



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

INTRODUÇÃO AO PROCESSAMENTO DIGITAL DE IMAGEM

Aluno: Kleber Kruger - RA: 230226

Métodos de Limiarização

Setembro de 2019

1 Introdução

Este relatório descreve o segundo trabalho realizado como avaliação para a disciplina de Introdução ao Processamento de Imagem Digital (MO433). O objetivo foi implementar algoritmos de limiarização global e local em imagens monocromáticas. O relatório está dividido da seguinte maneira: a Seção 2 fundamenta a técnica de limiarização; a Seção 3 apresenta o programa implementado neste trabalho e explica como utilizá-lo. Na Seção 4 são descritos os experimentos realizados juntamente a algumas considerações importantes.

Os códigos e os resultados foram submetidos pela plataforma Google Classroom. O nome do arquivo é: kleber-kruger.zip. Descompactado é composto por três pastas: in (com as imagens de entrada utilizada nos testes), out (com os resultados do processamento) e src (código do programa).

2 Limiarização

Limiarização é um processo de segmentação de imagens que se baseia na diferença dos níveis de cinza que compõe diferentes objetos de uma imagem. A partir de um limiar estabelecido de acordo com as características dos objetos que se quer isolar, a imagem pode ser segmentada em dois grupos: o grupo de pixels com níveis de cinza abaixo do limiar e o grupo de pixels com níveis de cinza acima do limiar. Em uma imagem limiarizada, atribui-se um valor fixo para todos os pixels de mesmo grupo.

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

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

Figura 1: Fórmula de limiarização

A imagem $g(x, y)$ será binária, ou seja, terá somente dois valores possíveis para cada pixel.

A escolha do valor de limiar pode ser global ou local. A limiarização global define um único valor de limiar para toda a imagem, enquanto que métodos de limiarização adaptativa local calcula um limiar para cada pixel da imagem com base na informação contida em uma vizinhança do pixel. Em ambos os casos, se um pixel (x,y) na imagem de entrada possui valor mais alto que o limiar, então o pixel (x,y) é classificado como objeto, caso contrário é rotulado como fundo.

É importante observar que a escolha de um limiar global pode não apresentar bons resultados em todos os casos, como por exemplo, imagens com condições de iluminação diferentes em partes distintas. Neste caso, o limiar adaptável local tende a oferecer melhores resultados, pois o algoritmo define o limite para um pixel com base na pequena região ao seu redor, ou seja, gerando limiares diferentes para regiões distintas da mesma imagem.

3 Implementação do Programa

3.1 Bibliotecas Utilizadas

Este trabalho foi implementado na linguagem Python 3.7. Utilizou-se a biblioteca multiplataforma e *open-source* chamada OpenCV, amplamente empregada em aplicações de processamento de imagens e de visão computacional. Seu uso foi para fazer o carregamento e a escrita das imagens no disco rígido. A biblioteca Numpy também foi utilizada. Esta oferece suporte para vetores e matrizes, além de conter um grande número de funções otimizadas para operar em elementos multidimensionais.

Para obter informações dos arquivos no *file system*, utilizamos as bibliotecas `os`, `glob` e `pathlib`. A biblioteca `matplotlib` para fazer a plotagem dos histogramas.

Os métodos de limiarização implementados foram: o método global, o método de Bernsen, de Niblack, de Sauvola e Pietaksinen, de Phansalskar More e Sabale, o método de contraste, da média e da mediana (todos descritos no enunciado).

3.2 Instruções de Uso

O programa deve ser executado com os seguintes parâmetros:

- **-input/-in/-i:** caminho para o(s) diretórios ou a(s) imagem(s) de entrada.
- **-output/-out/-o:** caminho para o(s) diretórios ou a(s) imagem(s) de saída. A nomenclatura padrão para os arquivos de saída foram: <nome-original>-out-<tipolimiarização><parâmetro>.
- **-global/-g:** quando usado, define valor(es) para limiarização global.
- **-method/-m:** método usado para limiarização local. As opções são:
 0. Bernsen
 1. Niblack
 2. Sauvola e Pietaksinen
 3. Phansalskar More e Sabale
 4. Contraste
 5. Média
 6. Mediana
- **-invert/-v:** Define a cor para o fundo e o objeto:
 0. Fundo branco, objeto preto
 1. Fundo preto, objeto branco

Os valores das constantes dos métodos de limiarização local foram os sugeridos no enunciado do trabalho. Escolhemos uma máscara de vizinhança de tamanho 3x3, considerando a quantidade de operações efetuadas e o custo de processamento.

