para a aplicação dessas distribuições de erros, pode-se utilizar os filtros anteriores como máscaras, o resultado é uma nova matriz com pontos preta e branca. A difusão do erro é aplicada à imagem original (em cada pixel $f(x, y)$), então o erro se acumula e interfere diretamente na decisão de se o

pixel na nova imagem ($(g(x, y))$) é branco ou preto. O objetivo do projeto é explorar a técnica de meio-tom com diferentes distribuições de erros.

2 Características do Programa

O programa foi realizado em linguagem Python, com bibliotecas externas NumPy e OpenCV. Os argumentos são passados na execução via *argv*, permitindo modificar várias condições de execução.



Figure 1: Numpy, Python e OpenCV

Os argumentos do programa podem ser:

- *in_image*: Imagem para aplicar a técnica.
- *in_distr*: Nome da distribuição de erro, ou “all” para executar todas as distribuições.
- *channel*: Indica em qual canal a técnica será aplicada.
- *-zigzag*: Se o caminho é alterno (opcional).

3 Implementação

Para a implementação foi seguido o algoritmo de Floyd e Steinberg, apresentando em classes [Pedrini, 2022], porém o código varia no uso da distribuição de erros, que neste caso são as máscaras presentes na tabela 1. O tipo de imagem que eles inserem também é levado em consideração, se é colorida en ‘BGR’, ‘HSV’ ou em tons de cinza. A leitura de acordo com o tipo de imagem é feita através do OpenCV.

Então a imagem é percorrida (de acordo com o valor de *-zigzag*), atribuindo intensidades de 0 aos pixels com intensidade menor que 128 e 255 aos demais, as máscaras também são invertidas quando a varredura é oposta [Graphics, 2004]. Quando estamos no pixel (x, y) , aplica-se a máscara de distribuição de erro, multiplicada pela diferença de erro ($f(x, y) - g(x, y)$). Para o cálculo da máscara, a operação foi vetorizada, utilizando um *slice* (utilizando índices) da matriz original multiplicado pelo erro calculado. Para que isso fosse feito, o controle de índice escolhido foi criar uma borda (preenchimento) e percorrer apenas a imagem original, adicionando metade do tamanho da distribuição de erros nos lados esquerdo e direito, bem como na parte inferior. Para recuperar a imagem de saída, removemos essas bordas.

4 Resultados e Comparações

Os resultados foram obtidos para 4 imagens diferentes (n), com 6 distribuições de erros diferentes (m), também para essas imagens foram tomadas como imagens em tons de cinza, em ‘BGR’ e ‘HSV’ de acordo com o OpenCV, dando um total de 6 imagens de saída (p) para cada distribuição de erro. Ao todo o número de imagens de saída é: $n * m * p$, onde n é o número de imagens, m o número de filtros e p o número de combinações pelos argumentos de entrada, dando 144 imagens de saída. As imagens de entrada estão disponíveis em https://www.ic.unicamp.br/~helio/imagens_coloridas/.

4.1 Comparação entre imagem original

Para comparar a técnica de meio-tons, usaremos a imagem *baboon* [2a](#) em cada distribuição de erro com um caminho em zigzag, os resultados são encontrados na imagem [2](#).



Figure 2: Resultado da aplicação de meios-tons com cada distribuição de erro

A diferença entre a imagem original e as resultantes é perceptível, principalmente ao ampliar a imagem, onde você pode ver o efeito de meio-tons. Os pontos brancos são mais evidentes nas novas imagens. No entanto, outra comparação interessante seria aplicar a técnica em imagens que já estão em tons de cinza.

4.2 Comparação entre varredura

Para a implementação deste projeto, foram considerados dois tipos de varreduras e zigzag unidirecional, conforme mostrado na imagem 3, para apreciar um pouco melhor a diferença entre as varreduras, foi feito com imagens em tons de cinza de *baboon*, os resultados da aplicação da distribuição Stucki estão na imagem 4.

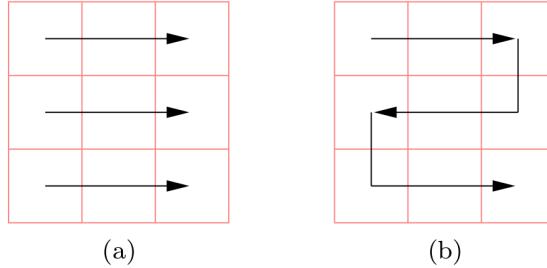


Figure 3: Formas de varredura da imagem.

