

# Trabalho 2 - Técnicas de Limiarização

MC920 - Introdução ao Processamento de Imagem Digital

Victor Ferreira Ferrari

RA 187890

vferrari@mpc.com.br

01 de Outubro de 2019

## 1 Introdução

Uma das técnicas mais simples e utilizadas de segmentação em imagens monocromáticas é a limiarização, que consiste na binarização da imagem em objeto (preto) e fundo (branco). Essa técnica funciona melhor para imagens cujo histograma possui dois grupos dominantes, pois basta definir um limiar único e o resultado final será bom (limiarização global).

Existem diferentes métodos de limiarização propostos pela literatura, que podem ser divididos em dois grupos: limiarização global e local. A limiarização global consiste em utilizar um único valor de limiar para separar a imagem, podendo esse valor ser encontrado de diversas formas. A limiarização local consiste em considerar uma vizinhança em forma de janela  $n \times n$  que define um limiar para o pixel central. A escolha do limiar, dada a vizinhança, pode ser feita de diversos modos.

O método de limiarização global utilizado será direto: o limiar é passado como argumento para o programa, que apenas o aplica na imagem. Os 7 métodos de limiarização local implementados são:

- Método de Bernsen;
- Método de Niblack;
- Método de Sauvola e Pietaksinen, para imagens com má iluminação;
- Método de Phansalskar, More e Sabale, para imagens de baixo contraste;
- Método do Contraste;
- Método da Média;
- Método da Mediana.

O intuito do projeto é explorar a técnica de limiarização com 8 diferentes métodos envolvendo limiarização global e local, com imagens monocromáticas. A escolha dos parâmetros também será brevemente explorada.

## 2 Características do Programa

O programa foi feito na linguagem *Python*, com auxílio das bibliotecas externas NumPy, Matplotlib e OpenCV (CV2). Os argumentos são passados na execução via `argv`, permitindo modificar diversas condições de execução. A entrada deve ser uma imagem monocromática. Serão usadas imagens em formato PGM para teste.

Os argumentos do programa podem ser:

- `file`: Imagem para aplicar a limiarização;
- `method`: Nome do método de limiarização a ser aplicado, podendo ser "global" para limiarização global, ou o primeiro nome dos métodos de limiarização local descritos na introdução (exceto Phansalskar, cujo código é "more"), ou "all" para executar para todos os métodos disponíveis com os mesmos parâmetros;
- `--thresh`: opcional, valor do limiar global (padrão é 128);
- `--size`: opcional, tamanho da janela de limiarização local (padrão é 7, para janela 7x7).
- `--k`: opcional, parâmetro k dos métodos de Niblack, Sauvola e Pietaksinen e Phansalskar, More e Sabale (padrão é 0,25).
- `--R`: opcional, parâmetro R dos métodos de Sauvola e Pietaksinen e Phansalskar, More e Sabale (padrão é 0,5).
- `--p`: opcional, parâmetro p do método de Phansalskar, More e Sabale (padrão é 2).
- `--q`: opcional, parâmetro q do método de Phansalskar, More e Sabale (padrão é 10).
- `--folder`: opcional, pasta de saída (precisa existir);

## 3 Implementação

Como visto na seção 1, foram implementados 8 métodos de limiarização, sendo 1 global e 7 locais. A leitura da imagem monocromática é feita via biblioteca `OpenCV`, em matrizes `NumPy`.

O método de limiarização global adotado é simples, no qual um valor de limiar é passado como argumento para o programa (como visto na seção 2) e a limiarização é feita em torno dele, ou seja, os pixels de intensidade maior que o limiar são considerados fundo (branco) e os de intensidade menor são considerados objeto (preto).

Nos métodos locais, a imagem é percorrida pixel a pixel, tomando-se a vizinhança quadrada da imagem de tamanho recebido como argumento do programa. Para cada pixel, um valor de limiar é calculado baseado no método e nos parâmetros correspondentes, e armazenado em uma matriz de limiares. Assim, a saída não interfere na execução, e a vizinhança sempre é obtida da imagem original. Depois, os valores de saída são determinados a partir de cada limiar. O controle de bordas é feito reduzindo o tamanho da janela nos cantos, então ela só possui o tamanho desejado quando há espaço.