Regras para os parâmetros:

0. A quantidade de valores para o parâmetro <input> é de 0 (não informado) até n;
1. Os valores de cada <input> é o caminho para uma imagem ou um diretório;
2. Caso o valor de um <input> seja um diretório, apenas as imagens na sua raiz serão processadas;
3. Caso não informe nenhum valor, <input> será definido com 'in' (diretório de entrada padrão);
4. A quantidade de valores para o parâmetro <output> é 0 (não informado), 1 ou n (mesma quantidade de <input>);
5. Os valores de cada <output> é o caminho para um diretório de saída ou o nome de um arquivo resultante;
 - (a) **Observação:** Se o processamento de uma imagem de entrada é feito de k maneiras (aplicação de filtros, algoritmos, etc), cada imagem de entrada resultará em k imagens de saída, e o nome do arquivo seguirá a nomenclatura padrão conforme o nome de saída especificado;
 - (b) Se nenhum nome foi especificado para o arquivo de saída, este seguirá a nomenclatura padrão conforme o nome original da imagem.
6. Um <output> associado a um <input> diretório, também é interpretado como um diretório, pois o resultado do processamento pode resultar em múltiplas imagens;
7. Quando <output> não for informado pelo usuário:
 - (a) Se <input> também não foi: input = in, output = out;
 - (b) Caso o <input> tenha sido definido pelo usuário:
 - i. Se <input> for o caminho para uma única imagem, o valor de <output> seguirá a nomenclatura padrão, e a imagem de saída será salva no mesmo diretório da imagem de entrada;

- ii. Se `<input>` for o caminho para um único diretório, `<output>` receberá o mesmo valor de `<input>`, e as imagens serão salvas conforme a nomenclatura padrão
 - iii. Se `<input>` for um conjunto de valores (imagens e/ou diretórios), cada valor de `<input>` respeitará as duas sub-regras anteriores.
8. Se a lista de `<input>` > 1 :
- (a) `<output>` poderá estar vazio (não informado). Cada input seguirá a regra 8.2;
 - (b) `<output>` poderá ter uma única entrada (obrigatoriamente um diretório de saída); - **Observação importante:** arquivos com mesmo nome resultante serão sobrescritos.
 - (c) `<output>` poderá ter n entradas (em que ambas as listas têm o mesmo tamanho, pois os valores de `<input>``<output>` serão atribuídos na mesma ordem; cada elemento da lista deve respeitar a regra padrão.
9. Caso apenas `<output>` tenha sido definido, `<input>` recebeu o valor padrão "in". Portanto, o valor de `<output>` precisa ser o caminho para um único diretório de saída;
10. `<types>` define os tipos (extensões) de arquivos que serão incluídos na lista de imagens de entrada
11. Se nenhum valor é informado pelo usuário, `types = '*'` (todos os tipos de imagens)

3.3 Funções do Programa

O programa é composto pelas seguintes funções:

- **main:** função inicial do programa que a partir dos argumentos de entrada (input, output, type, global, method e invert) executa o algoritmo de limiarização conforme as configurações recebidas.

- **parse_args:** função responsável por analisar e extrair os argumentos de entrada do programa, tais como as imagens de entrada, os destinos das imagens resultantes, os valores de limiarização global, os métodos de limiarização local e a configuração de objeto e fundo;
- **get_arg_output:** obtém o valor `-output` dos parâmetros de inicialização ou define o valor padrão;
- **get_output_file_name:** define o nome do arquivo de saída de acordo com o nome original do arquivo;
- **get_arg_file_fullname:** retorna o nome do arquivo de saída conforme a regra de nomenclatura padrão;
- **get_images_in_directory:** retorna a lista de imagens em um determinado diretório, considerando as extensões permitidas;
- **map_in_out:** mapeia cada arquivo de entrada em um arquivos de saída
- **get_arg_globs_methods:** obtém os valores `-global` e `-method` dos parâmetros de inicialização
- **get_args:** obtém todos os parâmetros de inicialização do programa
- **bernsen, niblack, sauvola_pietaksinen, phansalskar_more_sabale:** métodos utilizados para calcular a limiar de um pixel (x,y)
- **contrast, mean, median:** métodos que definem o valor do limiar conforme o cálculo do contraste, média e mediana, respectivamente;
- **threshold_local:** aplica o algoritmo de limiarização local em uma imagem de entrada;

- **threshold_global:** aplica o algoritmo de limiarização global em uma imagem de entrada;
- **__execute:** executa o procedimento de carregar uma imagem, chamar o algoritmo de limiarização e plotar o histograma;
- **execute:** encapsula o método `__execute` para fazê-lo em um conjunto de várias imagens;
- **run_task:** executa qualquer função medindo o seu tempo de execução

4 Resultados

4.1 Entradas

Na pasta `src`, junto ao código do programa, existe um arquivo `test.sh`. Este arquivo descreve as execuções do programa durante os experimentos. Foram testadas com imagens de diferentes tamanhos, mas para não onerar o tempo de execução do teste, mantivemos as imagens de exemplo disponibilizadas no enunciado (http://www.ic.unicamp.br/~helio/imagens_pgm/) Os parâmetros de configuração para cada imagem de entrada foram: `-global 64 128 192 -method 0 1 2 3 4 5 6`. Ou seja, para todas as imagens, executou-se o método de limiarização global baixo, alto e intermediário, e todos os métodos de limiarização locais.