A diferença observada nas figuras é muito sutil, mas pode-se perceber que o varredura unidirecional direciona o pontilhado, então é possível ver certas linhas brancas que não estão presentes na imagem obtida pelo caminho em zigzag. Um exemplo disso pode ser visto no meio dos olhos do *baboon* 5, onde pode ser visto que o caminho unidirecional gera linhas brancas na direção em que viaja, o que não é muito pronunciado em o caminho em zigzag.

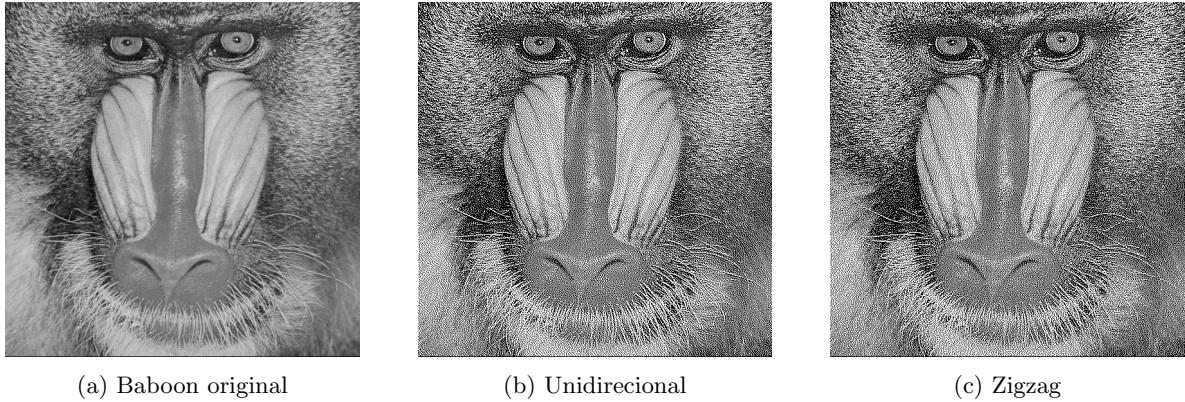


Figure 4: Imagem tons de cinza de *baboon* após a aplicação da técnica de meio-tonos com varredura unidirecional e zigzag.

Ou seja, as diferenças entre as varreduras são sutis, mas ainda presentes, que podem ser apreciadas em imagens menores, ou que apresentam características diferentes.

4.3 Comparação entre distribuições de erro

Para perceber mais claramente a diferença entre as distribuições, a comparação foi feita com as imagens coloridas como entrada em tons de cinza, os resultados em cada distribuição de erro são dados na figura 7.

4.3.1 Imagem monocromática (tons de cinza)

Nesta figura, a diferença entre as distribuições de erro é um pouco mais aparente, graças à forma como os pontos brancos são distribuídos na imagem, afetando a sutileza do pontilhado. A distribuição que mais desvia a imagem do original é a 7b (Stevenson e Arce). Além disso, devido ao fato de reduzir o nível de cor para um binário, torna a diferença mais evidente, por isso não é recomendado se o objetivo for reproduzir a imagem original com o maior grau de fidelidade possível. Em vez disso,



(a) Unidirecional



(b) Zigzag

Figure 5: Linhas brancas no trajeto unidirecional

várias das outras distribuições parecem produzir melhores resultados do que 7b, aproximando-se da imagem original. É possível notar que a distribuição de Floyd e Steinberg produziu mais artefatos que as demais, visíveis na imagem 7a.

Outro ponto importante ocorre com a distribuição de Burkes, que parece 'borrar' a imagem, perdendo a nitidez que os demais filtros mantêm. A distribuição Jarvis, Judice e Ninke forma padrões com pontos brancos que lembram linhas 7f. Em geral, a distribuição que apresenta o melhor resultado é a distribuição Stucki 7e. A imagem resultante é nítida, próxima do original e com poucos padrões. A distribuição Sierra 7d também dá resultados muito satisfatórios para esta imagem, mas não melhor do que 7e.

Todas as observações feitas também podem ser encontradas na seguinte imagem 8, que também se encontra em tons de cinza.

