

Trabalho 3

MO443 - Introdução ao Processamento de Imagem Digital

Diego Alysson Braga Moreira

RA:230640

diegoalyssonbm@gmail.com / d230640@dac.unicamp.br

UNICAMP

I. INTRODUÇÃO

Com o objetivo de realizar segmentação de imagens, neste trabalho serão aplicadas diferentes técnicas de limiarização, que consiste em classificar pixels de acordo com os limiares de uma imagem, alcançando assim uma imagem segmentada em diferentes regiões. Para isto implementaremos uma limiarização binária, chamada de binarização, dividindo estas imagens em primeiro plano (Foreground) e plano de fundo (Background).

Para verificar este efeito, serão utilizadas duas principais formas de limiarização, através de uma limiarização global ou limiarização local, que geram diversas técnicas. Para validação, múltiplas imagens são utilizadas, com diferentes tipos de iluminação.

Cada um desses problemas serão abordados e discutidos nas próximas seções, com instruções de execução, modelos e decisões de implementação, saídas dos dados, resultados e conclusões.

II. EXECUÇÃO

Para a execução padrão deste projeto, o arquivo de implementação dos códigos, imagens e pastas foram enviadas em conjunto. Abaixo segue a lista dos componentes que podem ser encontrados neste trabalho:

- Pasta Códigos: Arquivo de Execução do Código: trabalho3.py (descrito a seguir), histogram.py (descrito na seção III-B)
- Pasta com Imagens: baboon.pgm, fiducial.pgm, monarch.pgm, peppers.pgm, retina.pgm, sonnet.pgm e wedge.pgm.
- Pasta Raiz: Pasta Código, Pasta com Imagens e Relatório em PDF.

Onde, as imagens estão em uma subpasta chamada imagens_pgm, a qual deve ser utilizada como padrão de localização das imagens utilizadas. As pastas e seu conteúdo (arquivos .py e imagens) estão na pasta Raiz do projeto, juntamente com o relatório em pdf. O arquivo .py necessário para a execução e observação dos resultados é o arquivo trabalho3.py.

Nesta seção serão apresentadas todas as instruções de execução para ao código enviado, suas dependências, funcionalidades, parâmetros, decisões de implementação e saídas de dados.

O projeto deve ser executado com uma linha básica de chamada no terminal, como exemplo:

```
python3 trabalho3.py -i <imagem_entrada.pgm>-o <imagem_saida.pgm>
```

A. Dependências

Para a execução correta dos códigos em anexo, as seguintes dependências são essenciais:

- opencv-python: Utilizado para importação, exportação, exibição das imagens e convoluções.
- NumPy: Manipulação de vetores e matrizes. Como métodos de rotações e conversões.
- sys: Utilizada para aquisição dos argumentos em linha de comando.
- time: Utilizada para contabilizar o tempo de execução para os métodos estudados.

B. Funcionalidades

Nesta seção será descrito as funcionalidades disponíveis no código em anexo, os parâmetros recebidos por este código e alguns exemplos de execução. Detalhes de implementação e técnicas são descritas na seção III.

Dentre as funcionalidades listadas neste trabalho, tem-se diferentes técnicas de limiarização, tanto global quanto local, entre elas, foram implementadas as seguintes métodos: Método Global, Método de Otsu, Método de Bernsen, Método de Niblack, Método de Sauvola e Pietaksinen, Método de Phansalskar, More e Sabale, Método do Contraste, Método da Média e por fim o Método da Mediana.

Para cada uma das imagens também disponibiliza-se métodos de verificação de proporção de pixels pretos e extração de histograma. Esta segunda parte, sendo disponibilizada em um arquivo a parte.

C. Parâmetros

Para a correta execução dos códigos em anexo, deve-se seguir o seguinte modelo de forma geral:

```
python3 trabalho2.py (INPUT) [OUTPUT] [OPCOES]
```

A partir do padrão acima, os elementos em parênteses são obrigatórios, os elementos entre colchetes podem expandir suas possibilidades de acordo com as necessidades de cada questão, com utilização opcional, que serão explicadas a seguir.

- 1) python3: Palavra chave para utilização da linguagem python, 3 edição, no sistema linux.
- 2) trabalho3.py: Parâmetro obrigatório, define o arquivo python a ser executado com os códigos para a solução

dos problemas, neste caso, trabalho3.py é o arquivo em questão.

- 3) INPUT: Especifica o arquivo imagem que será tratado através dos algoritmos. Também parâmetro obrigatório. Deve-se explicitar o comando Input através da palavra -i ou --ifile, o nome da imagem também deve especificar a extensão do arquivo. Neste trabalho, arquivos PGM (.pgm) estão sendo utilizados como entrada. Como exemplo:

- `python3 trabalho3.py -i baboon.pgm`

- 4) OUTPUT: Este parâmetro opcional é utilizado para especificar o nome do arquivo da imagem gerada após o processamento do algoritmos. O nome deve conter também o formato de saída, que neste trabalho utiliza-se como padrão formatos PGM (.pgm). Caso não informado, a imagem resultante terá o nome no formato, <Imagem><Método><Limiar><Tamanho da Vizinhança><K><R><p><q><Fração de Pixels Pretos Pré-Processamento><Fração de Pixels Pretos Pós-Processamento>.pgm, que representa a imagem selecionada, método, o limiar, tamanho da vizinhança utilizada e parâmetros K,R,p e q, selecionados pelo usuário quando necessário no método utilizado. Para definir o valor deste parâmetro, deve-se utilizar a palavra -o ou --ofile. Como possível caso de utilização, tem-se:

- `python3 trabalho3.py -i baboon.pgm -o saida.pgm`

- 5) OPCOES: Este parâmetro representa as múltiplas opções disponibilizadas entre as diferentes técnicas disponíveis no trabalho.

- Parâmetro: -m ou --method.

- Descrição:

Através deste primeiro parâmetro, pode-se selecionar qual método será utilizado, entre os métodos disponíveis, listados anteriormente, que foram solicitados e disponibilizados na descrição do trabalho 3. Os possíveis valores aceitos para este parâmetro são representados pelos nomes do método ou um número definido para o mesmo, como segue: Global (1), Otsu(2), Contrast (3), Bernsen (4), Niblack (5), SauvolaPietaksinen (6), PhansalskarMoreSabale (7), Mean (8) ou Median (9). Para selecionar este parâmetro, deve-se fornecer a palavra -m ou --method e o método a ser utilizado, nome ou número correspondente. Caso não selecionado nenhum método, o método Global é utilizado como padrão.

- `python3 trabalho3 -i fiducial.pgm -m Niblack`
- `python3 trabalho3 -i retina.pgm --method 4`

- Parâmetro: -v ou --invert.

- Descrição: Através deste parâmetro é possível selecionar a inversão de seleção entre objeto e fundo da imagem, desta forma, pixels que seriam considerados brancos são considerados pretos e vice-versa. Esta opção foi adicionada visto que para algumas

imagens dependem do contexto que estão inseridas. Utiliza-se então a palavra -v ou --invert seguido do valor booleano para esta variável. Como valor padrão, tem-se falso (0).

- `python3 trabalho3 -i monarch.pgm -v falso`
- `python3 trabalho3 -i baboon.pgm --invert 1`

- Parâmetro: -n ou --neighborhoodSize.

- Descrição: Este parâmetro representa o tamanho da vizinhança que será considerada para aplicação da técnica de limiarização. O valor aqui atribuído será então considerado como uma área $n \times n$, ou seja, caso n seja 3, tem-se um valor de vizinhança 3×3 . Para utilizar este parâmetro, utiliza-se a palavra -n ou --neighborhoodSize seguido do tamanho desejado para a vizinhança no método atual. Como valor padrão, o que não representa o melhor valor para as diferentes técnicas, firmou-se como 15.

- `python3 trabalho3 -i monarch.pgm -n 3`

- `python3 trabalho3 -i wedge.pgm --neighborhoodSize 15`

- Parâmetro: -t ou --threshold.

- Descrição: Este atributo é utilizado para selecionar qual o limiar utilizado para binarizar imagens através do método Global. Este parâmetro deve ser utilizado somente em conjunto com o método global (1), visto que este necessita que o limiar seja definido pelo usuário. Valores entre 0 e 255 serão aceitos como valores válidos para o limiar. Caso não selecionado previamente pelo usuário, o valor padrão deste parâmetro é 128.

- `python3 trabalho3 -i baboon.pgm -m global -t 100`

- `python3 trabalho3 -i retina.pgm -m 1 --threshold 186`

- Parâmetro: -k ou --kValue.

- Descrição: Este parâmetro define o valor da constante k para os métodos de Niblack, Sauvola e Pietaksinen, Phansalskar, More e Sabale. Esta constante é utilizada para multiplicar o desvio padrão que futuramente é somando com a média da vizinhança, porém para cada método pode-se ter uma variação tanto na sua utilização quanto no seu melhor valor para uma determinada imagem. Como valor padrão define-se k , neste trabalho, como -0.2, o que não representa o melhor valor de k .

- `python3 trabalho3 -i retina.pgm -m 3 -k 0.8`

- `python3 trabalho3 -i wedge.pgm -m 5 --kValue -0.8`

- Parâmetro: -r ou --rValue.