No método de Bernsen, o limiar é calculado como:

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

No método de Niblack:

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

No método de Sauvola e Pietaksinen:

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

No método de Phansalskar, More e Sabale:

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

No método do contraste, é verificado se o pixel está mais próximo do máximo ou do mínimo para definir se é fundo ou objeto. Porém, como o limiar é retornado e não o valor exato do pixel, se o pixel deve ser preto, o limiar retornado é 255 e caso contrário, 0. Nos métodos da média e mediana, como os nomes indicam, o limiar é dado pela média ou mediana da janela.

Aplicado o método desejado de limiarização, um histograma é calculado para a imagem original, com a quantidade de pixels em cada intensidade de cinza na imagem, e transformado em uma imagem com auxílio da biblioteca `Matplotlib`. Além disso, as frações de pixels pretos e brancos na imagem resultado são calculados e impressos na saída padrão.

A matriz resultado e o histograma são salvos na pasta "Outputs/" (caso outra pasta não tenha sido passada como argumento de `--folder`).

## 4 Resultados e Comparação

Foram testadas 7 imagens monocromáticas em formato PGM, com variados níveis de contraste, resolução (detalhamento) e qualidade de iluminação. Todas as imagens de entrada então disponíveis em [https://www.ic.unicamp.br/~helio/imagens\\_pgm/](https://www.ic.unicamp.br/~helio/imagens_pgm/).

Três imagens em específico, de nomes `fiducial.pgm`, `sonnet.pgm` e `wedge.pgm` são interessantes de se analisar mais profundamente pois possuem má iluminação, então aplicar limiarização sem pré-processamento pode se mostrar difícil.

### 4.1 Análise das Imagens Originais

Os melhores métodos e valores de limiar podem ser induzidos a partir da análise do histograma de cada imagem. Os histogramas de todas as imagens estão na figura 1.

Excluindo-se as três imagens mal-iluminadas, temos que as imagens possuem distintos níveis de contraste. A imagem `peppers.pgm`, por exemplo, possui alto contraste, com diversos picos de intensidade em diferentes intensidades de maneira bem distribuída. A imagem `baboon.pgm` possui um contraste um pouco menor, ficando principalmente entre 50 e 200.

Observando o histograma da imagem `Retina`, vê-se uma divisão entre picos perto da região de intensidade 25. Isso pode ser útil quando se fala de limiarização global.

### 4.2 Limiarização Global

Quando se trata de limiarização global, o histograma pode ser uma ferramenta útil para descobrir o melhor limiar. O limiar padrão é 128, mas quando há dois picos claros de intensidade, outro limiar pode ser interessante de explorar.

Como dito anteriormente, o histograma da imagem `Retina` possui uma divisão entre picos perto da região de intensidade 25. Esse pode ser um bom ponto de limiar global para segmentar algum detalhe.

Resultados para todas as imagens, tirando as com má iluminação, com limiar padrão (128), e o resultado da imagem `Retina` com limiar 25, podem ser vistos na figura 2.

As frações de preto para cada imagem são:

- Baboon: 0,461
- Monarch: 0,775
- Peppers: 0,526
- Retina: 0,843
- Retina T25: 0,179

Observando a imagem 2e, vemos que um detalhe da imagem que a figura 2d não engloba é exatamente o que foi segmentado. Isso não representa toda a imagem, mas um detalhe da imagem foi segmentado com a técnica.

Já as imagens com má iluminação não possuem resultados tão bons, pois a sombra é considerada objeto, como visto na figura 3.

Claramente, o limiar 128 não funcionou para essas imagens. A imagem na qual isso fica mais claro é na **Sonnet**, na qual o texto (objeto) é considerado parte do fundo, mas a sombra é considerada objeto.

### 4.3 Imagens com Má Iluminação

Como mencionado anteriormente, três imagens possuem má iluminação: `fiducial.pgm`, `sonnet.pgm` e `wedge.pgm`. As imagens podem ser vistas na figura 4.

Pela figura 1, pode-se ver que **sonnet**, que deveria pela natureza da imagem possuir poucos níveis de cinza, possui uma distribuição quase uniforme, graças ao gradiente de cinza criado pela má iluminação.

Um dos métodos implementados tem a característica de funcionar bem para imagens de má iluminação: o método de Sauvola e Pietaksinen. Assim, se essa característica estiver correta, basta definir os melhores parâmetros  $k$  e  $R$  para cada imagem. A literatura sugere os valores  $k=0,5$  e  $R=128$ , mas cada imagem deve ter seus parâmetros ótimos, então outros foram testados. As imagens finais, com seus parâmetros, podem ser vistas na figura 5.