4.3.2 Banda V em HSV

Quando tentamos analisar os erros em uma imagem em HSV (6), levamos em consideração o valor da banda V que vai de [0-255], que representa o brilho da imagem, onde também você pode ver claramente que o filtro Stevenson e Arce 9e causa uma imagem muito distante do original, produzindo grandes alterações, porém Stucki e Sierra (9e, 9d) apresenta melhores resultados como em uma imagem monocromática.

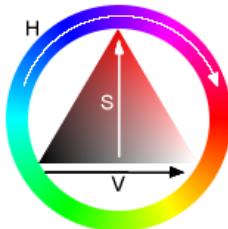
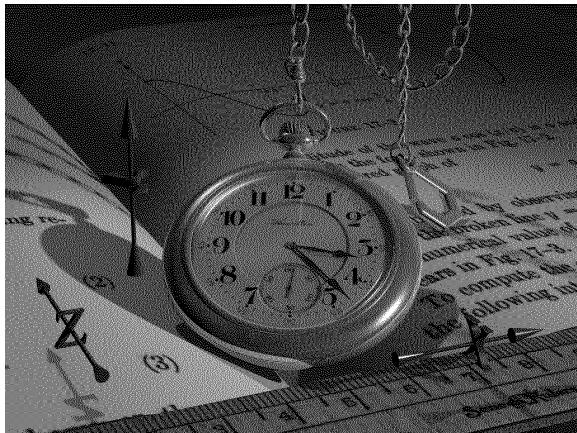
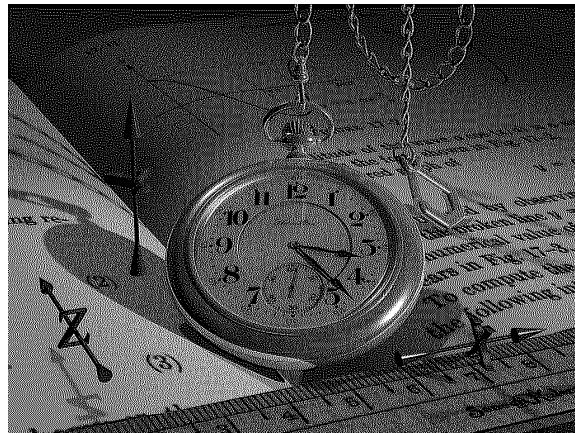


