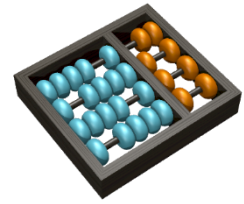




**Universidade Estadual de Campinas
Instituto de Computação**



Disciplina do 2º Semestre de 2020

IC - UNICAMP

Curso: Bacharelado em Ciência da Computação

MC920 - Introdução ao Processamento de Imagens Digitais

Trabalho 3 - Aplicação de Filtros e técnica de Thresholding em Imagens Monocromáticas

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

Professor: Hélio Pedrini

Campinas – SP
2020

Trabalho 3 - Introdução ao Processamento de Imagem Digital

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

1 - Introdução

O objetivo deste trabalho é o estudo e realização da limiarização de imagens, uma técnica de segmentação onde máscaras/filtros são aplicados em imagens no formato .pgm monocromáticas (tons de cinza) para identificação de objetos e bordas para reproduzir uma imagem de saída binária, dividida entre objeto (na cor preta) e fundo (na cor branca), 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 abordagens de aplicações, disponibilizando a recuperação, suavização, captura de objetos/planos e 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 abordagem de segmentação pode ser aplicada no processamento de imagens monocromáticas.

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, Sys, Matplotlib, que auxiliam no processamento de gráficos, aplicação de funções (para gerar histogramas) e operações (com exponenciais e logs), 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 3. Para se obter as imagens de saída e os histogramas das imagens de entrada, você deve criar a pasta no mesmo diretório onde executará o arquivo .py.

Para o trabalho 3 foi criado um único arquivo com extensão do tipo .py que é responsável pela aplicação das máscaras de limiarização definidas no enunciado. A seguir está o formato de construção para chamada do arquivo via linha de comando (Windows: CMD/ Linux: Bash):

```
C:\> python limiarizar.py "pathImagem" "nomeMetodoLimiarizacao"
```

Dados os métodos limiarização no enunciado, temos que a execução do comando acima apresenta valores default de parâmetros que são utilizados nos métodos disponíveis. Os valores default dos parâmetros adicionais são apresentados a seguir:

- thresh = 128 (Utilizado apenas no método global)
- size = 5 (Utilizado em todos os métodos locais)
- k = 0.25
- r = 0.5
- p = 2
- q = 10

Logo, é possível especificar diferentes valores para esses parâmetros, passando-os como argumento na execução do arquivo .py. Para especificar novos valores execute o método desejado como mostrado nos exemplos a seguir:

- C:\> python limiarizar.py "pathImagem" global --thresh 100
(Especifica o valor de thresh como 100 para o método global)
- C:\> python limiarizar.py "pathImagem" bernsen --size 7
- C:\> python limiarizar.py "pathImagem" more --size 9 --r 4 --q 11

Substitua "pathImagem" pelo caminho (path) correto da imagem à qual se deseja aplicar uma máscara. Além disso, substitua também "nomeMetodoLimiarizacao" para escolha do tipo de filtro que será aplicado. Os possíveis filtros estão listados a seguir. Observe que ao lado do método indicamos quais parâmetros adicionais afetam a aplicação do método escolhido:

- "global" → --thresh
- "bernsen" → --size
- "niblack" → --size, --k
- "sauvola" → --size, --k, --r
- "more" → --size, --k, --r, --p, --q
- "contrast" → --size
- "mean" → --size
- "median" → --size

Obs: Valores de parâmetros não especificados ficam como default.

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, se desejado um ou mais novos valores para os parâmetros adicionais, de acordo com o requisitado por cada método. Todas as imagens devem ser monocromáticas, no formato PGM (Portable Gray Map). 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 ImgsMono.