- Descrição: Este parâmetro define o valor da constante R para os métodos de Sauvola e Pietaksinen, Phansalskar, More e Sabale. Esta constante é utilizada para dividir o desvio padrão que futuramente é multiplicado por k e somando com a média da vizinhança, porém para cada método pode-se ter

uma variação tanto na sua utilização quanto no seu melhor valor para uma determinada imagem. Como valor padrão define-se r , neste trabalho, como 3, o que não representa o melhor valor de r .

- `python3 trabalho3 -i retina.pgm -m 4 -r 7`
- `python3 trabalho3 -i wedge.pgm -m 5 --rValue 3`
- Parâmetro: `-p` e `-q` ou `--pValue` e `--qValue`.
- Descrição: Estes dois parâmetros definem os valores das constantes p e q para o método de Phansalskar, More e Sabale. Esta constante é utilizada somente neste método como forma de modificar os valores do exponencial da média. Como valor padrão define-se p e q , neste trabalho, como 2 e 10 respectivamente, o que não representa o melhor valor de p e q para todas as imagens.
- `python3 trabalho3 -i monarch.pgm -m 5 -p 0.2 -q 0.8`
- `python3 trabalho3 -i fiducial.pgm -m 5 --pValue 3 --qValue 15`

Parâmetros (INPUT)	Palavra	Possibilidades	Opcionalidade
	<code>-i</code>	Nome da imagem de entrada (.pgm)	Obrigatória
[OUTPUT]	<code>-o</code>	Nome da imagem de saída	Opcional
Opções	<code>-m</code>	Seleção de métodos de Limiarização: Global (1), Otsu (2), Contrast (3), Bernsen (4), Niblack (5), SauvolaPietaksinen (6), PhansalskarMoreSabale (7), Mean (8) ou Median (9).	Opcional (Valor Padrão: método Global 1)
	<code>--method</code>		
	<code>-n</code>	Tamanho da vizinhança para aplicação do método. Tamanho $n \times n$ (n por n)	Opcional (Valor Padrão: 3)
	<code>--neighborhoodSize</code>		
	<code>-v</code>	Inversão de valores entre pixels pretos e brancos falso (0) ou verdadeiro (1)	Opcional (Valor Padrão: 0)
	<code>--invert</code>		
	<code>-t</code>	Atributo de limiar utilizado para binarizar imagens através do método Global.	Opcional (Valor Padrão: 128)
	<code>--threshold</code>		
	<code>-k</code>	Parâmetro k , utilizado nos métodos: Niblack (4), SauvolaPietaksinen (5), PhansalskarMoreSabale (6)	Opcional (Valor Padrão: -0.2)
	<code>--kValue</code>		
	<code>-r</code>	Parâmetro r , utilizado nos métodos: SauvolaPietaksinen (5), PhansalskarMoreSabale (6)	Opcional (Valor Padrão: 2)
	<code>--rValue</code>		
	<code>-p</code>	Parâmetro p , utilizado nos métodos: PhansalskarMoreSabale (6)	Opcional (Valor Padrão: 2)
	<code>--pValue</code>		
	<code>-q</code>	Parâmetro q , utilizado nos métodos: PhansalskarMoreSabale (6)	Opcional (Valor Padrão: 10)
	<code>--qValue</code>		

D. Exemplos de Execução

Para facilitar a execução dos códigos, segue alguns exemplos de linhas de código que pode ser utilizadas no terminal e algumas utilizações de parâmetros opcionais.

Método Global:

- `python3 trabalho3.py -i baboon.pgm`
- `python3 trabalho3.py -i -m 1 fiducial.pgm`

Método de Otsu:

- `python3 trabalho3.py -i -m 2 fiducial.pgm`

Método do Contrast:

- `python3 trabalho3.py -i monarch.pgm -o saida.pgm --method contrast -n 3`

Método de Bernsen:

- `python3 trabalho3.py -i peppers.pgm -m Bernsen --neighborhoodSize 7`

Método de Niblack:

- `python3 trabalho3.py -i peppers.pgm --method Niblack --n 15 --k -0.2`

Método de Sauvola e Pietaksinen:

- `python3 trabalho3.py -i peppers.pgm -m 6 -n 15 -k -0.2 -r 2`

Método de Phansalskar, More e Sabale:

- `python3 trabalho3.py -i retina.pgm -m 7 -n 9 -k -0.2 -r 2 -p 4 -q 16`

Método da Media:

- `python3 trabalho3.py -i sonnet.pgm -m 8 -n 7`

Método da Mediana:

- `python3 trabalho3.py -i peppers.pgm -m 9 -n 9`

III. IMPLEMENTAÇÃO E DECISÕES TOMADAS

Nesta seção será descrito o funcionamento dos algoritmos do trabalho 3 e alguns aspectos mais específicos de implementação e algumas decisões importantes.

De forma geral algumas decisões foram tomadas:

- Todas as entradas aceitam imagens tons de cinza.
- Imagens de diferentes tamanhos podem ser utilizadas. Não necessariamente matrizes quadradas ou com valores pares em linhas e colunas
- Os valores dos parâmetros opcionais, quando não declarados, exibem a imagem com valores padrões.
- Sempre que possível buscou-se utilizar métodos e equações em formato vetorizado.
- Para implementação deste trabalho, quase todos os métodos de limiarização foram implementados manualmente, sem utilizar nenhuma biblioteca. Apenas Otsu foi implementado utilizando a biblioteca `opencv2`.
- As bordas são ignoradas, não são tratadas pelo algoritmo e devem ser desconsideradas. Porém métodos de utilização de bordas, já estudadas em trabalhos anteriores poderiam ser utilizadas.

A. Métodos

Todos os métodos limiarização utilizados foram dispostos através do questionário do trabalho 3. Para implementá-los, como descrito anteriormente, todos foram criados a partir do zero, ou seja, sem a utilização de uma biblioteca destes métodos. Entre os métodos utilizados estão Global, Contrast, Bernsen, Niblack, Sauvola e Pietaksinen, Phansalskar, More e Sabale, Média e Mediana.

Estes métodos de limiarização são utilizados para limiar uma imagem, ou seja traçar limiares entre pixels. Neste trabalho este limiar será realizado em forma de binarização da imagem, desta forma, se um pixel $p(x, y)$ na imagem possui valor mais alto que o limiar, T , então o pixel $p(x, y)$ é classificado

como objeto (preto), caso contrário, ele é rotulado como fundo (branco).

Logo, uma imagem $g(x)$ pode ser definida como:

$$g(x, y) = \begin{cases} 0, & \text{se } f(x, y) \leq T \\ 1, & \text{se } f(x, y) > T \end{cases} \quad (1)$$

Para o cálculo do limiar as diferentes técnicas apresentam formas distintas. Que serão apresentadas:

- Métodos Globais:
- Método Global: Está é a única técnica implementada neste trabalho que apresenta uma limiarização de forma global, visto que Otsu foi utilizada pela biblioteca opencv2 e utilizada para comparação com a binarização básica aqui apresentada. Seguindo as seguintes condições de acordo com o limiar anteriormente definido: se o valor de intensidade de um pixel (x, y) for maior do que um limiar T , o pixel é considerado como parte do objeto, caso contrário, considerado como parte do fundo da imagem.
- Método de Otsu: Esta técnica foi adicionada ao trabalho para realizar comparações com a técnica global anteriormente apresentada, mesmo não sendo requerido no trabalho. Para Otsu é selecionado valores de limiar de acordo com a entropia. As entropias $H_b(T)$ e $H_w(T)$ associadas aos pixels do fundo e dos objetos, respectivamente, são dadas por:

$$h_b(T) = - \sum_{i=0}^T p_i \log_p i \quad (2)$$

$$h_w(T) = - \sum_{i=T+1}^{L-1} p_i \log_p i \quad (3)$$

onde, $\sum_{i=T+1}^{L-1} p_i = 1$, $p_i = \frac{n_i}{n}$, sendo n_i o número de pixels com nível de cinza i , p_i a probabilidade de um nível de cinza i na imagem, n o número de pixel total e L o número de níveis de cinza. Finalmente o limiar é calculado por:

$$T = \operatorname{argmax}[H_b(T) + H_w(T)] \quad (4)$$

- Métodos Locais: Para os métodos locais, que serão apresentados a seguir o limiar podem ser calculados através de uma área $n \times n$, ou seja, em uma única área da imagem.
- Método de Bernsen: O limiar é calculado para um determinado (x, y) da seguinte maneira:

$$T(x, y) = (z_{\min} + z_{\max})/2 \quad (5)$$

onde, z_{\min} e z_{\max} são valores mínimos e máximos de níveis de cinza em uma vizinhança $n \times n$ com centro no pixel (x, y)

- Método de Niblack: O limiar é calculado através da média local e no desvio padrão:

$$T(x, y) = \mu(x, y) + k\sigma(x, y) \quad (6)$$

onde, $\mu(x, y)$ é a média e $\sigma(x, y)$ o desvio padrão dos valores dos pixels em uma região $n \times n$ com centro em

(x, y) e k uma variável usada para ajuste da fração de borda.

- Método de Sauvola e Pietaksinen: Bom um aprimoramento da equação de Niblack, este método busca uma melhoria nos resultados adaptando o limiar de acordo com a média e desvio padrão local.

$$T(x, y) = \mu(x, y) \left[1 + k \left(\frac{\sigma(x, y)}{R} \right) - 1 \right] \quad (7)$$

onde, $\mu(x, y)$ e $\sigma(x, y)$ são definidos por Niblack, k e r variáveis adaptativos.