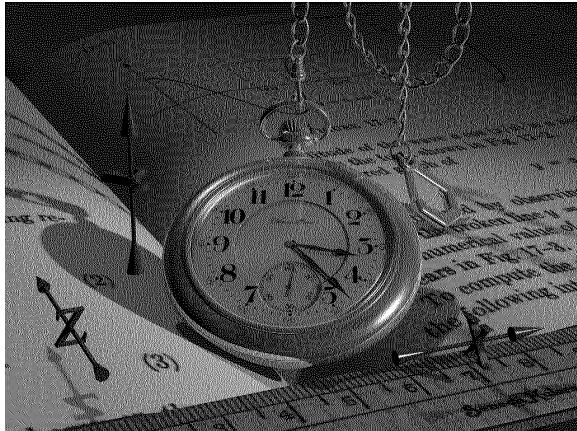
Figure 6: Sistema de cores HSV



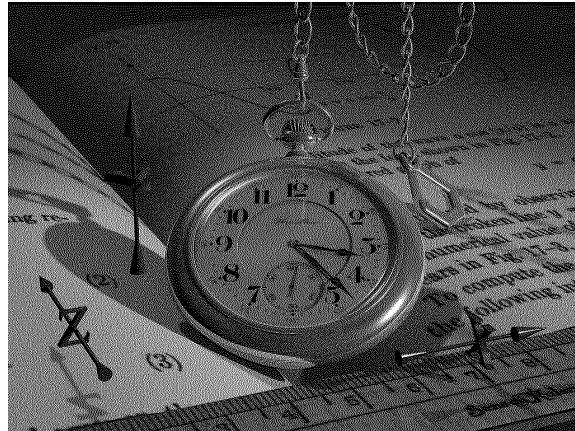
(a) Floyd e Steinberg



(b) Stevenson e Arce



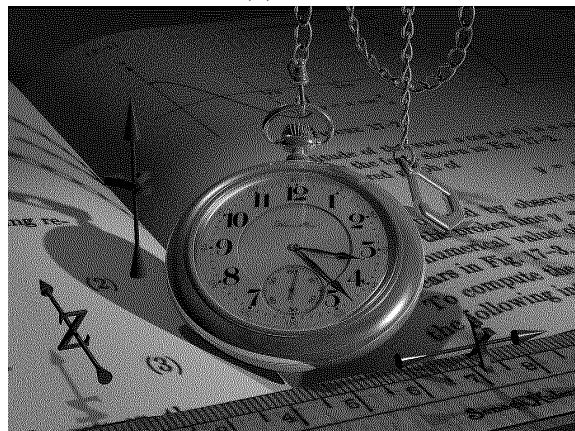
(c) Burkes



(d) Sierra

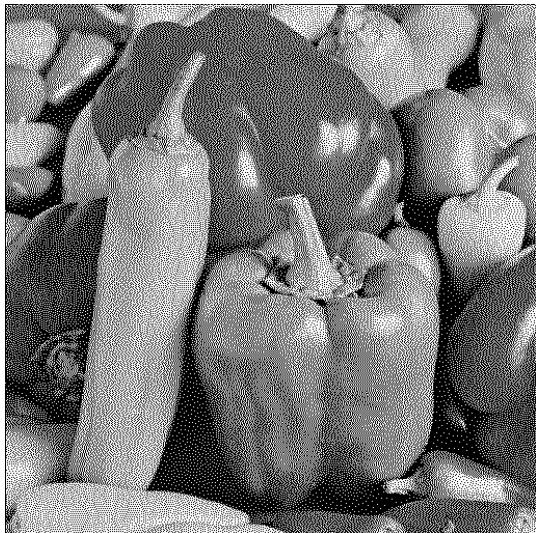


(e) Stucki

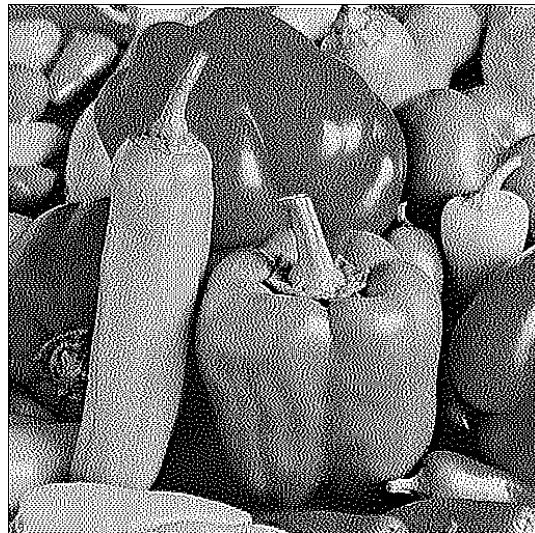


(f) Jarvis, Judice e Ninke

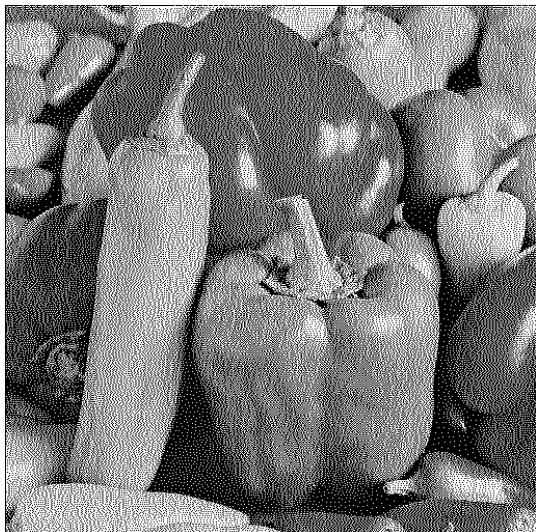
Figure 7: Resultado da aplicação de meios-tons com cada distribuição de erro pra *watch* image