2.3 - Um pouco sobre o Algoritmo

Para a execução do algoritmo, fazemos a leitura da imagem tons de cinza e, para o método global, realizamos a limiarização para imagem toda, onde valores menores ou iguais que 128 recebem 0 e se tornam objetos, e valores maiores que 128 recebem 255, e se tornam fundo. Por outro lado, nos métodos locais, percorremos a imagem em blocos, de forma similar à uma convolução, onde definimos o filtro desejado, seu tamanho e, se necessário, valores adicionais correspondentes ao método escolhido. Além disso, as imagens geradas para a saída tem valores binários (0 ou 255), sendo colocadas em um novo objeto e um histograma informando a quantidade de pixels para cada nível de cinza (entre 0 e 255) é gerado com o nome do método que foi aplicado.

2.4 - Saídas

As saídas (Imagens finais e histogramas das imagens de entrada) são obtidas na pasta “Outputs” criada previamente à execução do arquivo, além da fração de pixels pretos que é apresentada no próprio terminal. Todas as imagens de saída serão monocromáticas, no formato PNG (Portable Network Graphics) e com a mesma resolução de pixels da imagem de entrada. É possível definir para que as imagens de saída saiam em formato PGM (Portable Gray Map). Para isso, procure pelo método salvar_imagem no código e comente as linhas indicadas.

3 - Parâmetros Utilizados

Os parâmetros utilizados para esse projeto são o path da imagem a ser processada, o tipo de filtro a ser aplicado além dos parâmetros adicionais para cada método de limiarização (se desejado). Todos esses parâmetros estão citados previamente neste relatório.

4 - Soluções

A seguir estão as comparações e observações da aplicação de cada filtro com suas correspondentes imagens de saída. 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. Os histogramas das imagens de entrada podem ser encontrados na pasta Outputs.

4.1 - Método Global

Iniciando com o método de limiarização global, podemos observar que é um método que não consegue separar objeto e fundo em imagens com alta precisão. Logo, para que isso aconteça, as imagens não podem apresentar grandes áreas com sombreamento ou claridade. Caso isso ocorra, ao depender dos valores dos pixels da região sombreada toda a área pode ser interpretada como objeto, reproduzindo uma imagem com regiões totalmente pretas (Repare que nas imagens de fiducial, wedge e sonnet temos regiões totalmente pretas devido a sombras dos níveis de cinza), e também regiões muito claras que podem levar o algoritmo a gerar regiões totalmente brancas. Além disso, para essa filtragem não temos um valor de tamanho para vizinhança, e os valores de limiar utilizados estão na descrição das imagens apresentadas. (Valor default sugerido \rightarrow thresh = 128).

Imagem de entrada com pouca ou nenhuma sombra aparente:



baboon thresh = 100

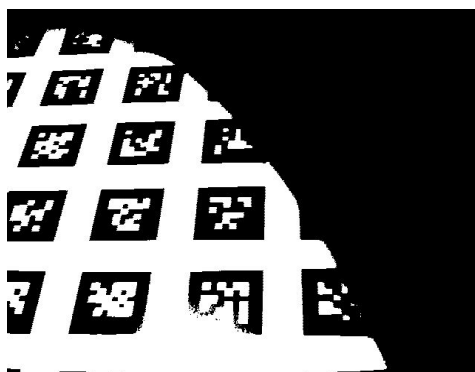


baboon thresh = 128

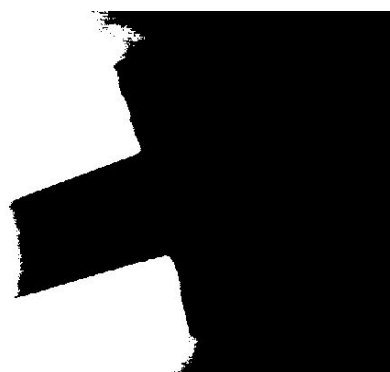


baboon thresh = 160

Imagens de entrada com sombra bastante aparente:



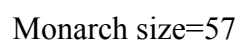
Fiducial thresh = 128



Wedge thresh = 128

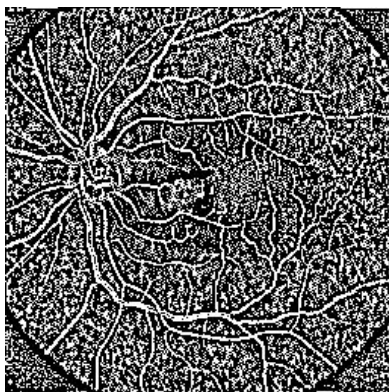
O dear I see
It is hard to see this
I thought the cat
If only your portrait I see
First when I reach my key
And that your cheeks below, so once you
Every hair contains a thousand lines
With sum of discrete curves
And with manual and tactical
And not the proper fractal.