- Método de Phansalskar, More e Sabale: Mais uma variação das formulas anteriores, adaptadas para imagens de baixo contraste. Com limiar calculado por:

$$T(x, y) = \mu(x, y) \left[1 + \exp(-q\mu(x, y)) + k \left(\frac{\sigma(x, y)}{R} \right) - 1 \right] \quad (8)$$

onde, $\mu(x, y)$ e $\sigma(x, y)$ são definidos por Niblack. k , r , p e q variáveis adaptativos.

- Método do Contraste: O valor do pixel (x, y) é definido quanto a sua proximidade de valor em relação aos máximos ou mínimos dos valores locais dos pixels da região.
- Método da Média: O limiar é escolhido de acordo com a média local. Se o valor do pixel (x, y) for maior do que a média de sua vizinhança, o pixel será considerado parte do objeto, caso contrário, considerado parte do fundo da imagem.
- Método da Mediana: O limiar é escolhido de acordo com a mediana local. Se o valor do pixel (x, y) for maior do que a mediana de sua vizinhança, o pixel será considerado parte do objeto, caso contrário, considerado parte do fundo da imagem.

B. Histograma e Fração de Pixels Pretos

Neste trabalho, também considerou-se a verificação do histograma de imagens e a fração de pixels pretos nestas imagens. Para isto utilizou-se o arquivo python histogram.py.

Este arquivo python recebe como entrada uma imagem através das regras já apresentadas anteriormente para o trabalho3.py, através da palavra -i (INPUT) e apresenta um histograma da imagem recebida, assim como o valor de pixels pretos nesta imagem que é apresentada no nome do arquivo criado e na janela de apresentação.

IV. RESULTADOS

Para os resultados obtidos, e comparação entre estes, serão demonstrados e comparados os resultados de apenas uma técnica por vez, e após isto, com a apresentação de novas técnicas, as diferenças serão denotadas, quanto a melhor precisão e aproximação das técnicas.

Devido ao grande apelo visual, para cada método será demonstrado o resultado obtido e para cada um destes resultados. Todas as imagens apresentadas nesta seção foram obtidas através do algoritmo produzido e em anexo deste trabalho

A. Métodos

Método Global:

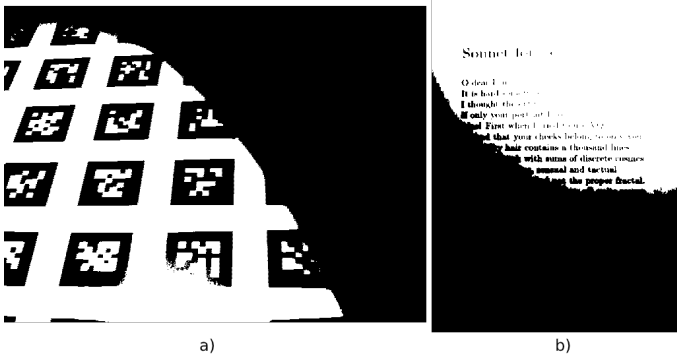


Fig. 1. a) Imagem: fiducial.pgm - Técnica: Global - Invertida: 0 - Threshold: 128 - Fração de Pixels Pretos: 61.97% b) Imagem: sonnet.pgm - Técnica: Global - Invertida: 0 - Threshold: 128 - Fração de Pixels Pretos: 52.05%

Pode-se observar, através da imagem 1 que o método de Limiarização global, assim como proposto, binariza os pixels considerando apenas seu valor em relação ao limiar estipulado pelo usuário, desta forma, imagens com diferenças de iluminação, como observado na imagem citada, têm uma binarização prejudicada. Observa-se que as regiões com pouca iluminação é considerada como objeto devido esta variação de iluminação. Os resultados podem ser observados tanto para a imagem a) fiducial quanto para b) sonnet.

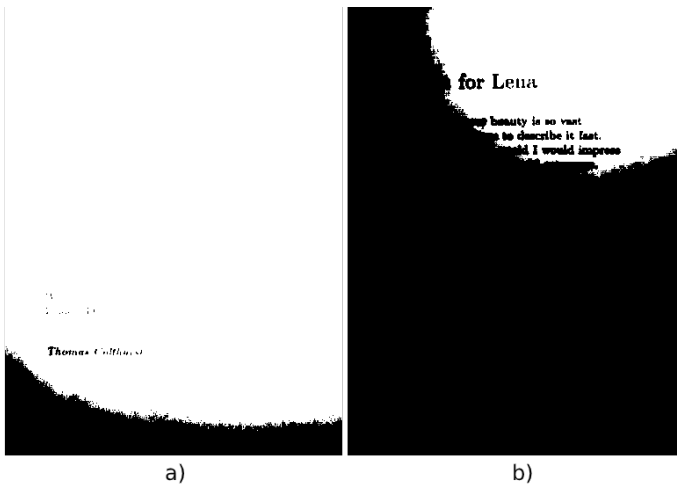


Fig. 2. Imagem: monarch.pgm - Técnica: Global - Invertida: 1 - Threshold: 128 - Fração de Pixels Pretos: 22.06%

Os resultados obtidos nas imagens da imagem 2 demonstram que a seleção do limiar é importante e que mal selecionado pode causar resultados muito piores que os alcançados anteriormente.

Apesar da simplicidade do método de limiarização Global, percebe-se que também apresenta bons resultados para imagens que têm uma alta diferença entre os valores dos pixel do



Fig. 3. Imagem: monarch.pgm - Técnica: Global - Invertida: 1 - Threshold: 128 - Fração de Pixels Pretos: 22.06%

fundo da imagem e do objeto, ou quando não há uma variação da iluminação na imagem, como demonstrado na imagem 3.



Fig. 4. Imagem: wedge.pgm - Técnica: Global - Invertida: 0 - Threshold: 128 - Fração de Pixels Pretos: 60.95%

Porém para imagens com variação de iluminação, onde o objeto que deseja-se detectar está sendo iluminado com intensidades diferentes de acordo com sua posição, observa-se resultados ruins de detecção, como observado na imagem 5.

Método de Otsu

Como mencionado anteriormente, o método de Otsu está sendo utilizado neste trabalho para efeitos de comparação com o método mais simples de limiarização global e com os outros métodos a seguir de limiarização local.

Pode-se verificar que os resultados são próximos dos observados anteriormente através da técnica de limiarização global anteriormente utilizada, visto que as duas técnicas

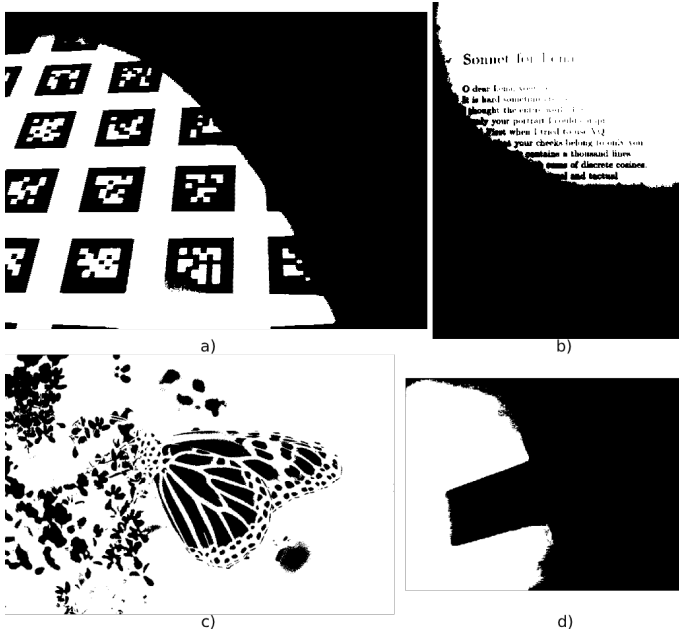


Fig. 5. a) Imagem: fiducial.pgm - Técnica: Otsu - Invertida: 0 - Fracao de Pixels Pretos: 64.13% b) Imagem: sonnet.pgm - Técnica: Otsu - Invertida: 0 - Fracao de Pixels Pretos: 57.16% c) Imagem: monarch.pgm - Técnica: Otsu - Invertida: 1 - Fracao de Pixels Pretos: 78.72% d) Imagem: wedge.pgm - Técnica: Otsu - Invertida: 0 - Fracao de Pixels Pretos: 66.38%

realizam o processo em um ambiente global da imagem. Desta forma, o método de Otsu tende a escolher o valor ótimo de separação dos valores presentes na imagem através do cálculo probabilístico, porém os resultados podem ser obtidos através da técnica anterior, apesar de ter que observar e testar várias vezes.

