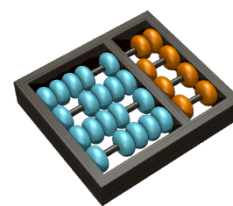




**Universidade Estadual de Campinas
Instituto de Computação**



Disciplina do 2º Semestre de 2020

IC - UNICAMP

Curso: Bacharelado em Ciência da Computação

MC920 - Introdução ao Processamento de Imagens Digitais

Trabalho 2 - Aplicação de Filtros e técnica de Meios-Tons em Imagens Coloridas

Alunos: Gabriel Volpato Giliotti **RA:** 197569

Professor: Hélio Pedrini

Campinas – SP
2020

Trabalho 2 - Introdução ao Processamento de Imagem Digital

Nome: Gabriel Volpato Giliotti - **RA:** 197569

1 - Introdução

O objetivo deste trabalho é realizar a aplicação da técnica de realce de “Meios-Tons” em imagens coloridas, onde máscaras/filtros são aplicados para cada banda de cor de uma imagem no formato .png, além de uma análise sobre o comportamento de cada filtro previamente definido no enunciado.

Atualmente, o processamento de imagens é um artifício muito utilizado em mídias sociais, aparelhos eletrônicos e softwares, para aplicação de filtros e outros recursos em imagens. Assim, os filtros/máscaras nos oferecem diferentes objetivos de aplicações, disponibilizando a recuperação, suavização ou destacamento (realce) de pontos específicos (pixels) em imagens. Com isso, o projeto dois visa introduzir a aplicação de diferentes tipos de máscaras de imagens, oferecendo base para o entendimento de como uma possível técnica de realce pode ser aplicada no processamento de imagens coloridas.

Junto desse relatório será entregue o arquivo .zip ou .tgz, que possui todos os arquivos citados nesse relatório e que foram utilizados para processar as imagens dadas como entrada.

2 - Programas

O script para processamento das imagens foram implementados em Python 3.7.6 junto das bibliotecas Numpy 1.19.2 e OpenCV 4.4.0 que são voltadas para o processamento de imagens. Outras bibliotecas como Math e Sys, que auxiliam no processamento, também foram utilizadas mas não oferecem recursos específicos para processamento de imagens.

2.1 - Como Executar

Inicialmente, antes de executar o script, devemos criar uma pasta chamada “Outputs” que corresponde ao local de saída das imagens da execução do arquivo .py referente ao trabalho 2. Para se obter as imagens de saída, você deve criar a pasta no mesmo diretório onde executará o arquivo .py.

Para o trabalho 2 foi criado um único arquivo com extensão do tipo .py que é responsável pela aplicação das máscaras de realce definidas no enunciado. A seguir está o formato de construção para chamada do arquivo via linha de comando (Windows: CMD/ Linux: Bash):

C:\> python halftoning.py “pathImagem” “nomeFiltro” “percurso”

Substitua “pathImagem” pelo caminho (path) correto da imagem à qual se deseja aplicar uma máscara. Substitua “tipoFiltro” para escolha do tipo de filtro que será aplicado e para o percurso, escolha por “Direto” ou “Alternado” (Todos os parâmetros sem aspas). Os possíveis nomes dos filtros são:

- “Floyd_Steinberg”
- “Stevenson_Arce”
- “Burkes”
- “Sierra”
- “Stucki”
- “Jarvis_Judice_Ninke”

2.2 - Entradas

Como entrada para o arquivo .py é necessário o path da imagem a ser processada, além da definição do tipo de filtro a ser aplicado e o tipo de percurso que deve ser feito sobre a imagem para aplicação da máscara. Todas as imagens devem ser coloridas, no formato PNG (Portable Network Graphics). As imagens utilizadas para esse projeto vieram de um repositório indicado pelo professor e estão compactadas no .zip ou .tgz junto ao arquivo .py em uma pasta chamada ColorImages.

2.3 - Um pouco sobre o Algoritmo

Para a execução do algoritmo, fazemos a leitura da imagem colorida e, para o método slide_filter (Similar ao dithering), enviamos cada banda de cor da imagem (BGR), uma por vez, onde a máscara selecionada é aplicada, e realizamos a propagação de erro sobre a imagem original para as iterações seguintes. Além disso, existem dois métodos, halftoning_direto e halftoning_alternado, que são chamados de acordo com um parâmetro dado na entrada. O Direto aplica a máscara na imagem no sentido usual, da esquerda para a direita, do topo para a base da imagem. Já o Alternado, aplica os filtros em zig-zag, ou seja, a primeira linha o filtro é aplicado da esquerda para direita, na linha de baixo, a máscara sofre um flip, e é aplicada da direita para esquerda, e esse processo se repete até o final da imagem. (Para entender melhor, acesse o arquivo halftoning.py e veja os comentários).