As frações de preto para cada imagem são:

- Fiducial padrão: 0,178
- Fiducial melhor: 0,306
- Sonnet padrão: 0,000
- Sonnet melhor: 0,109
- Sonnet com menos ruído: 0,089
- Wedge padrão: 0,000
- Wedge melhor: 0,029

Na figura, fica claro que os parâmetros sugeridos não fornecem boas respostas para essas imagens em específico. A imagem 5a é a única que consegue encontrar alguns elementos da imagem original, e ainda assim há muitas descontinuidades. Nas outras imagens, o resultado é branco.

Nas imagens com parâmetros escolhidos, percebe-se uma grande melhora na segmentação em relação ao método global. Na figura 5d, o soneto está quase legível, porém a imagem ainda possui muito ruído da sombra. Já na figura 5e, o soneto não está tão legível, mas há bem menos

ruído. Mudando os parâmetros, o resultado pode ainda ficar melhor, mas nos diversos testes feitos esses resultados foram os melhores encontrados.

Já na imagem 5g, vemos que a segmentação das bordas do objeto foi muito bem sucedida, com pouco ruído e poucas descontinuidades. O interior do objeto foi considerado fundo, e o objeto identificado foi a borda, porém isso já é suficiente para identificar o objeto.

## 4.4 Outros Métodos

Por fim, podemos examinar outros métodos de limiarização local. Na figura 6, pode-se encontrar o resultado de limiarização da figura `peppers.pgm` para todos os outros métodos.

Foram utilizados parâmetros padrão: Para Niblack,  $k=-0,2$ , para Phansalskar, More e Sabale,  $k=0,25$ ,  $R=0,5$ ,  $p=2$  e  $q=10$ . O tamanho de janela utilizado foi 15.

As frações de preto para cada imagem são:

- Bernsen: 0,093
- Niblack: 0,385
- Phansalskar, More e Sabale: 0,999
- Contraste: 0,493
- Média: 0,483
- Mediana: 0,445

Como pode ser visto, os métodos causaram diferentes efeitos nas imagens. Sem analisar diversos parâmetros, é difícil verificar os prós e contras de cada método. Observando as imagens, temos que o método de Bernsen não foi totalmente bem-sucedido em encontrar os objetos, possuindo muitas descontinuidades nas bordas encontradas.

No método de Niblack, a imagem já foi melhor descrita, mas com bastante "pontilhado" no meio dos objetos, com as bordas brancas. O método do contraste possui bem menos pontilhado, com áreas pretas maiores. O método da média forneceu um resultado que parece um meio-termo entre o Niblack e o Contraste. O método da Mediana gerou um resultado similar ao Niblack, mas as bordas estão menos definidas e possuem mais descontinuidades.

Um método totalmente discrepante foi o de Phansalskar, More e Sabale, que forneceu uma imagem preta. Isso se deve ao fato de servir mais para imagens com baixo contraste, e `Peppers` possui alto contraste.

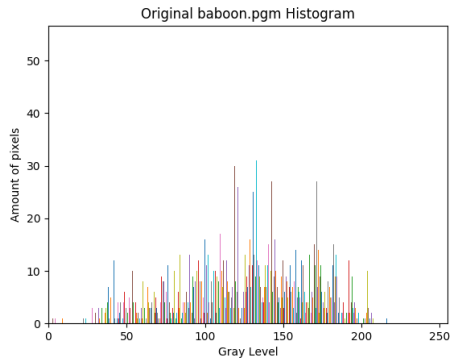
No geral, os métodos foram bem-sucedidos, mesmo com os valores sugeridos de parâmetro, em segmentar as imagens, com o melhor resultado sendo o método da Média.

## 5 Conclusão

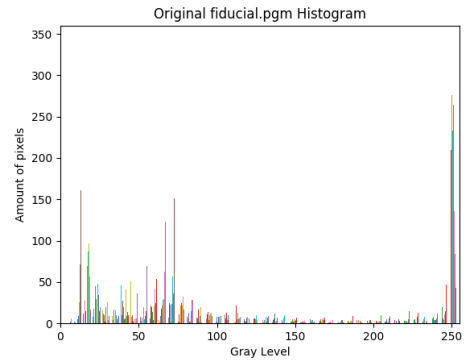
Dos testes feitos, diversas conclusões podem ser tiradas. Limiarização global, mesmo que fornecendo resultados que podem servir para situações específicas, principalmente se a escolha de limiar for auxiliada pelo histograma, não é uma técnica adequada para todos os casos, principalmente os de baixa iluminação.