Método de Bernsen:

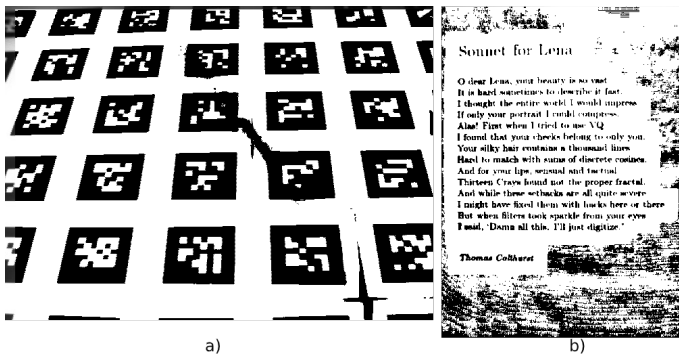


Fig. 6. a) Imagem: fiducial.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 43 - Fracao de Pixels Pretos: 33.68% b) Imagem: sonnet.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 43 - Fracao de Pixels Pretos: 21.58%

Percebe-se através da imagem 6 que ao aplicar-se o método de Bernsen tem-se um resultado bem melhor em relação à variação de iluminação na imagem, esta melhora dar-se devido ao calculo do limiar através de um método local.

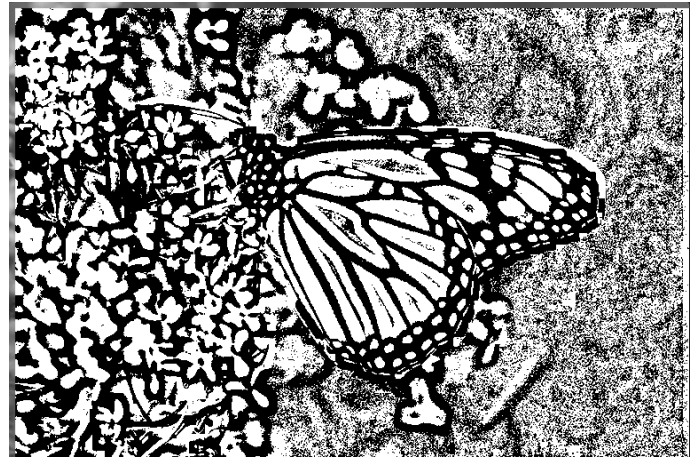


Fig. 7. Imagem: monarch.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 15 - Fracao de Pixels Pretos: 48.12%

Apesar dos bons resultados para imagens com diferenças de iluminação, este método não apresentou resultados tão bons para uma imagem com muito ruído ao fundo. Pode-se observar através da imagem 7 que apesar de ser perceptível uma separação entre os objetos, existe muito ruído no fundo da imagem.

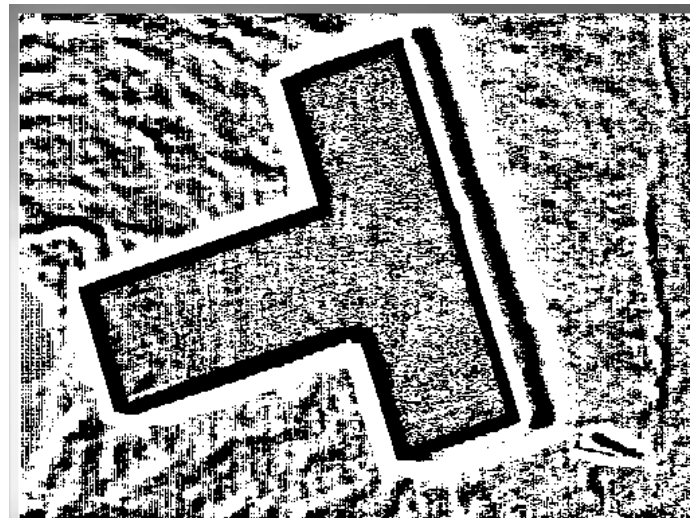


Fig. 8. Imagem: wedge.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 15 - Fracao de Pixels Pretos: 39.88%

Em relação à imagem 8, apesar do mesmo tipo de ruído encontrado na imagem anterior, que pode ser percebido também nesta imagem, ao fundo, observa-se que o objeto que deseja-se destacar, está bem separável do fundo.

Para explicitar a diferença e o impacto do uso e tamanho da região local o qual será realizado o limiar, apresenta-se a imagem 9. Nesta imagem pode-se observar a mesma imagem com diferentes valores de tamanho para a vizinhança n ($n \times n$). Na imagem a) tem-se uma vizinhança de 3×3 , b) 7×7 , c)

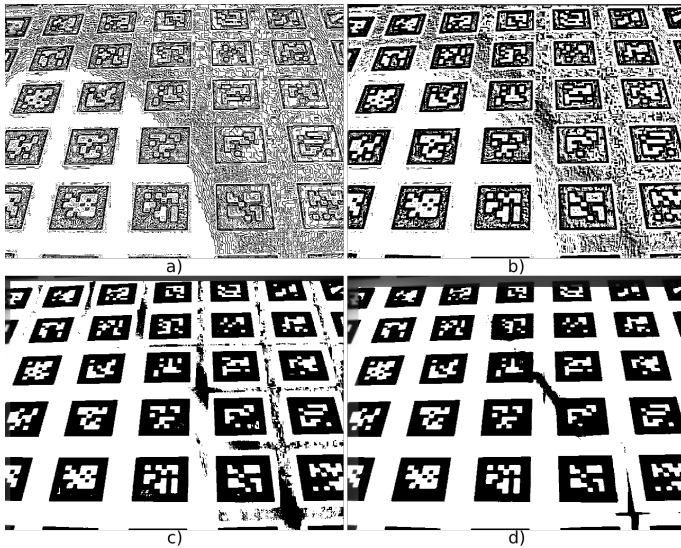


Fig. 9. a) Imagem: fiducial.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 3 - Fracao de Pixels Pretos: 32.89% b) Imagem: fiducial.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 7 - Fracao de Pixels Pretos: 37.18% c) Imagem: fiducial.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 21 - Fracao de Pixels Pretos: 38.39% d) Imagem: fiducial.pgm - Técnica: Bernsen - Invertida: 0 - NeighborhoodSize: 43 - Fracao de Pixels Pretos: 33.68%

21 x 21 e na imagem d) uma vizinhança de 43 x 43. Pode-se perceber que quanto menor a vizinhança mais segmentada será a imagem a partir de difentes pixels, enquanto que em imagens com uma maior vizinhança, tendem a ser mais homogêneas. Porém deve-se ter cuidado com vizinhanças muito grandes, que acabam gerando sombras onde não haviam.

Método de Niblack

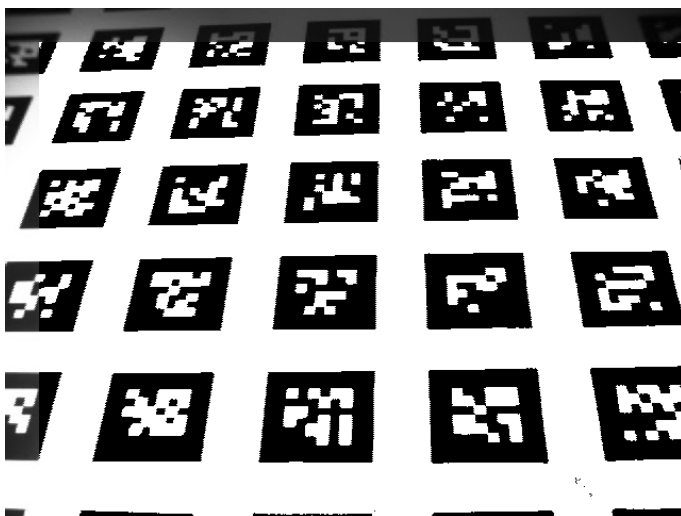


Fig. 10. Imagem: fiducial.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 65 - K: -0.2 - Fracao de Pixels Pretos: 30.55%

Como pode ser percebido através da imagem 10, o método de Niblack alcançou resultados muito bons em relação a imagem Fiducial. Percebe-se que é totalmente possível diferenciar os objetos que desejava-se destacar. Apesar da variação de iluminação, o método de Niblack conseguiu perceber os tons que estavam compondo cada porção da imagem

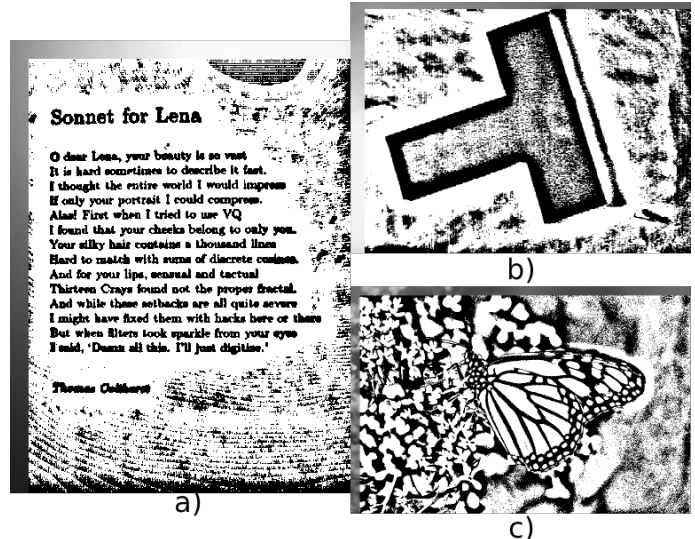


Fig. 11. a) Imagem: sonnet.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: -0.2 - Fracao de Pixels Pretos: 21.92% b) Imagem: wedge.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: -0.2 - Fracao de Pixels Pretos: 31.31% c) Imagem: monarch.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: -0.2 - Fracao de Pixels Pretos: 38.56%

Assim como no método anterior, percebe-se que os objetos que deseja-se destacar através da binarização podem ser observados. A imagem 11 demonstra a aplicação de Niblack em outras imagens. Ainda é possível observar bastante ruído em algumas imagens, porém os objetos estão bem destacados em a) e b). Quanto a imagem d), percebe-se que estes métodos não conseguem distinguir muito bem imagens com muito ruído, o que resulta em uma binarização dos objetos e fundo, porém ainda são exibidas muitas manchas onde deveria ser apenas fundo.