Para o método de Bernsen, podemos observar que o filtro aplicado responde de forma melhor para imagens com sombras, apresentando um comportamento mais satisfatório que o global, não criando regiões pretas e brancas (objetos e fundos) tão grandes quanto do método global. O problema é que ao utilizar um valor de tamanho janela pequeno, na saída são obtidos apenas os contornos dos objetos da imagem original ou talvez uma imagem totalmente branca e, se utilizado um valor de janela muito grande, as sombras podem acabar sendo identificadas como objetos, o que cria a necessidade de testagem exaustiva dos limites para encontrar um valor de tamanho de janela satisfatório para cada imagem, e esse valor pode variar de imagem para imagem. Para esse método os tamanhos de vizinhanças testados foram variados, de 5 (default) até 307, e o valor de limiar utilizado é o valor médio entre o maior e menor valor de pixel da vizinhança.

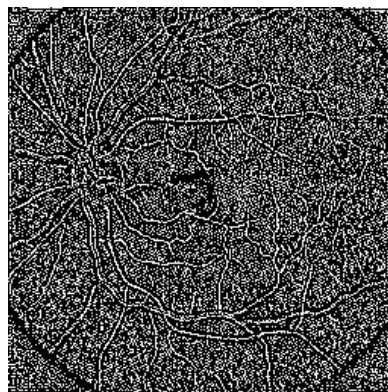


4.3 - Método Local de Niblack

No filtro de Niblack, utilizamos uma imagem da retina, que apresenta bastante sombreamento na imagem original. Variando o tamanho da janela do filtro (size) e o parâmetro k , pudemos observar um bom preenchimento entre os objetos (no caso, os nervos óticos ficaram como fundo e seus contornos ficaram bem aparentes). Assim, para obter resultados satisfatórios, valores de k foram variados entre 0 e 1 (indicados nas imagens a seguir) e os tamanhos de vizinhança foram variados entre 3 e 11 (também indicados nas imagens). Para valores de k maiores que 1, as imagens de saída são apresentadas praticamente preenchidas na cor preta. Portanto, para imagens de saída com melhor visibilidade, a variação do tamanho da janela de vizinhança foi essencial