Para esses últimos casos, há um método de limiarização local que fornece um bom resultado ao se testar alguns conjuntos de parâmetros, o método de Sauvola e Pietaksinen. Supõe-se que seja possível chegar em resultados melhores que os alcançados aqui, com mais testes e tempo.

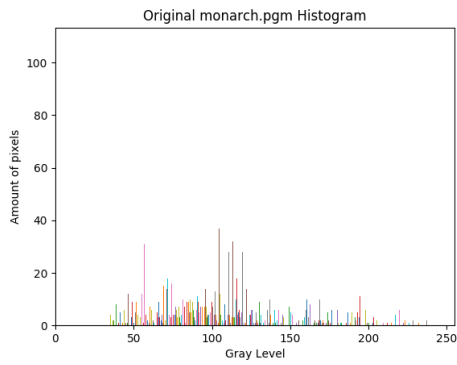
Enfim, não foi possível a realização de mais testes com todos os métodos e mais testes de parâmetros, que podem gerar melhores resultados. Outros testes (sem valores de parâmetros) podem ser vistos na pasta "Outputs/", presente na pasta de entrega.



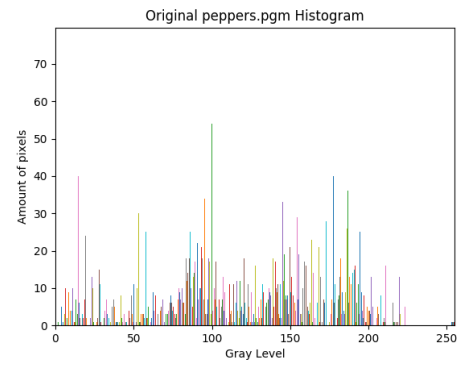
(a) Baboon



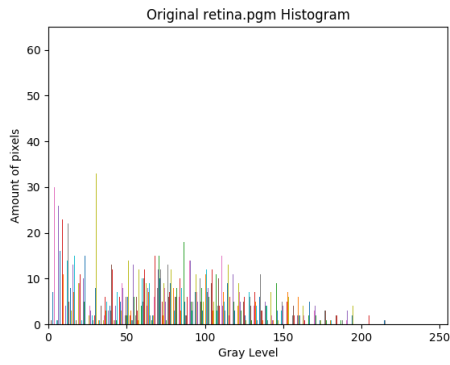
(b) Fiducial



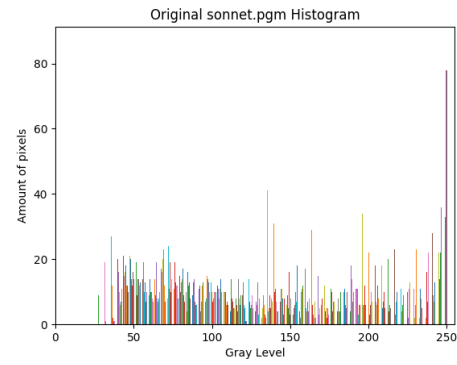
(c) Monarch



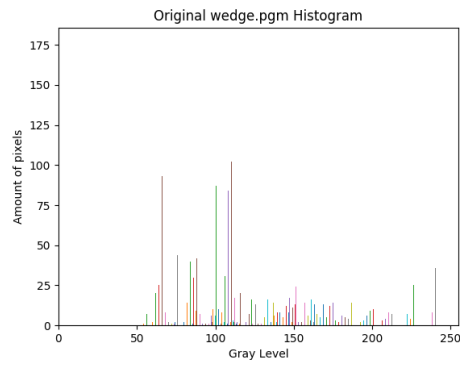
(d) Peppers



(e) Retina



(f) Sonnet



(g) Wedge

Figura 1: Histogramas de todas as imagens testadas.