Para enfatizar as diferenças resultantes entre a utilização da variável K, a imagem 12 é resultante da interação entre a imagem Monarch e diferentes valores de K com os outras variáveis mantidas como constante. Uma vez que K multiplica a variância dos pixels x,y em uma determinada região, quanto menor seu valor, menos será o impacto da variância destes pixels na imagem, o contrário também é válido.

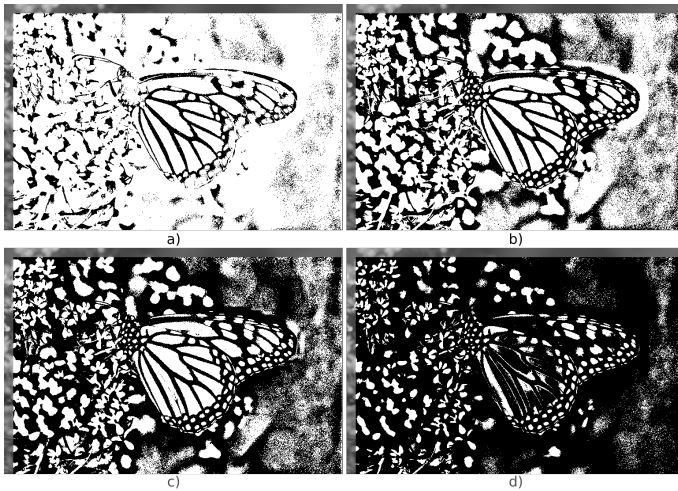


Fig. 12. a) Imagem: monarch.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: -0.8 - Fracao de Pixels Pretos: 14.76% b) Imagem: monarch.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: -0.2 - Fracao de Pixels Pretos: 38.56% c) Imagem: monarch.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: 0.2 - Fracao de Pixels Pretos: 57.21% d) Imagem: monarch.pgm - Técnica: Niblack - Invertida: 0 - NeighborhoodSize: 43 - K: 0.8 - Fracao de Pixels Pretos: 78.66%

Método de Sauvola e Pietaksinen

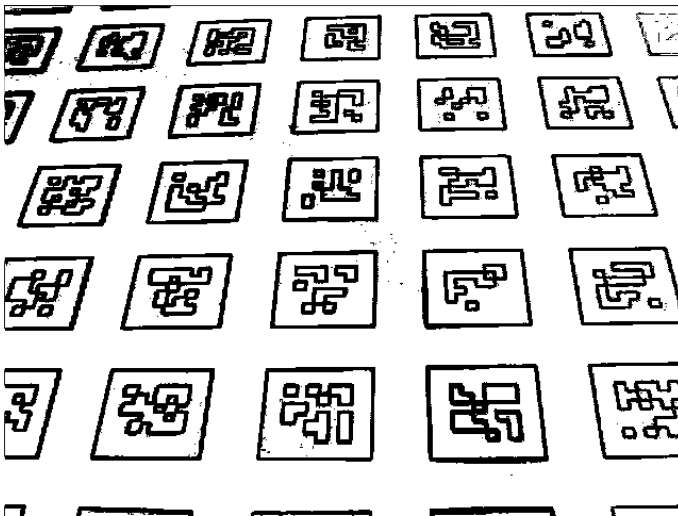


Fig. 13. Imagem: fiducial.pgm - Técnica: Sauvola e Pietaksinen - Invertida: 0 - NeighborhoodSize: 3 - K: 0.5 - R: 5 - Fracao de Pixels Pretos: 19.30%

Como observado na imagem 14, o método de Sauvola e Pietaksinen apresentam ótimos resultados para binarização nesta imagem. É possível perceber que as mudanças de iluminação não têm baixa influencia no processo de binarização com este método nesta imagem, os objetos ficaram muito bem separados e nítidos. Também é interessante observar que o tamanho da vizinhança utilizada para alcançar este resultado foi muito menor que nos métodos anteriores, o que

demonstra uma ótima robustez as mudanças de tons mesmo considerando uma área de imagem muito pequena.

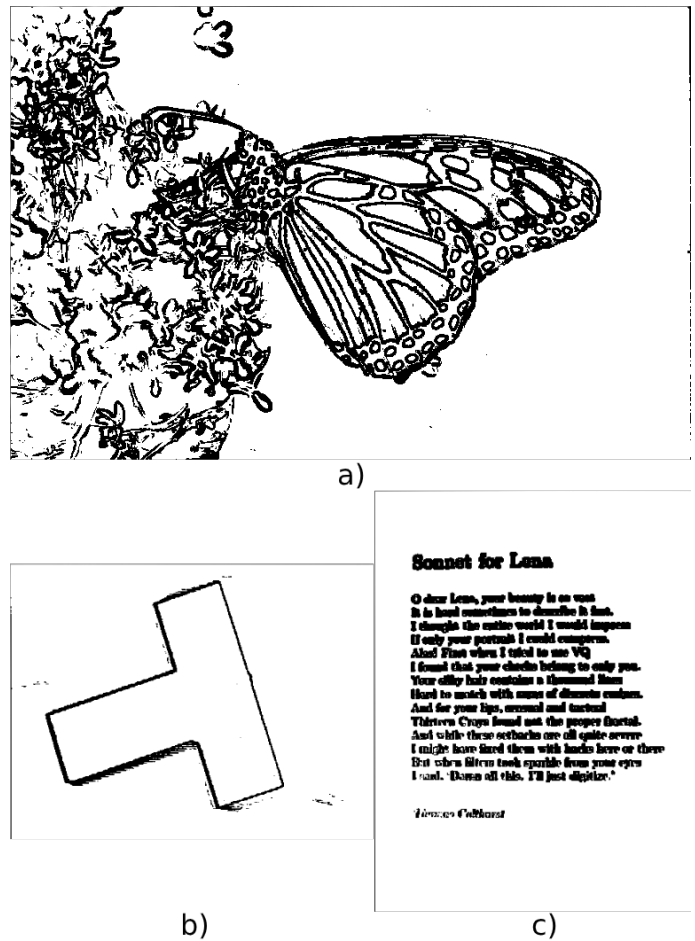


Fig. 14. a) Imagem: monarch.pgm - Técnica: Sauvola e Pietaksinen - Invertida: 0 - NeighborhoodSize: 3 - K: 0.5 - R: 7 - Fracao de Pixels Pretos: 21.43% b) Imagem: wedge.pgm - Técnica: Sauvola e Pietaksinen - Invertida: 0 - NeighborhoodSize: 3 - K: 0.5 - R: 3 - Fracao de Pixels Pretos: 3.58% c) Imagem: sonnet.pgm - Técnica: Sauvola e Pietaksinen - Invertida: 0 - NeighborhoodSize: 3 - K: 0.5 - R: 5 - Fracao de Pixels Pretos: 16.44%

O método de Sauvola e Pietaksinen também apresenta bons resultados nas outras imagens testadas neste trabalho. Mesmo em imagens com muito ruído, a), no qual algumas outras técnicas tiveram dificuldade de remover, através de Sauvola e Pietaksinen, pode-se perceber uma boa diferenciação entre fundo e objeto, observando-se também muito pouco ruído na imagem final. Para as outras imagens b) e c), foi possível extrair o objeto completamente da imagem, também com pouco ruído. Observa-se também que todos estes resultados foram obtidos com uma vizinhança muito pequena, demonstrando a capacidade deste método em lidar com variação de iluminação mesmo com pequenas áreas de amostras.

Não será realizado a demonstração de variação da variável R visto que os resultados são próximos aos observados na variação da variável K, uma vez que as duas variáveis agem diretamente na variância dos pixels de uma área. Enquanto K

multiplica esta variância, R divide a mesma.

