

# MC920 - Trabalho 3

Yuri Luiz de Oliveira  
188802

1 de Dezembro de 2020

## 1 Introdução

O trabalho consiste em aplicar métodos de limiarização global e local em imagens monocromáticas. A limiarização consiste em encontrar um limiar para identificar objetos e separá-los do fundo da imagem. Métodos globais definem que limiares que não consideram a vizinhança, enquanto métodos locais consideram o valor dos pixels vizinhos.

### 1.1 Métodos de limiarização

Os métodos de limiarização explorados neste relatório foram:

#### 1.1.1 Método Global

Neste método define-se um valor de limiar constante que será utilizado para todos os pixels da imagem.

#### 1.1.2 Método de Bernsen

Para cada pixel  $p(x,y)$  o limiar é calculado através da seguinte expressão:

$$T(x,y) = \frac{z_{min} + z_{max}}{2}$$

em que os valores são o mínimo e máximo local, respectivamente. Limiares locais são calculados baseados em uma vizinhança local  $n$  por  $n$ .

#### 1.1.3 Método de Niblack

O valor do limiar é calculado através da expressão

$$T(x,y) = \mu(x,y) + k\sigma(x,y)$$

sendo  $\mu(x,y)$  e  $\sigma(x,y)$ , a média e desvio padrão, respectivamente.

#### 1.1.4 Método de Sauvola e Pietaksinen

O valor do limiar é calculado através da expressão

$$T(x, y) = \mu(x, y) \left[ 1 + k \left( \frac{\sigma(x, y)}{R} - 1 \right) \right]$$

sendo que  $\mu(x, y)$  e  $\sigma(x, y)$  seguem a mesma definição do método anterior. Sauvola e Pietaksinen sugerem  $k = 0.5$  e  $R = 128$ .

#### 1.1.5 Método de Phansalskar, More e Sabale

O valor do limiar é calculado através da expressão

$$T(x, y) = \mu(x, y) \left[ 1 + p \exp(-q\mu(x, y)) + k \left( \frac{\sigma(x, y)}{R} - 1 \right) \right]$$

sendo que  $\mu(x, y)$  e  $\sigma(x, y)$  seguem a mesma definição do método anterior. Os autores sugerem  $k=0.25$ ,  $R=0.5$ ,  $p = 2$  e  $q = 10$ .

#### 1.1.6 Método do contraste

O pixel será considerado como pertencente ao objeto se seu valor estiver mais próximo do máximo local, caso contrário será considerado de fundo.

#### 1.1.7 Método da média

O valor do limiar é calculado como a média de sua vizinha. Ou seja,

$$T(x, y) = \mu(x, y)$$

sendo  $\mu(x, y)$  a média da vizinhança.

#### 1.1.8 Método da mediana

O valor do limiar é calculado como a mediana de sua vizinha.

### 1.2 Imagens

Todas as imagens abaixo foram utilizadas para testes, mas somente algumas serão usadas para expor os resultados.

### 1.2.1 baboon.pgm

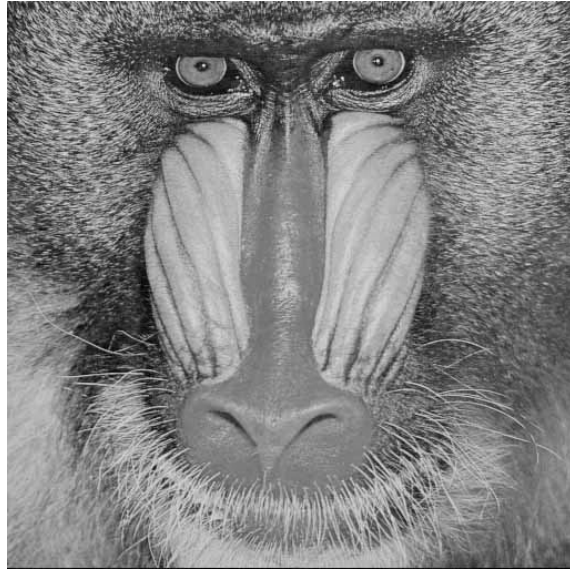


Figure 1: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/baboon.pgm](https://www.ic.unicamp.br/~helio/images_pgm/baboon.pgm)