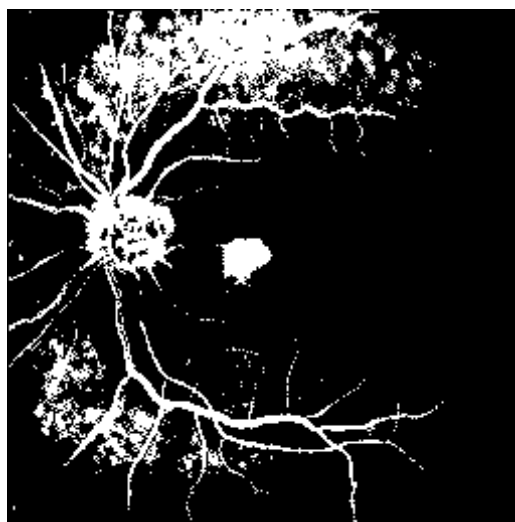
(a) Baboon



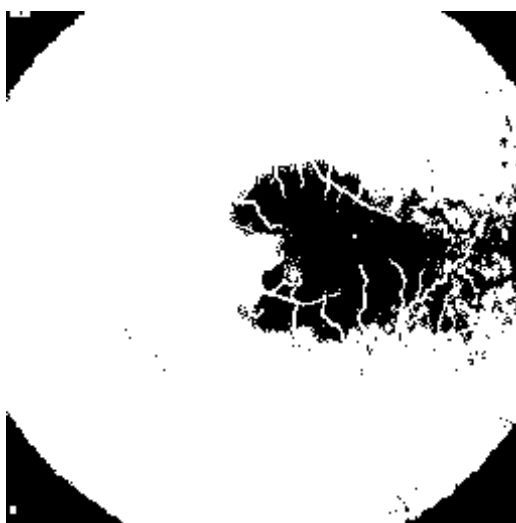
(b) Monarch



(c) Peppers



(d) Retina ( $T = 128$ )



(e) Retina ( $T = 25$ )

Figura 2: Imagens com limiarização global.

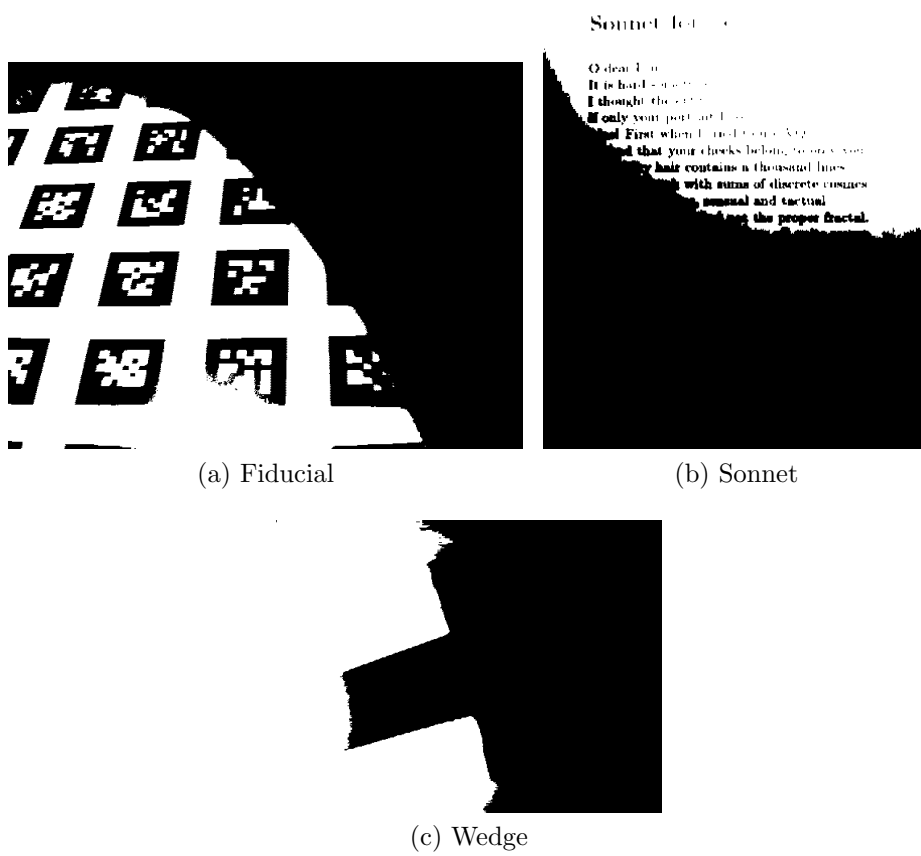
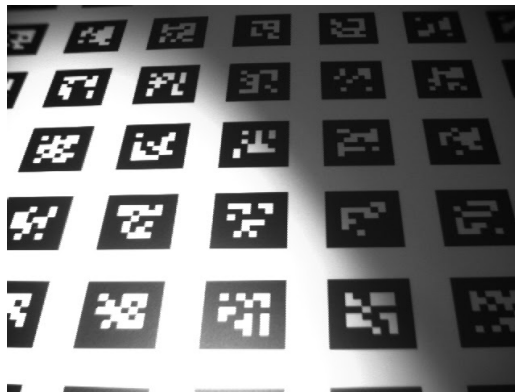
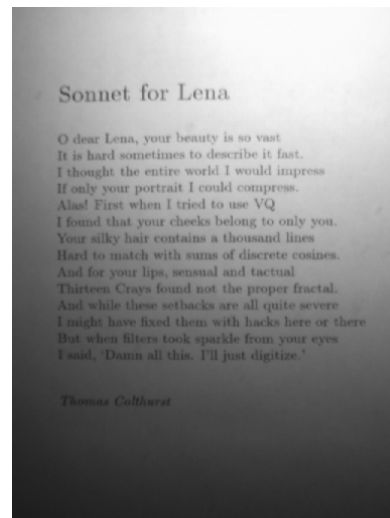