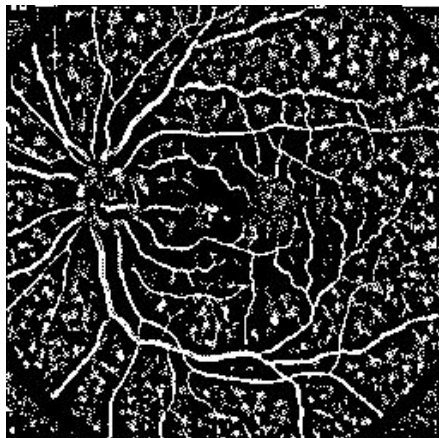
retina size=5 $k=0.25$



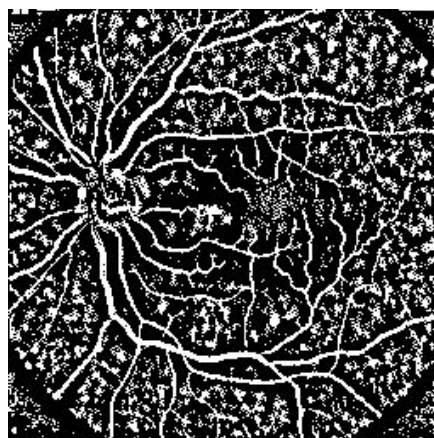
retina size=3 $k=0.25$



retina size=5 $k=0.75$



retina size=11 $k=0.75$



retina size=11 $k=0.5$

4.4 - Método Local de Sauvola e Pietaksinen

O método de Sauvola e Pietikainen, como apresentado no enunciado responde de forma mais eficaz em imagens com iluminação ruim. Vale ressaltar que como não normalizamos a imagem durante a aplicação do filtro, os valores de parâmetros sugeridos não geraram imagens com resultados satisfatórios. Além disso, quando uma imagem de iluminação ruim é dada, a saída para esse filtro apresenta os objetos de forma bastante clara e, quando uma

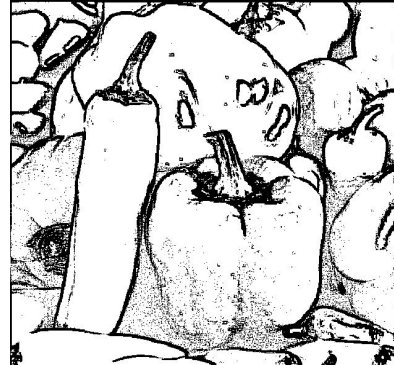
imagem com iluminação melhor é dada como entrada, o método consegue até mesmo identificar regiões com sombreamento e pixelizar, gerando um efeito parecido com uma sombra na imagem de saída. Finalmente, para esse filtro, o valor de $r = 8$ mostrou bons resultados, com $k = 0.25$ fixo e variando os tamanhos de vizinhança entre 5 e 11.



peppers size=5 $k=0.25$ $r=5$



peppers size=5 $k=0.25$ $r=8$



peppers size=5 $k=0.25$ $r=11$



peppers size=7 $k=0.25$ $r=8$



peppers size=11 $k=0.25$ $r=8$

4.5 - Método de Phansalskar, More e Sabale

Para o filtro de Phansalskar, More e Sabale, foram feitos diversos testes, onde variamos os tipos de imagens e os diferentes parâmetros do método. Assim, foi observado que para imagens com pouca sombra ou claridade, os parâmetros recomendados responderam de forma satisfatória para a imagem de saída. Já para imagens com iluminação ruim, um primeiro aspecto observado é que o método possibilita de certa forma controlar a sombra (variando os diferentes parâmetros) para capturar os objetos. Logo, foi observado que variar os parâmetros k e q apresentam resultados similares com relação a variação de outros parâmetros. Com o parâmetro r , foi observado que r 's bem pequenos melhoram a captura de objetos nas imagens e, portanto, se a iluminação for bem ruim, r deve ser bastante pequeno. Os valores de limiar e de janelas utilizadas estão junto das fotos indicadas a seguir.



Baboon size=5 k=0.25 r=2 p=2 q=10



Baboon size=5 k=0.25 r=0.5 p=20 q=10

Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I could impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tarted
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