2.4 - Saídas

As saídas são obtidas na pasta “Outputs” criada previamente, antes da execução do arquivo. Todas as imagens de saída serão coloridas, no formato PNG (Portable Network Graphics) e com a mesma resolução de pixels da imagem de entrada. Além disso, por causa do modo como a filtragem é aplicada sobre a imagem (sem a utilização de bibliotecas que auxiliam o processamento), o tempo para gerar uma imagem de saída tem baixa velocidade, fazendo

com que o tempo de aplicação dos filtros sejam de até 40 segundos para uma dada imagem (O tempo pode variar de acordo com a resolução da imagem de entrada e de acordo com o filtro a ser aplicado).

3 - Parâmetros Utilizados

Os parâmetros utilizados para esse projeto são o path da imagem a ser processada, o tipo de filtro a ser aplicado e o percurso de aplicação do filtro sobre a imagem. Todos esses parâmetros estão citados previamente neste relatório.

4 - Soluções

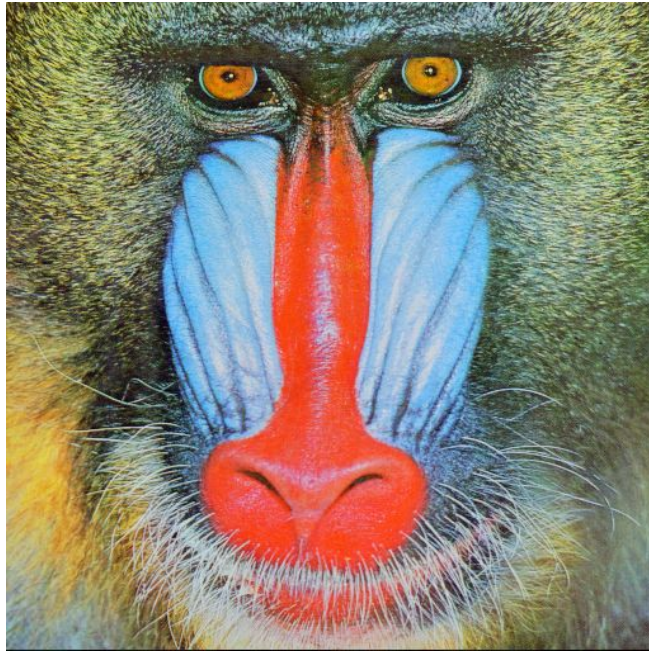
A seguir estão as comparações e observações da aplicação de cada filtro com sua correspondente imagem original. A escolha das imagens foram aleatórias, na tentativa de melhor se apresentar os resultados obtidos. Além disso, tentamos apresentar as aplicações de forma mais abrangente possível.

4.1 - Floyd e Steinberg

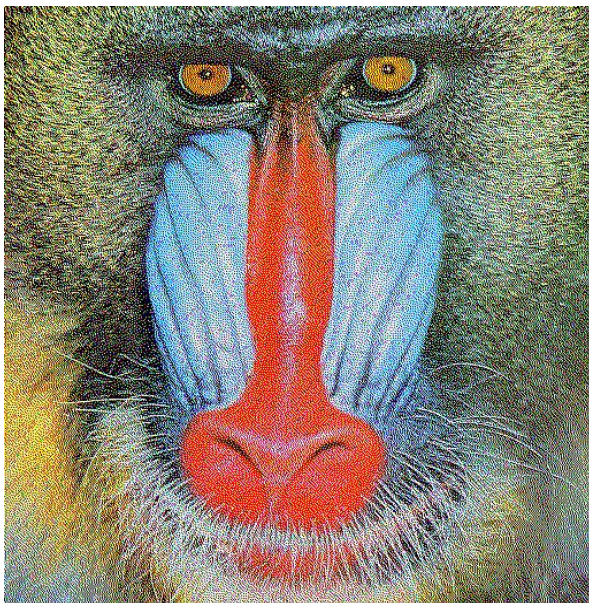
Inicialmente, o filtro de Floyd e Steinberg cria a ilusão de uma imagem com várias tonalidades de cores com pontos de coloração bem distribuídos, quando o filtro é aplicado com um percurso Direto (isso é, da esquerda para a direita, do topo para a base da imagem) sobre cada banda de cor da imagem original. Já na aplicação Alternada (Em zig-zag sobre cada linha de pixels da imagem), o filtro cria pequenos segmentos de coloração iguais, isso é, maior numero de pixels consecutivos com mesma cor, o que causa um clareamento e uma melhor realce de cores sobre determinadas regiões da imagem. Observe a diferença de brilho entre as cores mais evidentes, como as cores azuis, vermelhas e amarelas:

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

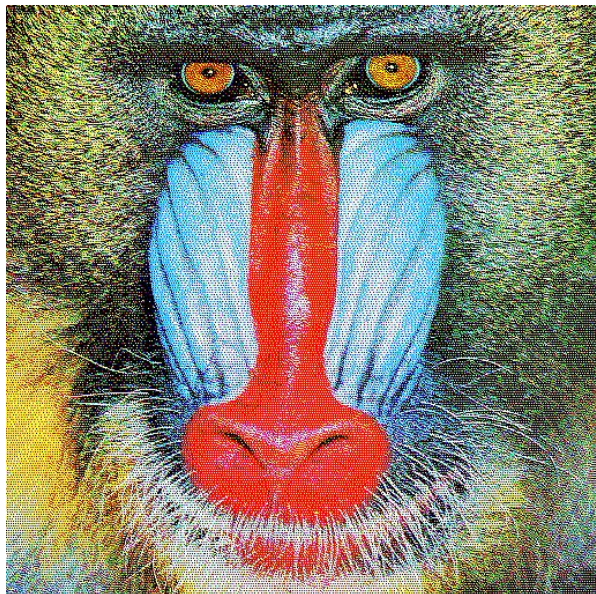
Floyd e Steinberg



Original baboon.png



Floyd e Steinberg Direto baboon.png



Floyd e Steinberg Alternado baboon.png

4.2 - Stevenson e Arce

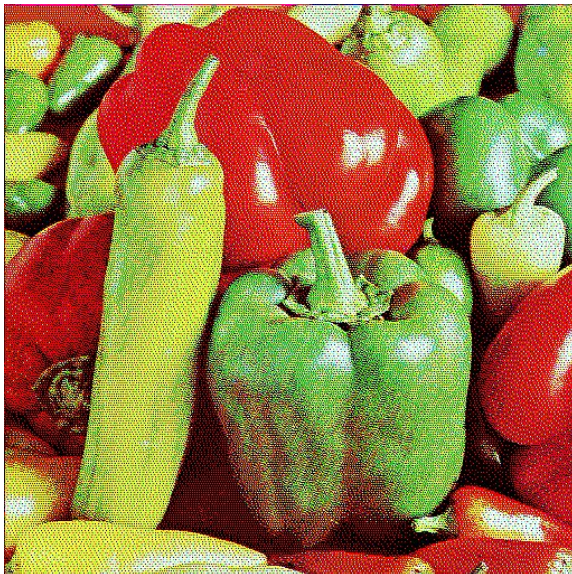
Em seguida, o filtro de Stevenson e Arce parece criar um aspecto granular maior que o filtro de Floyd e Steinberg para os pixels das imagens dadas na entrada. Outro ponto a se reparar é que em imagens com detalhes de sombras, o filtro de Stevenson e Arce parece não recriar tão bem regiões de sombra com sua “granularidade”. Observe a diferença entre as regiões com sombras do percurso Alternado sobre a imagem a seguir:

			$f(x, y)$		32/200	
12/200		26/200		30/200		16/200
	12/200		26/200		12/200	
5/200		12/200		12/200		5/200

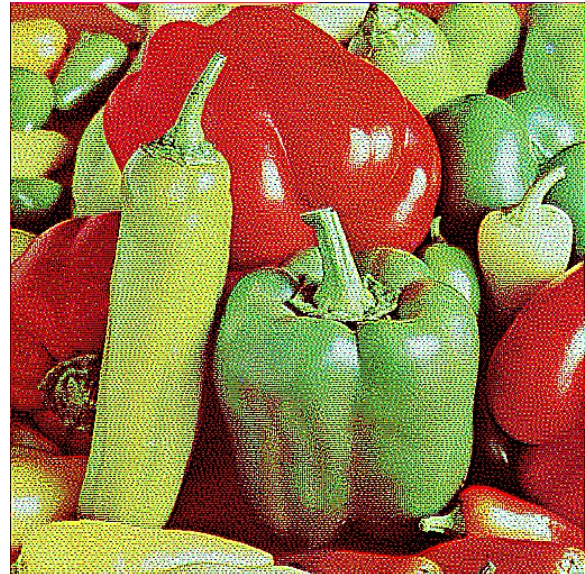
Stevenson e Arce



Original peppers.png



Peppers.png Floyd e Steinberg (Alternado)