### 1.2.2 fiducial.pgm

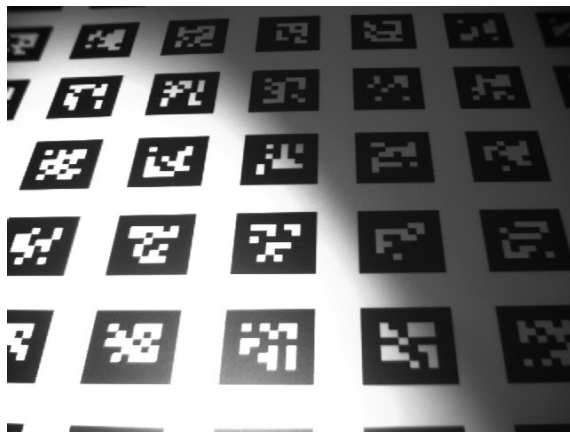


Figure 2: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/fiducial.pgm](https://www.ic.unicamp.br/~helio/images_pgm/fiducial.pgm)

### 1.2.3 monarch.pgm



Figure 3: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/monarch.pgm](https://www.ic.unicamp.br/~helio/images_pgm/monarch.pgm)

### 1.2.4 peppers.pgm



Figure 4: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/peppers.pgm](https://www.ic.unicamp.br/~helio/images_pgm/peppers.pgm)

### 1.2.5 retina.pgm

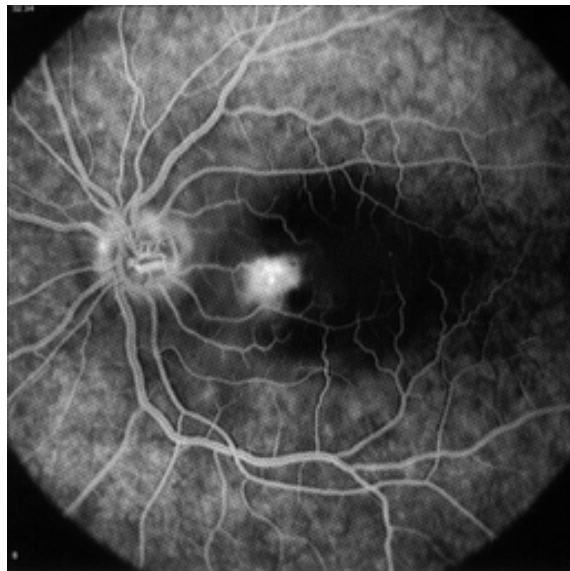


Figure 5: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/retina.pgm](https://www.ic.unicamp.br/~helio/images_pgm/retina.pgm)

### 1.2.6 sonnet.pgm

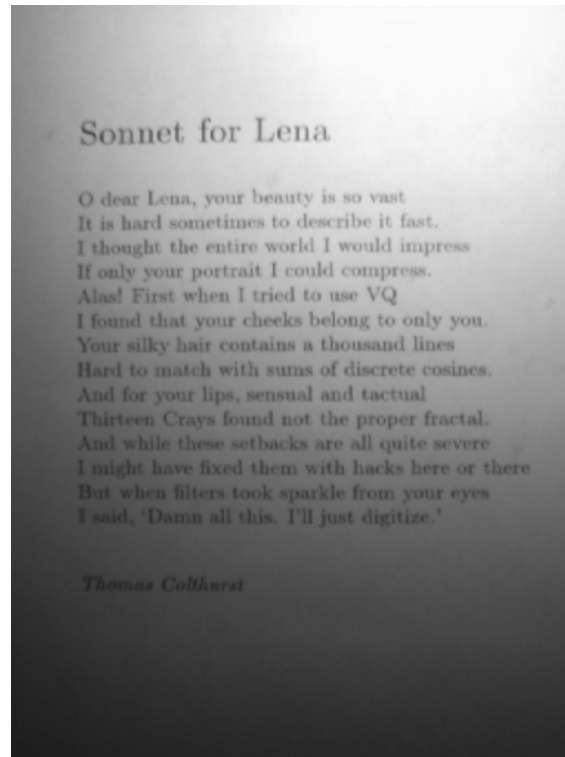


Figure 6: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/sonnet.pgm](https://www.ic.unicamp.br/~helio/images_pgm/sonnet.pgm)