Phansalskar, More e Sabale

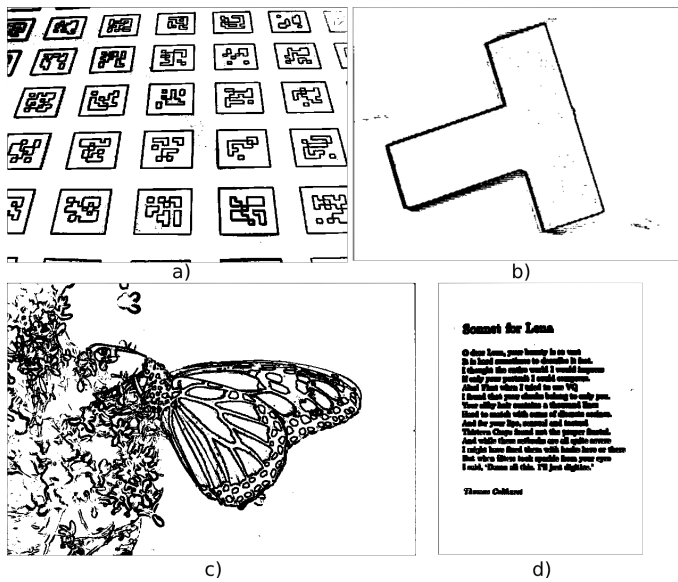


Fig. 15. a) Imagem: fiducial.pgm - Técnica: Phansalskar, More e Sabale - Invertida: 0 - NeighborhoodSize: 3 - K: 0.8 - R: 5 - P: 2 - Q: 10 - Fracao de Pixels Pretos: 19.11% b) Imagem: wedge.pgm - Técnica: Phansalskar, More e Sabale - Invertida: 0 - NeighborhoodSize: 3 - K: 0.2 - R: 3 - P: 2 - Q: 10 - Fracao de Pixels Pretos: 3.61% c) Imagem: monarch.pgm - Técnica: Phansalskar, More e Sabale - Invertida: 0 - NeighborhoodSize: 3 - K: 0.8 - R: 7 - P: 2 - Q: 10 - Fracao de Pixels Pretos: 21.38% d) Imagem: sonnet.pgm - Técnica: Phansalskar, More e Sabale - Invertida: 0 - NeighborhoodSize: 3 - K: 0.8 - R: 3 - P: 2 - Q: 10 - Fracao de Pixels Pretos: 19.61%

Como esperando do método de Phansalskar, More e Sabale, uma extensão do método de Sauvola e Pietaksinen apresenta os resultados da imagem 15, onde pode-se observar que os objetos foram muito bem segmentados do plano de fundo da imagem, mesmo para imagens com variação de iluminação ou com muito ruído. Também não foi necessário valores maiores que 3 de n, necessitando assim de uma área bem reduzida para realizar o processo, o que economiza em complexidade computacional.

Quanto aos valores das variáveis p e q, estas são utilizadas para controlar a variação da média da região, sendo q multiplicado pela média e p multiplica a média exponencial. Desta forma, não será demonstrado sua utilização, visto que outras variáveis já demonstraram a variação da sobreposição da variância sobre a média, neste caso, pode-se dar mais peso a média e ter um melhor controle sobre o resultado final, utilizando-se p e q como um mecanismo para este controle sobre o resultado que deseja-se alcançar.

Método do Contraste

Pode-se perceber através desta técnica resultados muito parecidos (na verdade iguais) com o método de Bernsen. Desta forma, assim como discutido anteriormente na seção do método 3, esta técnica apresenta um resultado razoável em relação a segmentação de imagens com variação de

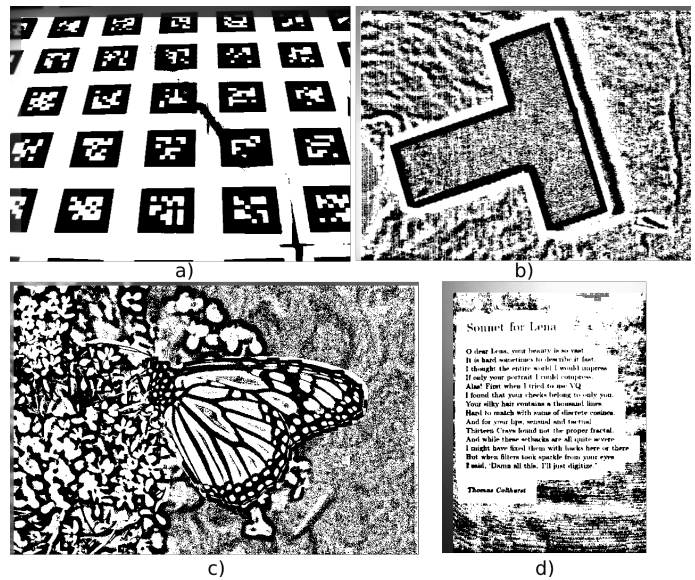


Fig. 16. a) Imagem: fiducial.pgm - Técnica: Contraste - Invertida: 0 - NeighborhoodSize: 43 - Fracao de Pixels Pretos: 33.68% b) Imagem: wedge.pgm - Técnica: Contraste - Invertida: 0 - NeighborhoodSize: 15 - Fracao de Pixels Pretos: 39.88% c) Imagem: monarch.pgm - Técnica: Contraste - Invertida: 0 - NeighborhoodSize: 15 - Fracao de Pixels Pretos: 48.12% d) Imagem: sonnet.pgm - Técnica: Contraste - Invertida: 0 - NeighborhoodSize: 43 - Fracao de Pixels Pretos: 21.58%

iluminação, imagens a), b), d) da figura 16, porém com melhores resultados que métodos de segmentação global. Porém para imagens com muitos detalhes e ruídos estes métodos não conseguem um bom resultado, visto que muitas manchas aparecem na imagem final, onde deveria ser considerado somente fundo, observados da imagem 16 c).

Para demonstrar que as imagens resultantes entre os métodos de Bernsen e do Contraste são idênticas, realizou-se a diferenças entre estas imagens 6 a) e 16 a), que resulta na imagem 17. Pode-se observar que todos os pixels resultantes têm valor 0, devido a diferença entre os pixels.

Média

O método da média, como demonstrado na imagem 18 tem um bom resultado quando a variação de iluminação em imagens que não têm muito ruído ou detalhes. Apesar destes resultados, este método depende muito de grandes valores de vizinhança, apresentando melhores resultados com maiores valores de n.

Na imagem 19 pode-se observar outros resultados para este método, nestes verifica-se que existe bastante ruído ao fundo da imagem. Estas manchas podem ser amenizadas com o tamanho de n, porém não chegam a apresentar resultados tão bons quando alguns dos métodos anteriormente apresentados.

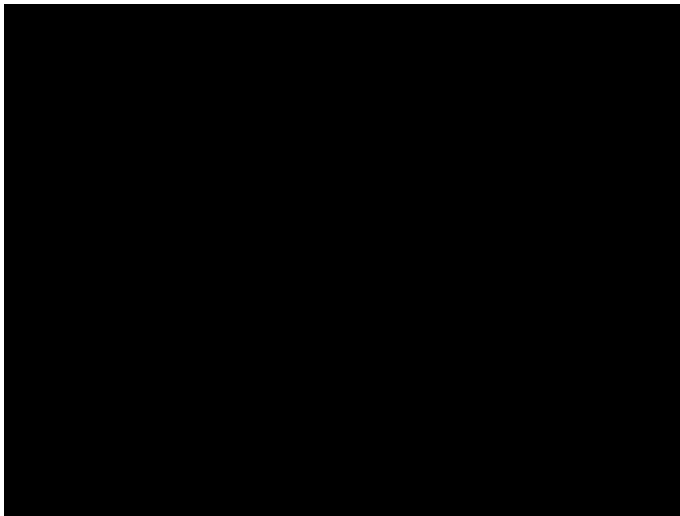


Fig. 17. Diferença entre as imagens 6 a) e 16 a)

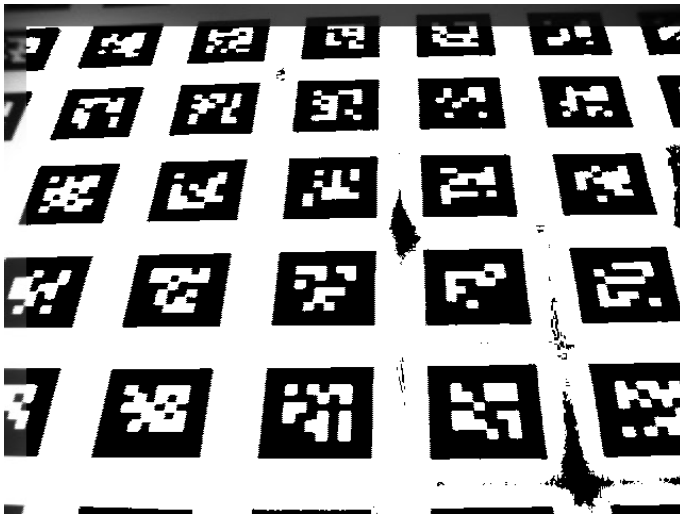


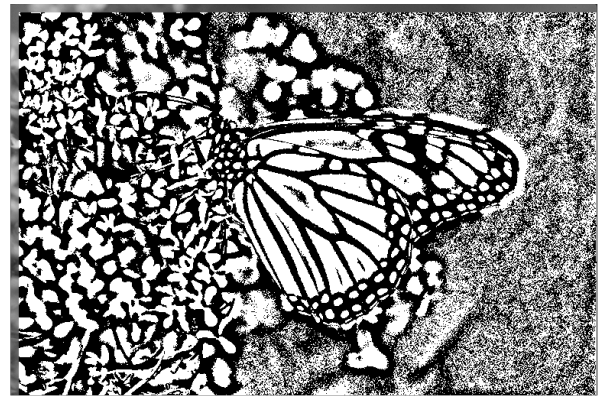
Fig. 18. Imagem: fiducial.pgm - Técnica: 8 - Invertida: 0 - NeighborhoodSize: 43 - Fracao de Pixels Pretos: 33.82%

Mediana

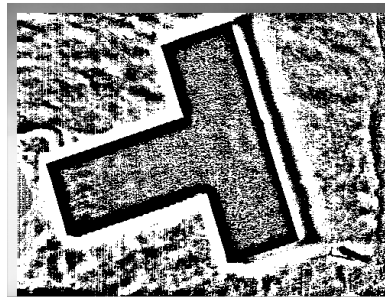
O método das Medianas, imagem 20, apresenta resultados bons em relação aos métodos globais, porém fracos se comparados ao métodos locais, visto que para alcançar resultados aceitáveis é necessário uma área muito grande, tamanho de n. Observa-se também que o resultado desta técnica apresenta muito ruído e manchas na imagem resultante. Os melhores resultados encontrados foram com a imagem fiducial, para segmentação de textos, que apesar de ainda conter muito ruído, imagem 20 d), apresenta um texto legível e bem segmentado.