Peppers.png Stevenson e Arce (Alternado)

4.3 - Burkes

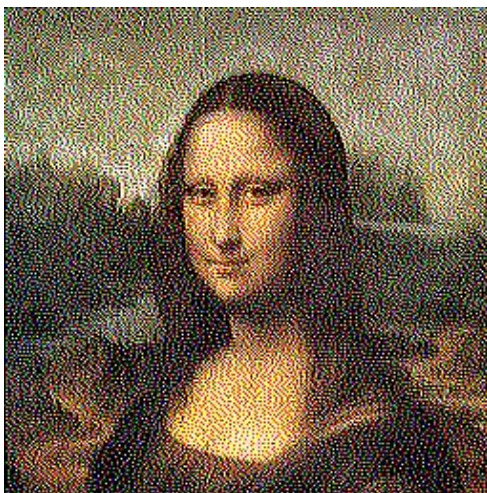
Para o filtro de Burkes, para qualquer imagem que o aplicamos, o conjunto de pixels da imagem de saída parece recriar a ilusão manchas bem definidas na imagem, isto é, sombras, formas fundos parecem ficar melhor arredondados e restritos em áreas específicas nas imagens de saída. Porém, detalhes de sombras presentes da imagem original, são ocultados por outras cores.

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

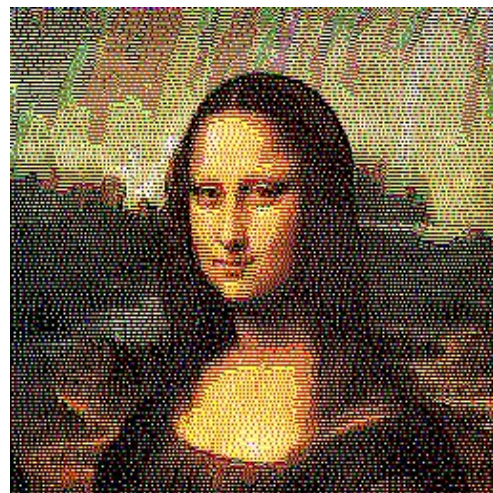
Burkes



Monalisa.png Original



Monalisa.png Burkes Direto



Monalisa.png Burkes Alternado

4.4 - Sierra

O filtro Sierra aparenta apresentar uma ilusão onde os formatos e sombras da imagem original parecem ter sido mantidos com certa fidelidade. Entretanto, a imagem de saída parece apresentar uma construção de linhas de cores em diagonal, isto é, pequenas linhas de pixels de mesma cor, em diagonal parecem estar formando a imagem. Tal comportamento é possivelmente mais visível em regiões de sombras ou cantos das imagens de saída.