Figura 3: Imagens com má iluminação, após limiarização global com limiar 128.

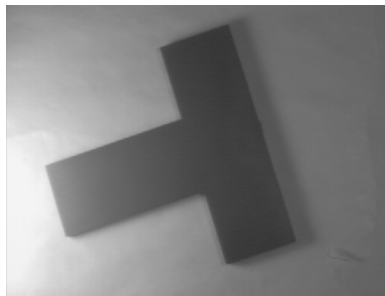




(a) Fiducial



(b) Sonnet



(c) Wedge

Figura 4: Imagens de entrada do programa com má iluminação.

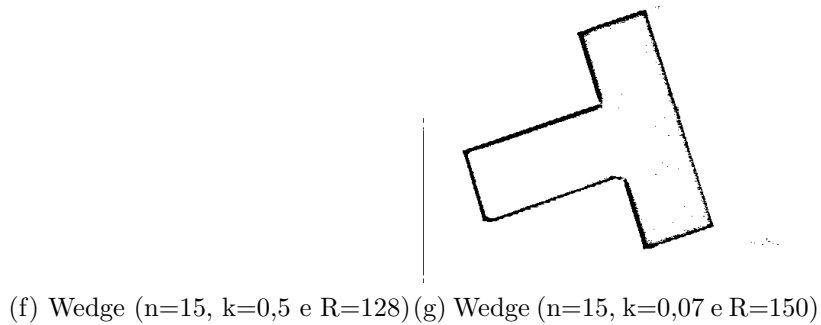
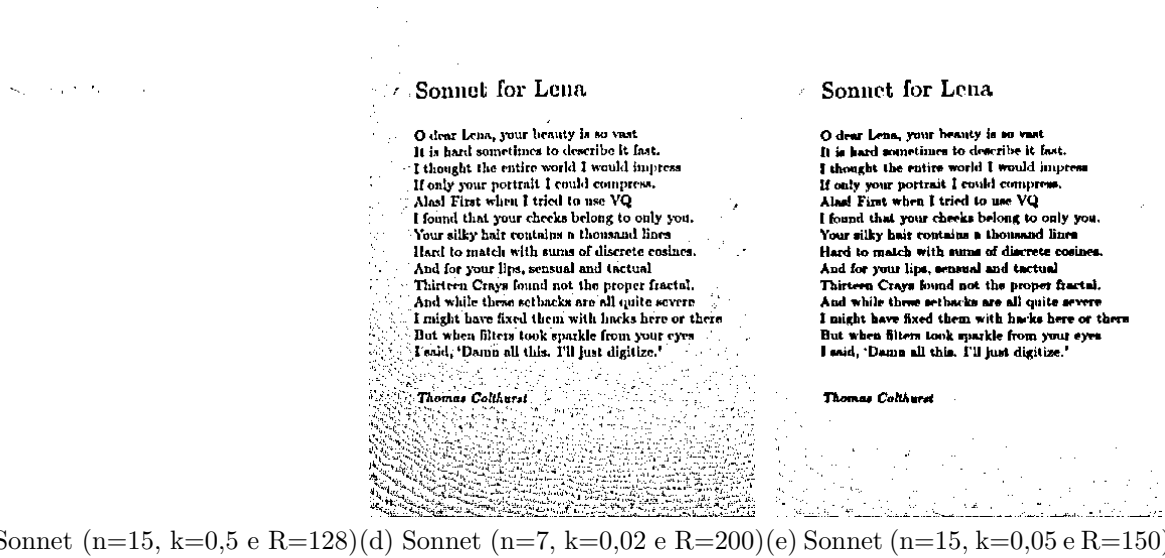
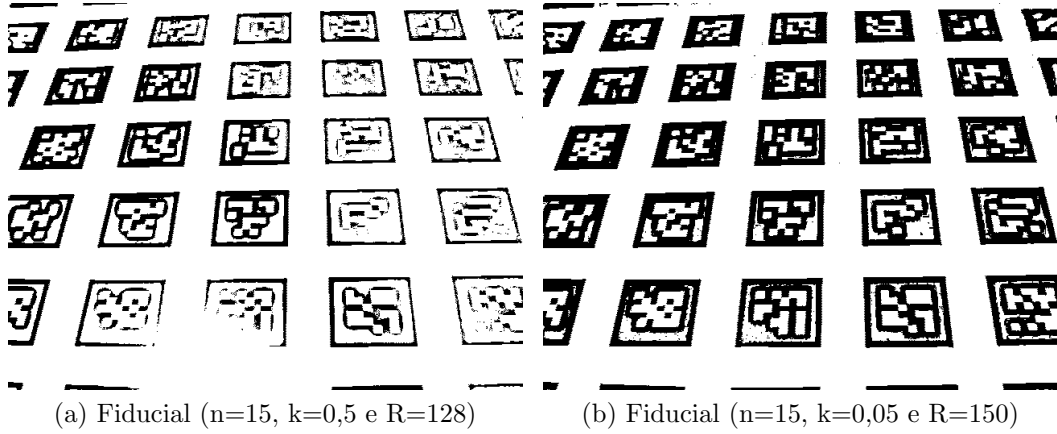