### 1.2.7 wedge.pgm

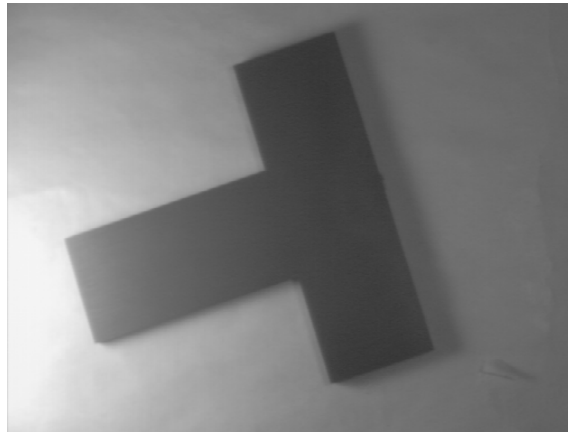


Figure 7: Imagem retirada de [https://www.ic.unicamp.br/~helio/images\\_pgm/wedge.pgm](https://www.ic.unicamp.br/~helio/images_pgm/wedge.pgm)

## 2 Ambiente de desenvolvimento e testes

A solução para o trabalho foi implementada e testada em Python 3.8.5, com auxílio das bibliotecas numpy, opencv, matplotlib, argparse e scipy. O sistema operacional utilizado foi Ubuntu 20.04.

### 2.1 Preparando o ambiente

Para instalar as bibliotecas necessárias basta entrar na pasta do trabalho e rodar o comando

```
$ pip3 install -r requirements.txt
```

## 3 Rodando a solução

A solução aceita argumentos de linha de comando e deve ser executada através do interpretador de python 3. O seguinte comando roda a solução para a imagem presente em *"images/peppers.pgm"*, com método de bernsen e salva a imagem em *"out/peppers\_bernsen.pgm"*. Os identificadores dos métodos aceitos são 'global', 'bernsen', 'niblack', 'sauvola', 'phansalskar', 'contrast', 'mean', 'median', para método global, método de Bernsen, método de Niblack, método de Sauvola e Pietaksinen, método de Phansalskar, More e Sabale, método de contraste, método da média e método da mediana respectivamente.

```
$ python3 thresholding.py -i images/peppers.pgm -m bernsen -o out/peppers_bernsen.pgm
```

### 3.1 Passando parâmetros dos métodos por argumento

O programa aceita  $T$  (limiar do método global),  $n$  (largura e altura da vizinhança a ser considerada),  $k$ ,  $R$ ,  $p$  e  $q$ . Caso seja passado um parâmetro que não é utilizado pelo método atual o valor será desconsiderado. Caso um parâmetro não seja passado, será utilizado seu valor padrão.

### 3.2 Outros argumentos de execução

Além dos argumentos utilizados no comando acima, existem opções que podem ser passadas para os comandos.

#### 3.2.1 `display_mode`

Este argumento é utilizado para definir se a saída será as imagens (origem e resultante), histograma ou nada. Os valores aceitos são 'default', 'hist' ou 'off'.

### 3.3 Comando para rodar todas as formas de difusão de erro de uma imagem específica

Para obter o resultado de todos os métodos em uma imagem específica, basta utilizar o script *all\_methods.sh*. Este script recebe a imagem para a qual deve ser rodada a solução e o diretório ao qual devem ser salvos os resultados. O comando seguinte executa o script para a imagem *"images/peppers.pgm"* e salva os resultados no diretório *"out"*. **Note que as imagens utilizadas tem que estar dentro do diretório *"images"* presente caminho atual e devem ter extensão *".pgm"*.**

```
$ ./all_methods.sh peppers out
```

### 3.4 Formato dos dados

As imagens processadas **devem estar em formato PGM**.

## 4 Detalhes e decisões de implementação

A solução foi implementada através da função *generic\_filter* da biblioteca *scipy* (pacote *scipy.ndimage*). Esta função percorre a imagem, pixel a pixel, e calcula valores para cada posição para uma matriz resultante. Desta forma, todos os métodos foram implementados de forma semelhante, alterando somente função que calcula o valor resultante. Note que cada posição da matriz resultante é o limiar de cada pixel. Ou seja, a matriz resultante  $M$  pode ser escrita através da seguinte expressão:

$$M(x, y) = T(x, y)$$

sendo  $T(x,y)$  definido para cada método na seção 1.1 Métodos de limiarização. Após a obtenção de  $M(x,y)$ , o pixel de cada imagem era calculado como preto se fosse maior do que o limiar da posição na matriz  $M(x,y)$  e branco caso contrário.