(a) Floyd e Steinberg



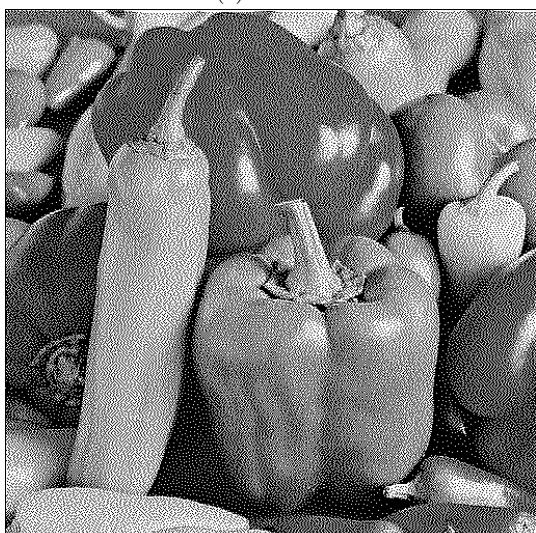
(b) Stevenson e Arce



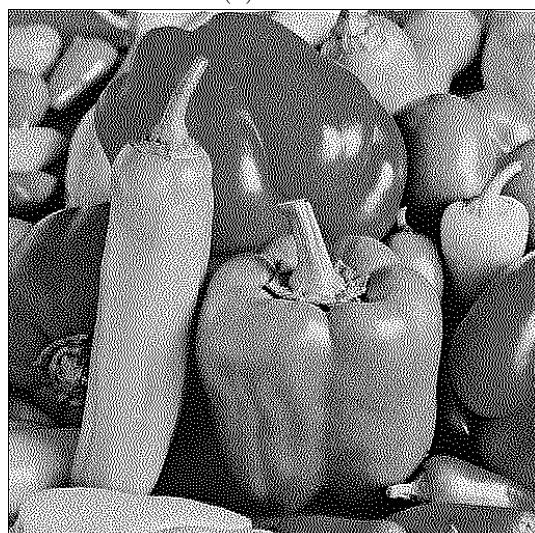
(c) Burkes



(d) Sierra

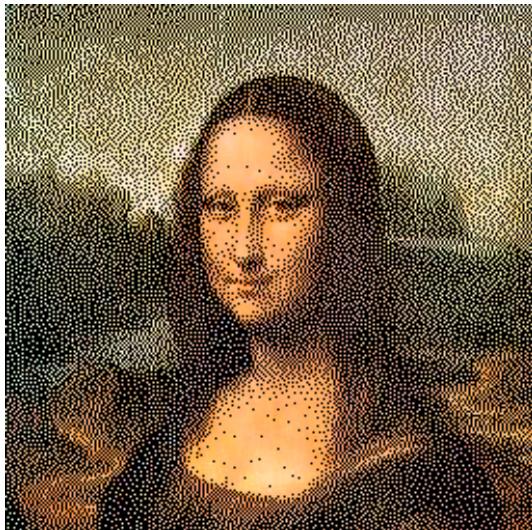


(e) Stucki

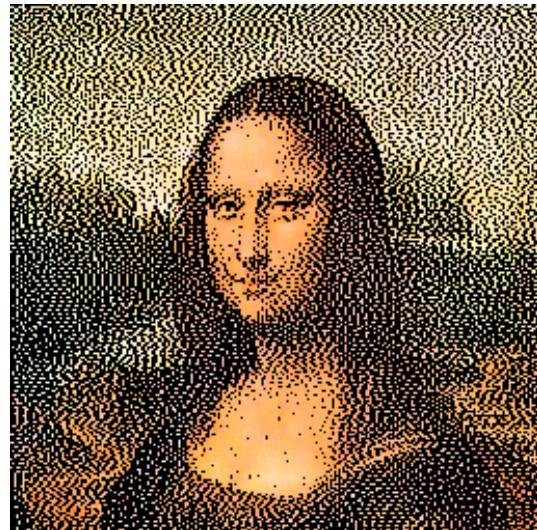


(f) Jarvis, Judice e Ninke

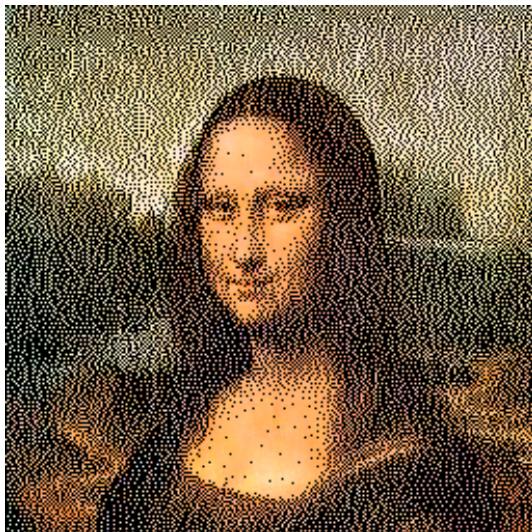
Figure 8: Resultado da aplicação de meios-tones com cada distribuição de erro pra *peppers* image