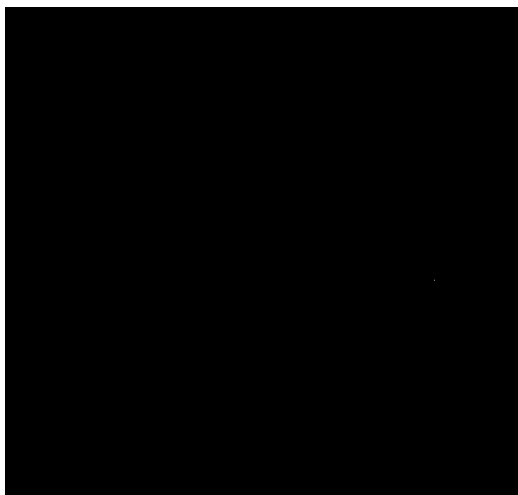
Figura 5: Imagens de má iluminação após limiarização por método de Sauvola e Pietaksinen, com seus parâmetros.



(a) Bernsen



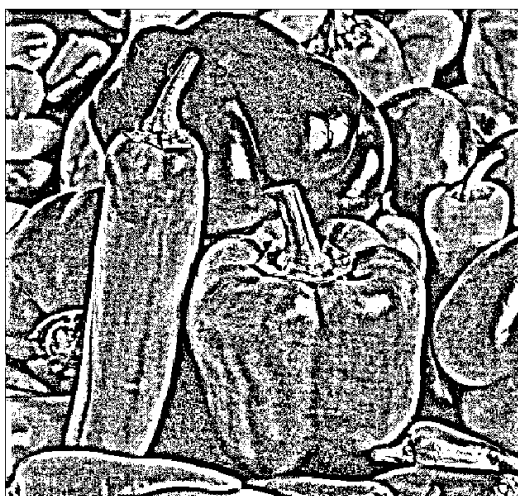
(b) Niblack



(c) Phansalskar, More e Sabale



(d) Contraste



(e) Média



(f) Mediana

Figura 6: Imagem `peppers.pgm` após limiarização local com diversos métodos.