## 5 Resultados obtidos

Para analisar os resultados de cada método vamos selecionar imagens convenientes para cada análise, dado que os métodos podem funcionar bem para diferentes contextos ou imagens (fotos de texto, imagens com ou sem sombra, imagens de um único objeto, etc.).

### 5.1 Método global

O valor de limiar utilizado foi de 128.



Figure 8: Imagem resultante da aplicação do método global à imagem peppers.pgm

Como este método utiliza um limiar constante é natural que funcione melhor para imagens que possuem uma grande diferença nos valores dos pixels dos objetos e do fundo. Escolhemos então esta imagem e consideramos as sombras mais escuras como fundo. Como as sombras desta imagem são muito escuras é mais fácil selecionar um limiar que separa bem o fundo dos objetos. Se selecionarmos um limiar mais escuro, entretanto mais claro do que as sombras, os objetos podem ficar ainda mais salientes. A vizinhança utilizada foi de 3x3. A imagem resultante possui 47% dos pixels pretos.

## 5.2 Método de Bernsen



Figure 9: Imagem resultante da aplicação do método de Bernsen à imagem peppers.pgm

O método de Bernsen funcionou muito bem para separar as sombras mais escuras das pimentas. A vizinhança utilizada foi de 3x3. A imagem resultante possui 91% dos pixels pretos.

### 5.3 Método de Niblack

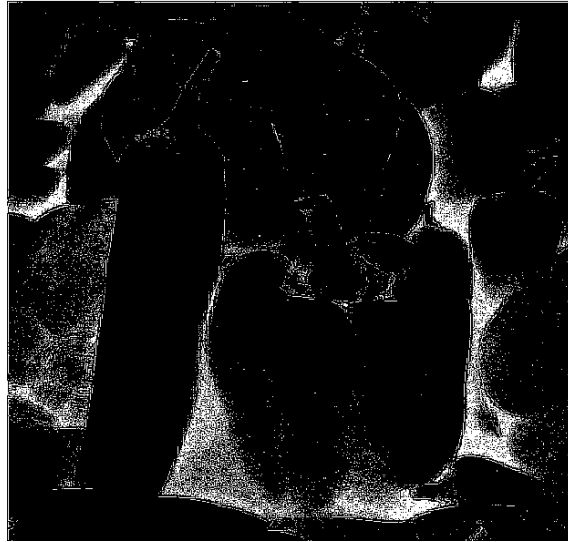


Figure 10: Imagem resultante da aplicação do método de Niblack à imagem peppers.pgm

O método de Niblack (com vizinhança 3x3 e  $k=0.1$ ) obteve um resultado um pouco pior do que o método de Bernsen (com vizinhança 3x3). Entretanto, ao aumentar a vizinhança para 6x6 os detalhes de contorno do método de Niblack ficam mais evidentes, como visto na Figura 11. A imagem resultante possui 93% dos pixels pretos.



Figure 11: Imagem resultante da aplicação do método de Niblack à imagem peppers.pgm com vizinhança 6x6.

Ao utilizar o parâmetro  $k=0.5$ , o método ficou muito sensível as bordas do objeto, como visto na Figura 12.



Figure 12: Imagem resultante da aplicação do método de Niblack à imagem peppers.pgm com vizinhança 6x6 e  $k=0.5$ .

## 5.4 Método de Sauvola e Pietaksinen

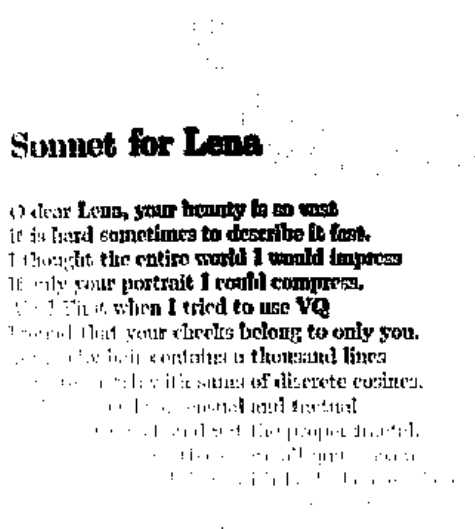


Figure 13: Imagem resultante da aplicação do método de Sauvola e Pietaksinen à imagem sonnet.pgm