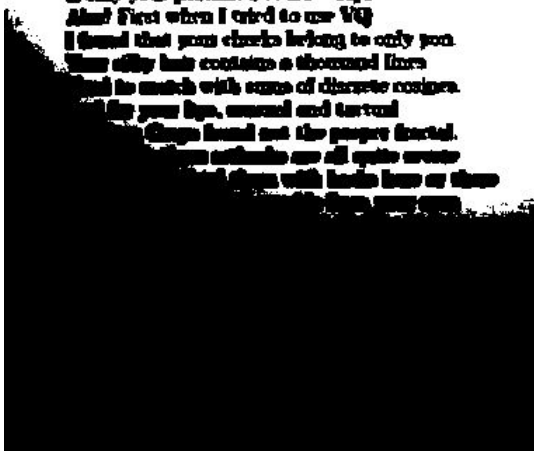
Thomas Coltheart



Sonnet size=5 k=0.25 r=0.05 p=2 q=10

Sonnet for Lena

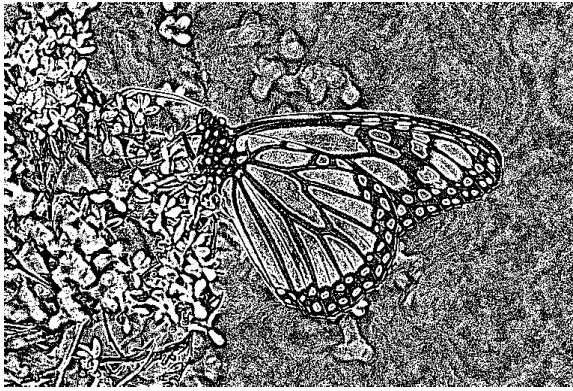
O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I could impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tarted
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'



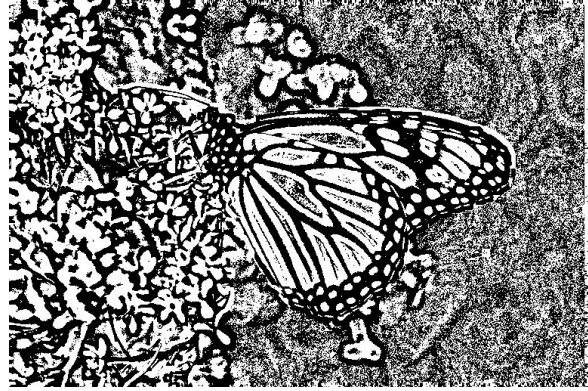
Sonnet size=5 k=0.25 r=0.05 p=15 q=10

4.6 - Método do Contraste

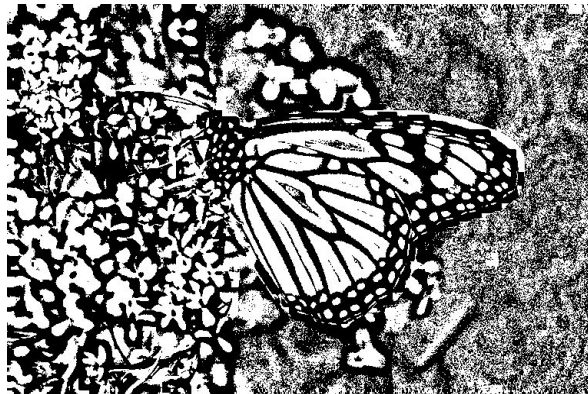
Para o método do Contraste, podemos observar que só é possível variar o tamanho da janela de vizinhança do filtro. Logo, nas imagens de saída são apresentados ruídos por toda imagem, mas também é um método que consegue identificar os objetos e criar uma área branca em torno deles, deixando-os evidentes na saída. Logo, não podemos estabelecer um tamanho de janela muito grande ou muito pequeno, pois existe o risco de criar áreas muito grandes ou muito pequenas em torno de um objeto. Tal comportamento é aparente tanto para imagens com iluminação boa como imagens com iluminação ruim.