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

Sierra



watch.png Original



Watch.png Burkes Alternado



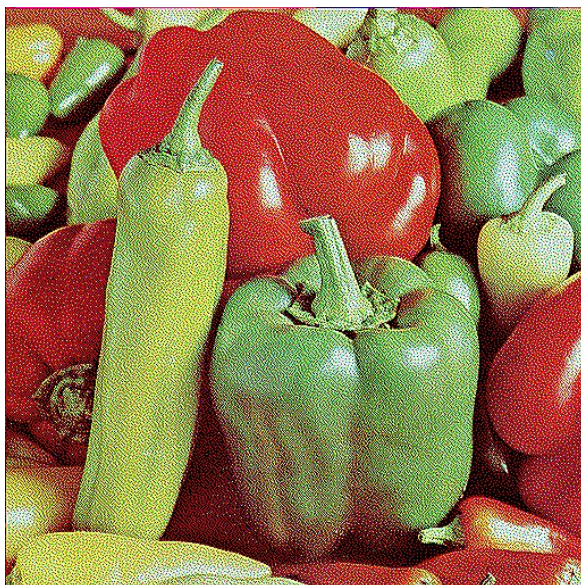
Watch.png Burkes Direto

4.5 - Stucki

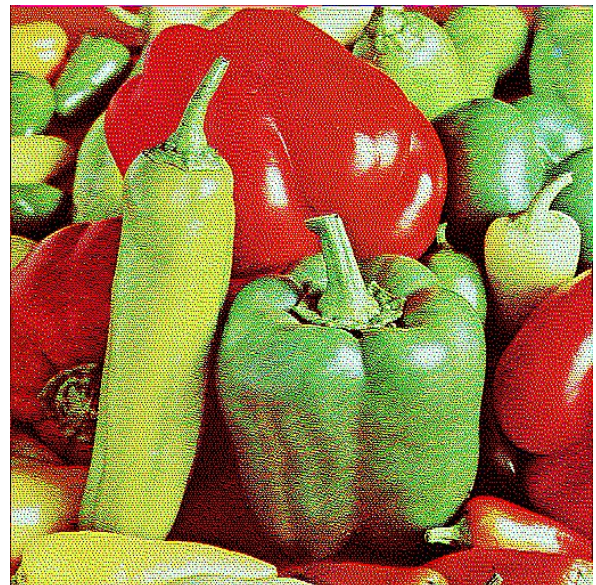
O filtro de Stucki não apresentou diferenças muito significativas com relação ao filtro de Sierra, logo, temos que existe aqui uma preservação de sombras e contornos para esse filtro também. Diferente do filtro de Sierra, as linhas de pixels geradas pelo filtro de Stucki parecem estar em um sentido mais horizontal que diagonal.

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

Stucki



Peppers.png Stucki (Direto)



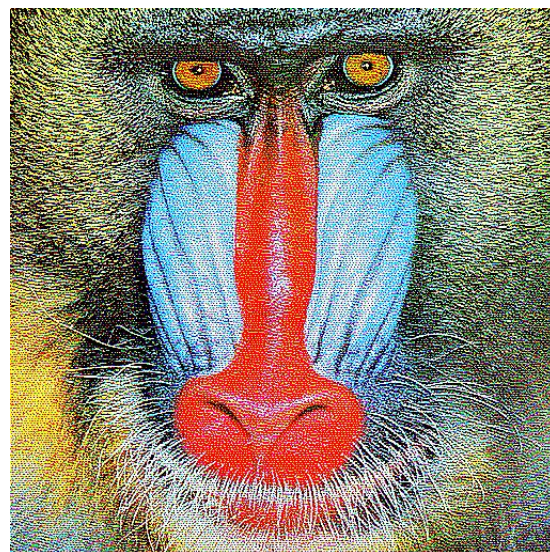
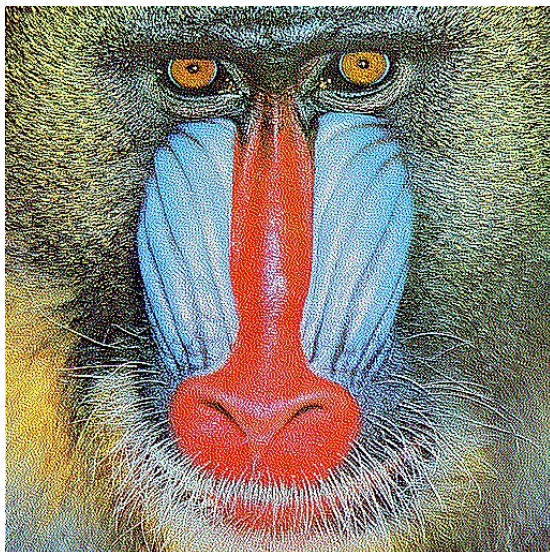
Peppers.png Stucki (Alternado)

4.6 - Jarvis, Judice e Ninke

Finalmente, o filtro de Jarvis, Judice e Ninke também não apresentou mudanças muito significativas com relação aos filtros Stucki e Sierra, ou seja, os 3 filtros apresentam comportamentos similares. Neste, as linhas de pixels geradas pelo filtro parecem estar em um sentido mais horizontal que diagonal também.

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

Jarvis, Judice e Ninke



Jarvis, Judice e Ninke Direto baboon.png Jarvis, Judice e Ninke Alternado baboon.png

5 - Conclusão

Concluindo, devemos ressaltar que os filtros foram aplicados para mais de uma imagem. em diferentes modos (Direto e Alternado) e este relatório apresenta características observadas para a maioria das imagens. Além disso, originalmente a técnica de meios tons é melhor aplicável para imagens monocromáticas, ou seja, os resultados do comportamento dos filtros tem maior facilidade de observação com imagens de entrada em tons de cinza, por isso, imagens com tons brancos, pretos e sombras (mesmo sendo coloridas) podem ajudar na comparação. Enfim, com o Projeto 2 obtivemos contato com a técnica de realce de imagens de meios tons e como aplicá-la em imagens coloridas, aprendendo a manipular as bandas BGR, percorrendo a imagem em diferentes percursos, observando seu efeito e aprendendo onde tal técnica e filtros são melhor aplicáveis. (No caso da observação ser difícil com as imagens expostas nesse relatório, execute o arquivo entregue junto a esse relatório, seguindo os passos apresentados no início deste documento)