Como descrito no enunciado, o método de Sauvola e Pietaksinen busca melhorar os resultados do método de Niblack para fotos de documentos com má iluminação, vemos, desta forma, um resultado muito bom em salientar texto, entretanto, não conseguiu mostrar o texto que está em pontos mais escuros da foto com os parâmetros sugeridos pelos autores ( $k=0.5$  e  $R=128$ ). A vizinhança utilizada foi de  $3 \times 3$ . A imagem resultante possui 5% dos pixels pretos.

Ao experimentar outros valores para  $k$  (mantendo  $R=128$ ), vemos que ao diminuir o valor de  $k$ , a iluminação começa a ser interpretada como parte do conteúdo da imagem, como visto na figura 14 em que  $k=0.1$ . Enquanto, ao utilizar valores maiores, o texto fica mais apagado, como visto na figura 15 em que  $k=0.8$ .



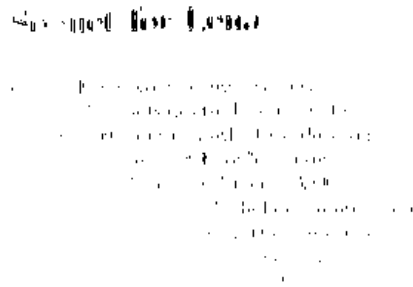


Figure 15: Imagem resultante da aplicação do método de Sauvola e Pietaksinen à imagem sonnet.pgm com  $k=0.8$  e  $R=128$ .

Ao experimentar com  $R$  menores, percebemos que o texto fica mais apagado e com  $R$  maiores a iluminação começa a ser identificada como parte do conteúdo, como visto nas figuras 16 e 17 (mantendo  $k=0.5$ ).

## Sonnet for Lema

My dear Lema, your features are so sweet  
I wish I could engrave a book about you  
So that the entire world I could equip  
With a sonnet portrait I could engrave  
So that every child I meet on my  
Way to school could know what I am  
And how I feel about you, and how  
I wish I could engrave a book about you  
So that every child I meet on my  
Way to school could know what I am  
And how I feel about you, and how  
I wish I could engrave a book about you  
So that every child I meet on my  
Way to school could know what I am  
And how I feel about you, and how  
I wish I could engrave a book about you

Figure 16: Imagem resultante da aplicação do método de Sauvola e Pietaksinen à imagem sonnet.pgm com  $k=0.5$  e  $R=64$ .



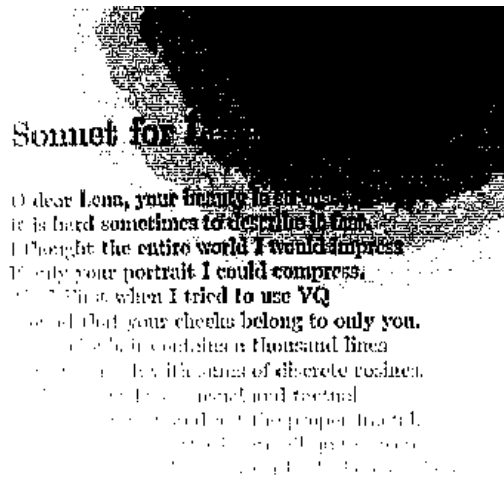


Figure 17: Imagem resultante da aplicação do método de Sauvola e Pietaksinen à imagem sonnet.pgm com  $k=0.5$  e  $R=256$ .

Notamos que ao deixar o texto presente na parte mais escura da imagem original (inferior esquerdo) o texto situado na parte mais clara da imagem fica comprometido (com mais ruído nas letras) e mais difícil de compreender. O melhor resultado obtido com uma vizinhança de  $6 \times 6$ ,  $k=0.45$  e  $R=128$  (ainda deixando o texto do canto inferior esquerdo muito claro). (Figura 18).