monarch size = 5



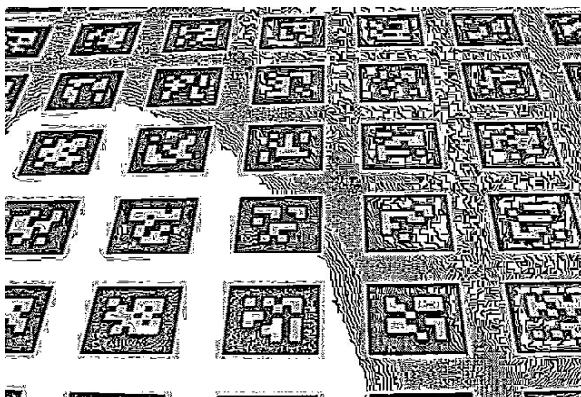
monarch size = 11



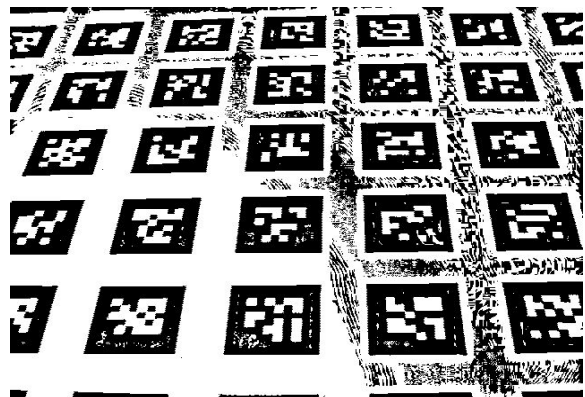
monarch size = 17

4.7 - Método Local da Média

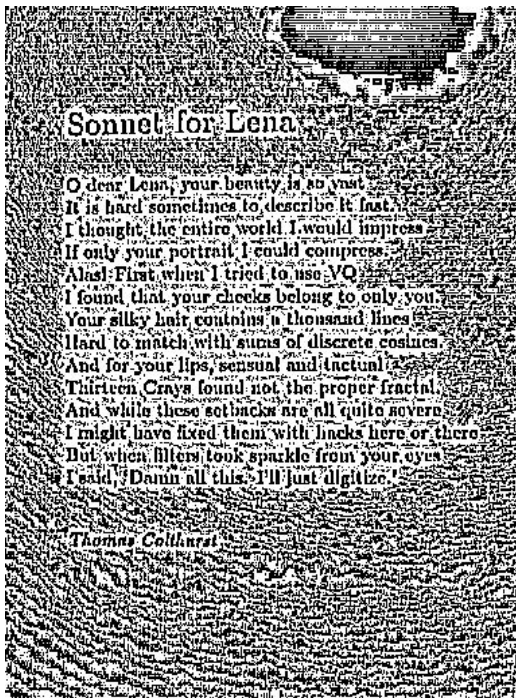
O método da média tem o comportamento similar ao método do Contraste. A diferença nesse caso é que enquanto o contraste espalha ruídos por toda a imagem, tenha ela sombras ou não, o método da média tende a manifestar ruídos somente nas regiões de sombra, enquanto que nas regiões mais claras os objetos são detectados de forma praticamente limpa. Podemos observar também que o tamanho da janela de vizinhança do filtro também afeta o tamanho dos ruídos de sombra e também o preenchimento dos objetos apresentados nas imagens de saída. Esse método também é capaz de criar um contorno branco em volta dos objetos.