(a) Floyd e Steinberg



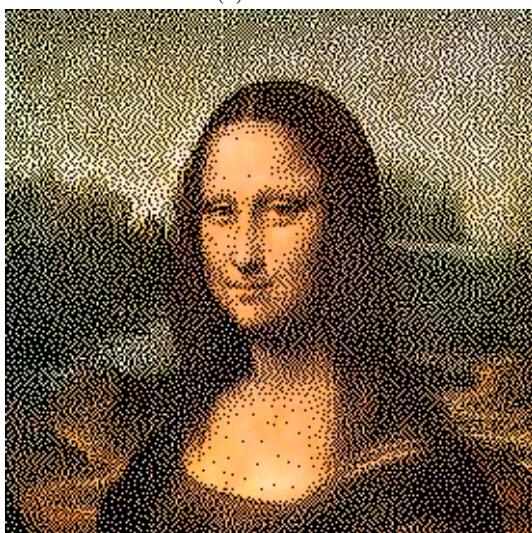
(b) Stevenson e Arce



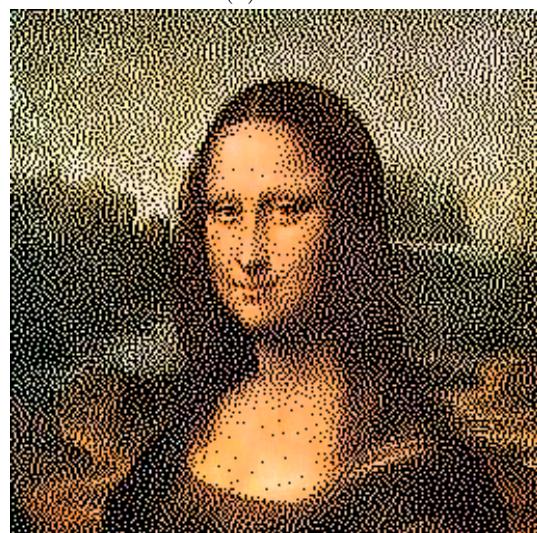
(c) Burkes



(d) Sierra



(e) Stucki



(f) Jarvis, Judice e Ninke

Figure 9: Resultado da aplicação de meios-tones com cada distribuição de erro pra *monalisa* image

5 Conclusão

A técnica de meio-toms tem muitas variações, e cada variação produz resultados diferentes. Essas diferenças podem ser mais ou menos perceptíveis e dependem não apenas da distribuição de erros dada, mas também das características da imagem. Por outro lado, de acordo com a análise entre os 3 tipos de comparações e as imagens, pode-se concluir que:

- A distribuição Floyd e Steinberg produz imagens com artefatos visíveis.
- A distribuição Stevenson e Arce produz imagens distorcidas e granuladas.
- A distribuição Burkes produz imagens desfocadas.
- As melhores distribuições vistas foram Sierra e Stucki.
- As diferenças entre caminhos unidirecionais e alternativos são difíceis de perceber, mas existem e ao ampliar a imagem ficam mais evidentes.

Em imagens coloridas, as diferenças são menos visíveis do que em tons de cinza, devido ao maior número de intensidades. No entanto, é possível afirmar que a execução do projeto foi alcançada de forma satisfatória, isso porque o uso de distribuições de erros em um único código foi generalizado, graças às vantagens que o **NumPy** apresenta.

References

[Graphics, 2004] Graphics, C. C. (2004). Floyd-steinberg dithering. <https://research.cs.wisc.edu/graphics/Courses/559-s2004/docs/floyd-steinberg.pdf>. Accessed: 2022-05-02.

[Pedrini, 2022] Pedrini, H. (2022). Introdução processamento digital de imagem - realce. https://www.ic.unicamp.br/~helio/disciplinas/M0443/aula_realce.pdf. Accessed: 2022-05-02.