## Sonnet for Lena

O dear Lena, your beauty is so great  
It is hard sometimes to describe it fast.  
I thought the entire world I would 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 silly hair contains a thousand lines  
Hard to match with sums of discrete cosines.  
And for your lips, sensual and tactual  
Claude L. Orsay found not the proper fractal.  
So I hope these sketches are all quite severe  
For I have mixed them with Lucks late or there  
So that each could be from your eyes  
I hope they will put forth the

Figure 18: Imagem resultante da aplicação do método de Sauvola e Pietaksinen à imagem sonnet.pgm com vizinhança 6x6,  $k=0.45$  e  $R=128$ .

## 5.5 Método de Phansalskar, More, Sabale

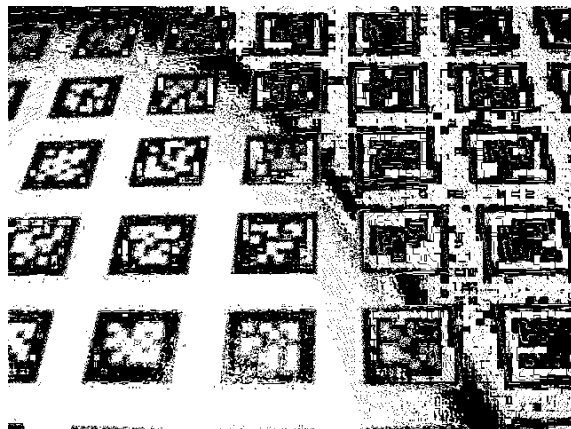


Figure 19: Imagem resultante da aplicação do método de Phansalskar, More, Sabale à imagem fiducial.pgm

Como descrito no enunciado, o método de Phansalskar, More, Sabale é uma variação do método de Savula e Pietaksinen para lidar com imagens de baixo contraste, portanto selecionamos a imagem fiducial.pgm, que possui baixo contraste (evidenciado por seu histograma - Figura 26). Como resultado, utilizando os parâmetros sugeridos pelos autores ( $k=0.25$ ,  $R=0.5$ ,  $p=2$ ,  $q=10$ ), o método foi capaz de separar boa parte dos objetos da sombra, como visto na figura 19. A imagem resultante possui 43% dos pixels pretos.

Entendemos que, como este método é uma variação do método de Sauvola e Pietaksinen, esperávamos que funcionasse bem para textos (principalmente em imagens com baixo contraste) como é o caso da imagem sonnet.pgm, entretanto, o resultado do método de Phansalskar, More, Sabale na imagem do soneto foi bem ruim e não chegamos a nenhuma combinação dos parâmetros que tornasse o texto (ou parte dele) legível. O resultado deste método aplicado a sonnet.pgm pode ser visto na imagem 20.

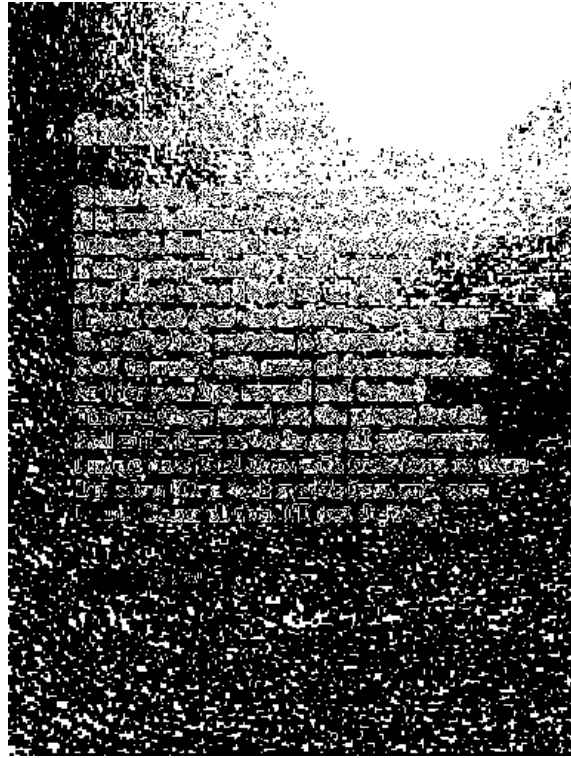


Figure 20: Imagem resultante da aplicação do método de Phansalskar, More, Sabale à imagem sonnet.pgm