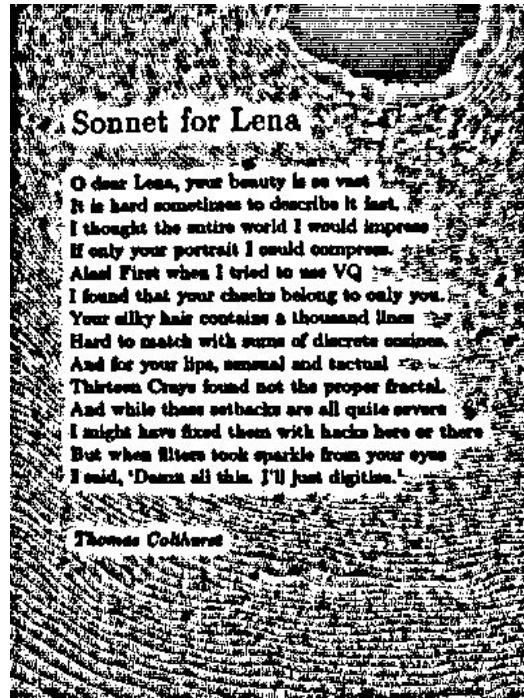
fiducial size=5



fiducial size=17



sonnet size=5



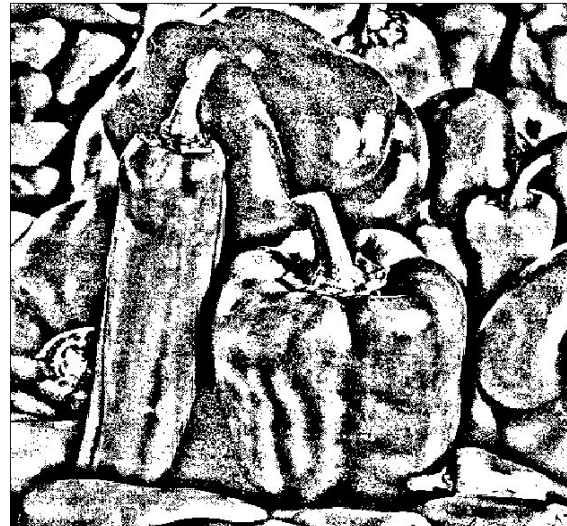
sonnet size=17

4.8 - Método Local da Mediana

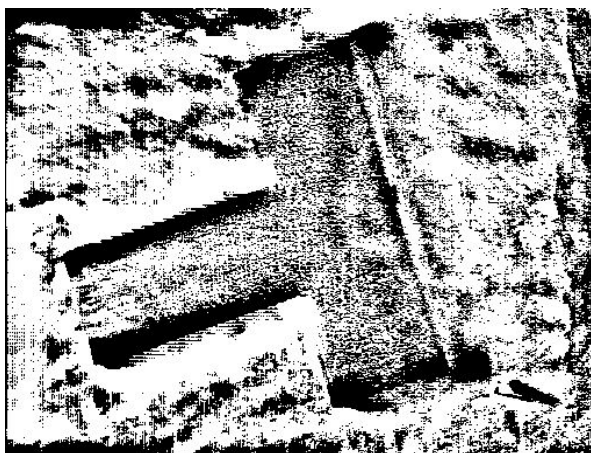
Finalmente, para o método local da mediana temos também que o único parâmetro variável é o tamanho da janela de vizinhança do filtro. Aqui foi observado que método da mediana não responde tão bem para detectar objetos como os métodos apresentados anteriormente. Sua distribuição de ruídos é uniforme por toda imagem e os tamanhos de vizinhança devem ser altos comparados com os métodos anteriores (foram testados variando entre 5 e 65). Outro ponto observado aqui é que locais onde os objetos detectados criam sombras foram considerados objetos.



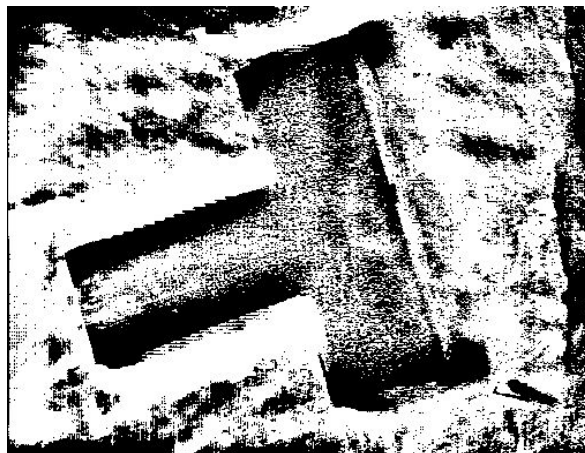
peppers size=25



peppers size=45



wedge size=45



wedge size=45

5 - Conclusão

Por fim devemos ressaltar que os filtros foram aplicados para mais de uma imagem, com diferentes parâmetros e este relatório apresenta características observadas para a maioria das imagens. Dos métodos apresentados nesse relatório, nenhum deles parece ser perfeito, e logo cada um deles pode ter um melhor comportamento em diferentes imagens (com maior ou menor luminosidade e sombreamento). Enfim, com o Projeto 3 obtivemos contato com a técnica de limiarização de imagens de meios tons e como aplicá-la em imagens monocromáticas, na tentativa de entender os diferentes comportamentos para cada método, com seus diferentes parâmetros, observando seus efeitos e aprendendo onde tais técnicas e filtros podem ter melhores aplicações. (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)