B. Porcentagem de Preto

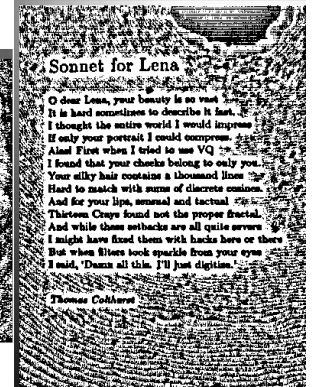
Apresentou-se todos os métodos e para cada imagem observada também adicionou-se a legenda a porcentagem de preto



a)



b)



c)

Fig. 19. a)Imagem: monarch.pgm - Técnica: Média - Invertida: 0 - NeighborhoodSize: 23 - Fracao de Pixels Pretos: 49.19 b)Imagem: wedge.pgm - Técnica: Média - Invertida: 0 - NeighborhoodSize: 27 - Fracao de Pixels Pretos: 45.17% c)Imagem: sonnet.pgm - Técnica: Média - Invertida: 0 - NeighborhoodSize: 15 - Fracao de Pixels Pretos: 41.13%

nesta imagem. Para alcançar este resultado, pode-se utilizar o arquivo python trabalho3.py ou histogram.py.

Percebeu-se que para cada imagem um conjunto de pixel compõe o objeto, estes pixels são a quantidade mínima de pixels que representam tais componentes. Desta forma, pode-se utilizar esta informação como qualidade para segmentação em objetos e fundo da imagem. Em muitos métodos, têm-se um alto valor de porcentagem visto que muito ruído é introduzido na imagem final, enquanto que para os métodos mais otimizados, as taxas de porcentagem de preto mantêm-se baixas, aproximando-se da quantidade mínima de pixels necessário para a sua representação.

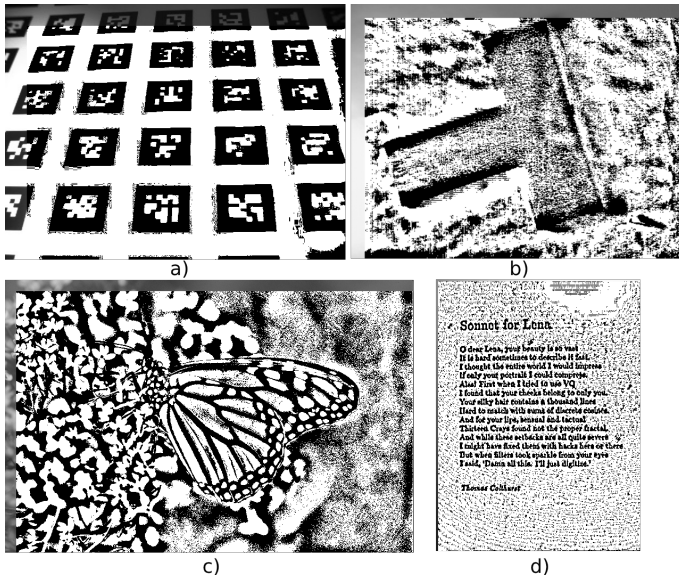


Fig. 20. a)Imagem: monarch.pgm - Técnica: Média - Invertida: 0 - NeighborhoodSize: 23 - Fracao de Pixels Pretos: 49.19 b)Imagem: wedge.pgm - Técnica: Média - Invertida: 0 - NeighborhoodSize: 27 - Fracao de Pixels Pretos: 45.17% c)Imagem: sonnet.pgm - Técnica: Média - Invertida: 0 - NeighborhoodSize: 15 - Fracao de Pixels Pretos: 41.13%

C. Histogramas

Quanto aos histogramas de cada imagem de entrada, como pedido na especificação, segue a imagem geradora e seu histograma:

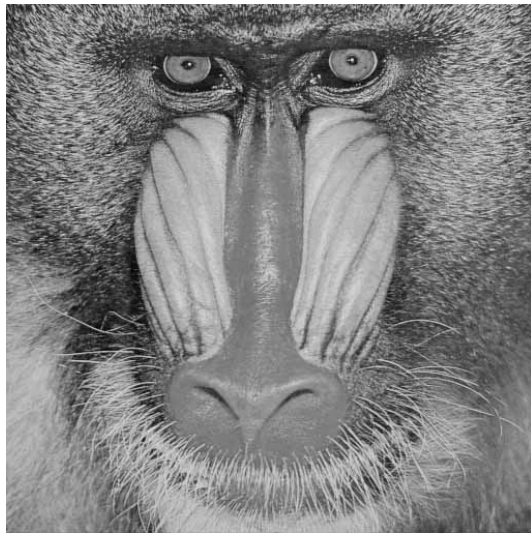


Fig. 21. Imagem Baboon.png

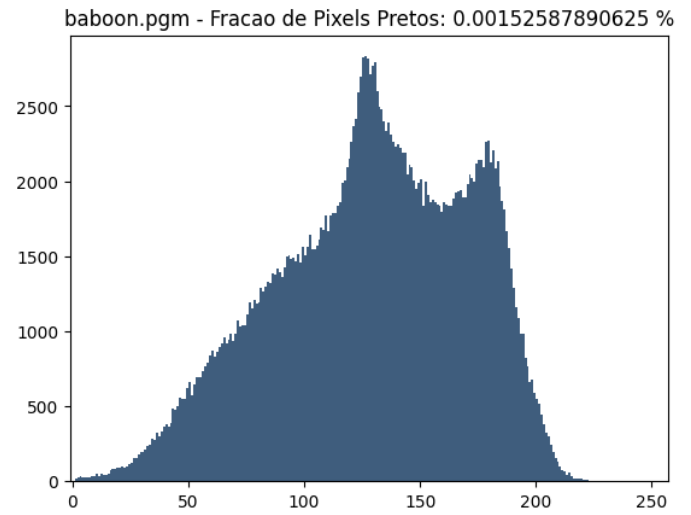


Fig. 22. Histograma da imagem Baboon.png

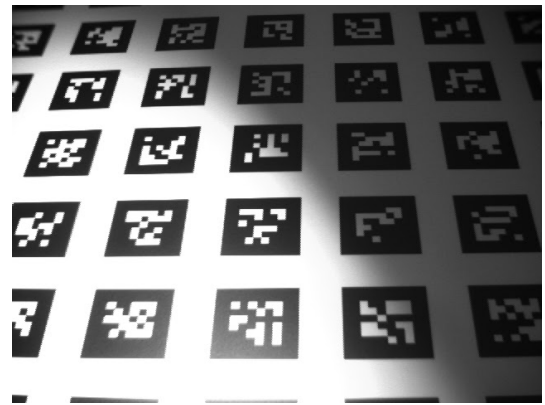


Fig. 23. Imagem fiducial.png

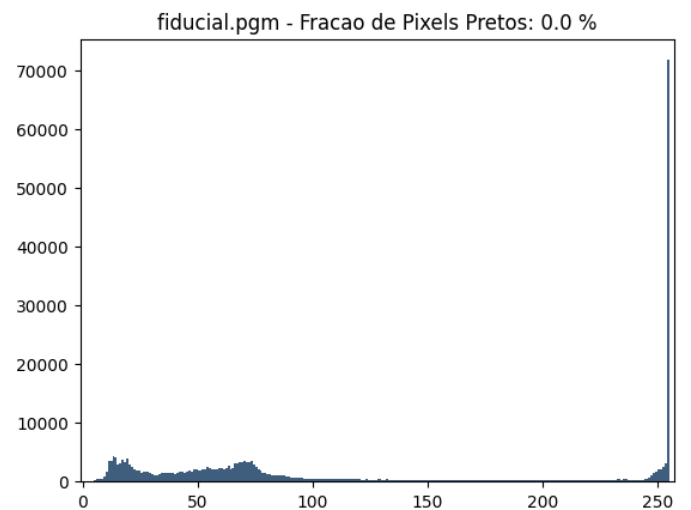


Fig. 24. Histograma da imagem fiducial.png



Fig. 25. Imagem monarch.png

peppers.pgm - Fracao de Pixels Pretos: 0.0016213499359566776 %

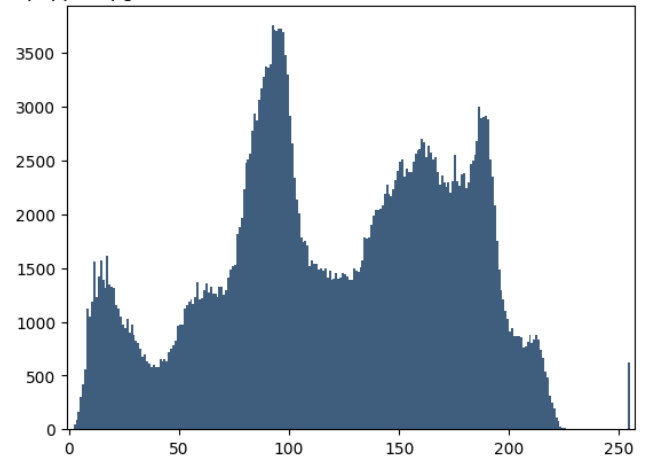


Fig. 28. Histograma da imagem peppers.png

monarch.pgm - Fracao de Pixels Pretos: 0.0 %

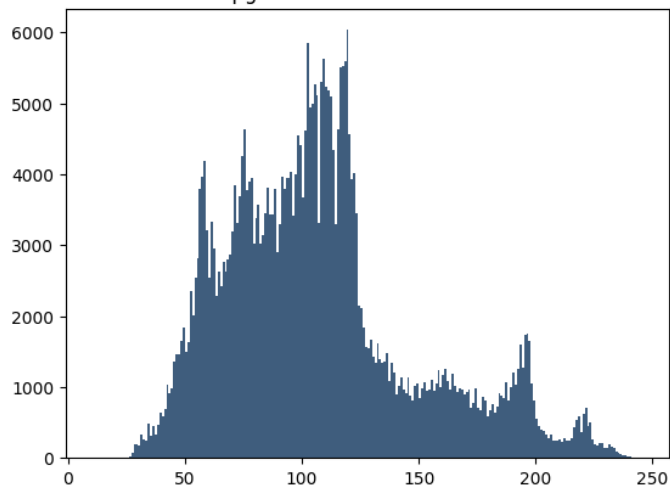


Fig. 26. Histograma da imagem monarch.png

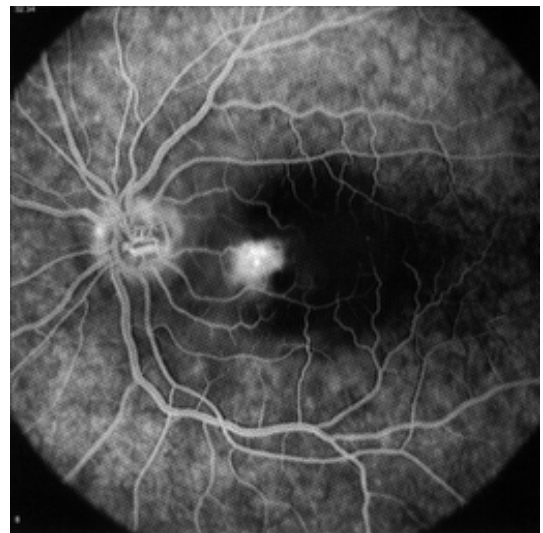


Fig. 29. Imagem retina.png



Fig. 27. Imagem peppers.png

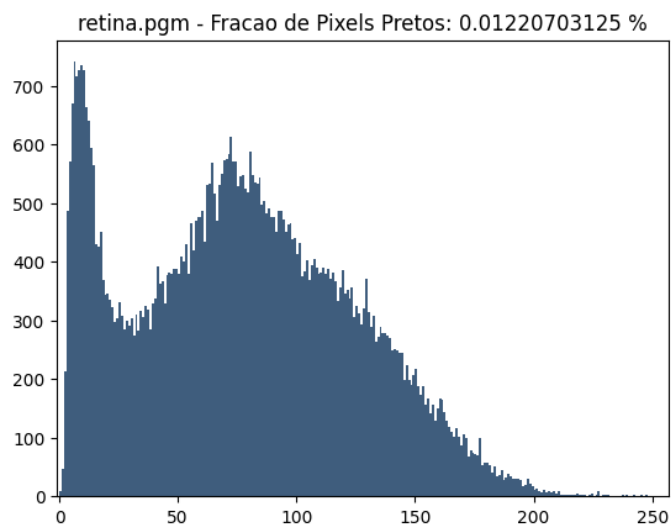


Fig. 30. Histograma da imagem retina.png

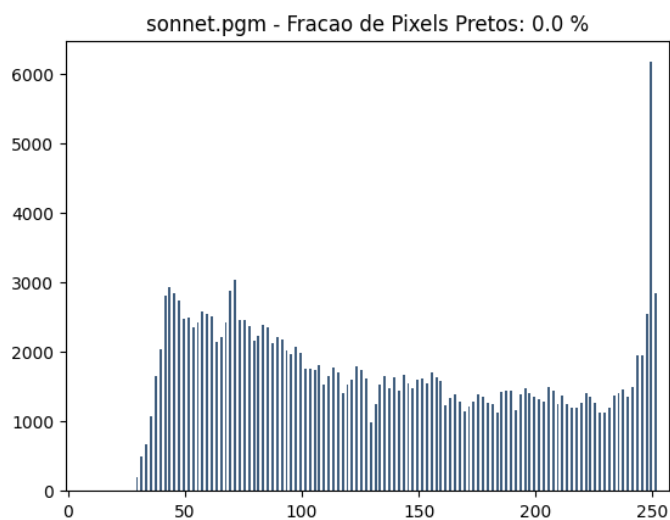


Fig. 32. Histograma da imagem sonnet.png

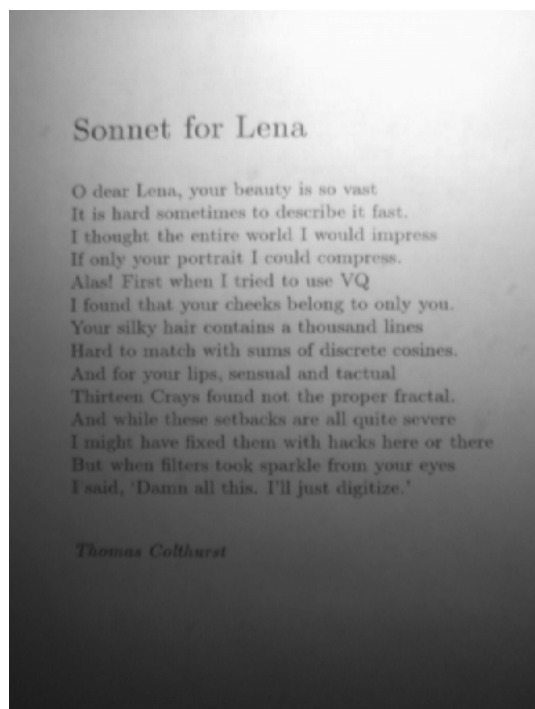


Fig. 31. Imagem sonnet.png

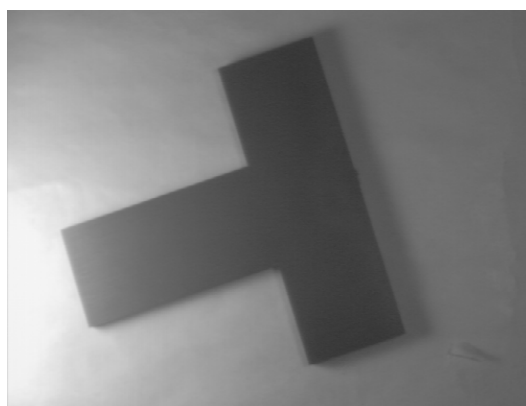


Fig. 33. Imagem wedge.png

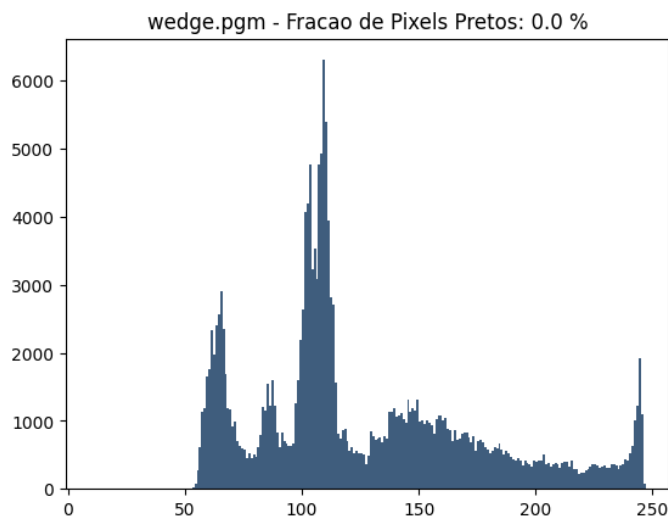


Fig. 34. Histograma da imagem wedge.png

V. CONCLUSÕES

Neste trabalho, estudou-se e implementou-se técnicas de Limiarização para segmentação de imagens. Observou-se o comportamento de 9 diferentes técnicas, suas capacidade de binarização e como estes métodos lidavam com variações de luz e ruído.

Conclui-se que métodos de limiarização locais têm uma maior capacidade de segmentação para ambientes de variação de iluminação, sendo métodos mais robustos. Entre as técnicas de segmentação local, algumas são melhor aplicadas em ambientes com ruído, outras em ambientes com variação de iluminação. Porém de modo geral, o método de Phansalskar, More e Sabale condensa as tecnicas anteriores dando uma maior possibilidade de controle, tanto em relação a média quando a variância das imagens.