## 5.6 Método de contraste

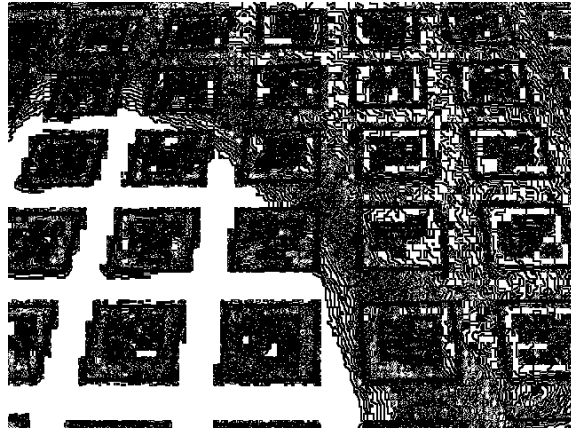


Figure 21: Imagem resultante da aplicação do método de contraste à imagem fiducial.pgm

Com o método de contraste, no interior dos objetos, perdeu-se detalhamento, onde existe mudanças drásticas de contraste (de preto para branco) muito próximas umas das outras. A imagem resultante possui 66% dos pixels pretos.

## 5.7 Método da média

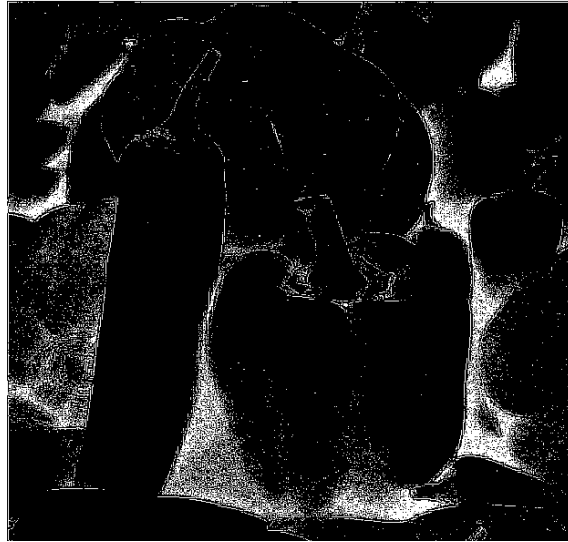


Figure 22: Imagem resultante da aplicação do método da média à imagem peppers.pgm

O método da média se mostrou eficaz em identificar as sombras mais escuras da imagem peppers.pgm como fundo, o que é esperado, uma vez que a sombra é mais escura (pixels mais próximos a zero) do que as pimentas, logo a média dos valores dos pixels das sombras naturalmente será menor do que a média (nas regiões entre as pimentas e as sombras). A imagem resultante possui 92% dos pixels pretos.

## 5.8 Método de mediana

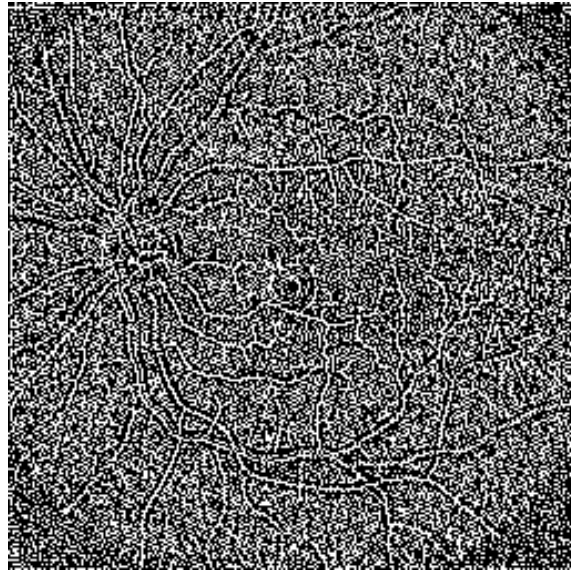


Figure 23: Imagem resultante da aplicação do método da mediana à imagem retina.pgm

Para todas as imagens, o método da mediana ficou com um ruído muito intenso. Entretanto, o resultado sobre a imagem retina.pgm, mostrou que as linhas do olho ficaram evidenciadas. A imagem resultante possui 68% dos pixels pretos.

## 5.9 Histograma das imagens originais

### 5.9.1 Peppers.pgm

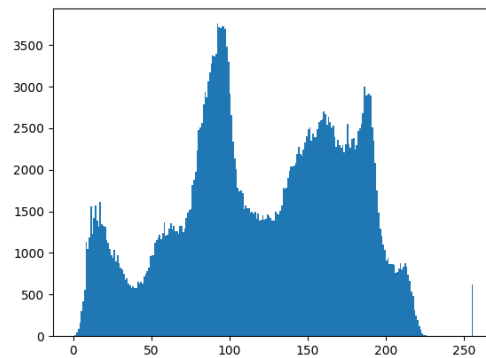


Figure 24: Histograma da imagem peppers.pgm

### 5.9.2 Sonnet.pgm

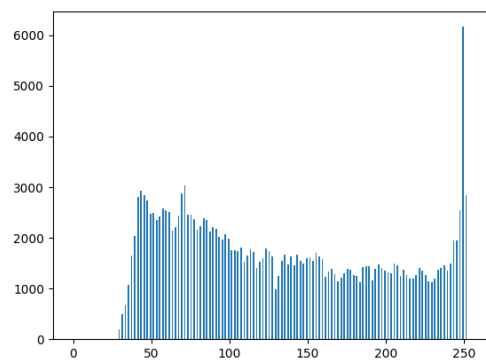


Figure 25: Histograma da imagem sonnet.pgm



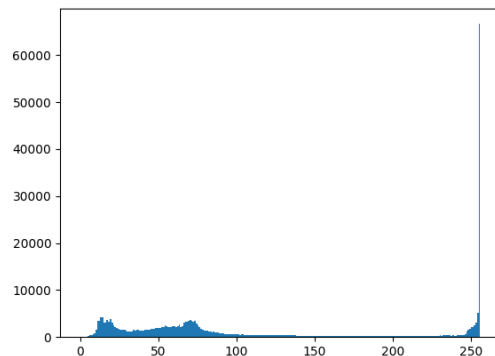


Figure 26: Histograma da imagem fiducial.pgm

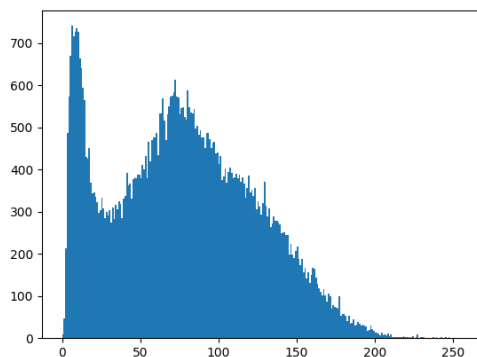


Figure 27: Histograma da imagem retina.pgm

## 6 Principais falhas durante o desenvolvimento

Dado que a utilização da função `generic_filter` reduz bastante a complexidade de implementação da solução, a única falha significativa de implementação foi a ausência de normalização dos limiares. Ou seja, em alguns casos, os valores do limiares ficavam maiores do que 255, o que tornava mais difícil o aparecimento de pixels de objeto (pixels pretos). Esta falha ficou muito evidente ao utilizar o método de Sauvola e Pietaksinen, utilizando os parâmetros sugeridos, na imagem do soneto, pois a imagem ficava inteira branca (somente fundo), como visto na figura 28.

Figure 28: Resultado do método de Sauvola e Pietaksinen sem normalização dos limiares na imagem sonnet.pgm

A solução deste problema foi normalizar os limiares de 0 a 255 antes de comparar os pixels a eles. O resultado pode ser visto na imagem 13.

## 7 Limitações

A solução pode ser utilizada somente com imagens em escala de cinza, com intensidades entre 0 a 255 (por causa da normalização dos limiares). A solução não está adaptada para imagens coloridas, uma vez que considera que os pixels são um inteiro e não a composição de três valores (para o caso de imagens coloridas).

## 8 Conclusões

É difícil apontar um método de limiarização como melhor, dado que cada um funciona bem para imagens diferentes, como o método de Sauvola e Pietaksinen que funciona bem para imagens de documentos com má iluminação enquanto o método de Phansalskar, More e Sabale funciona bem para imagens de baixo

contrate, mesmo o segundo sendo uma variação do primeiro. É importante, portanto, levar em consideração a aplicação para escolher o melhor método. E, mesmo métodos mais simples, com o método global, podem ser utilizados de forma efetiva (em imagens em que o objetos tem intensidades de pixels muito diferentes em relação ao fundo, por exemplo).