4.2 Saídas

Na pasta `out` é possível visualizar o resultado dos experimentos com as imagens de entrada fornecidas. Na imagem 3(a) a classe de fundo foi definida com o valor preto e a classe de objeto com o valor branco, e a imagem 3(b) o fundo e o objeto foram definidos com os valores inversos.

Figura 2: Duas configurações para as cores de fundo e do objeto

Na Figura 3 são mostrados os resultados da comparação da técnica de limiarização global com os parâmetros: 64, 128 e 192. Nesta, o limiar 128 apresentou a melhor definição, entretanto, nas imagens apresentadas na Figura 4 o melhor limiar varia conforme a região observada da imagem. Ou seja, não há um valor superior a um valor global adequado, pois o melhor resultado varia conforme a imagem.

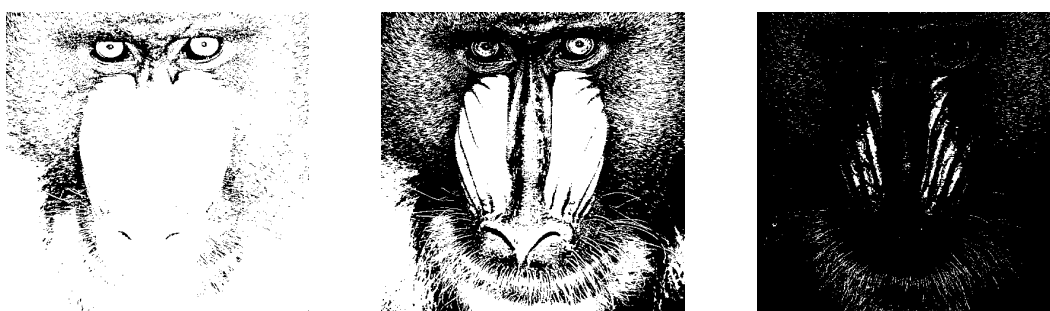


Figura 3: Limiarização global.

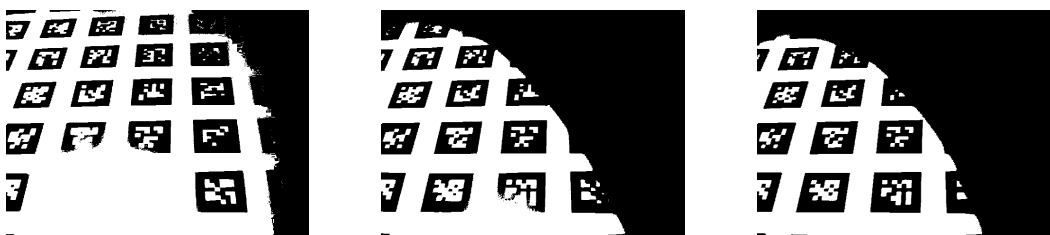


Figura 4: Limiarização global.

Na imagem 5 são comparadas as técnicas de limiarização global e local para uma imagem com diferente iluminação em pontos distintos. Com essa comparação é possível observar a limitação da técnica de limiarização global, uma vez que as bordas não são detectadas em todas as regiões. Já a técnica de limiarização global atingiu melhor resultado, delimitando as regiões de bordas da imagem.

Para não tornar este relatório extenso, as demais imagens de saída e os histogramas das imagens resultantes estão no Apêndice e no arquivo kleberkruger.zip (diretório "out"),

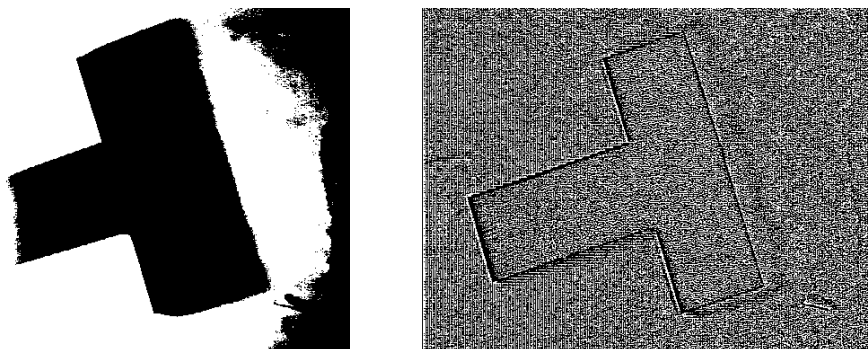


Figura 5: Limiarização global versus limiarização adaptativa local

enviado via Google Classroom. As técnicas de limiarização local mostraram diferenças e os resultados podem ser conferidos nas Figuras 6 a 9.

4.3 Tempo de Execução

O tempo de execução das funções de limiarização global foram incomparavelmente mais rápidas, conforme pode ser visto no log de execução (`/out/execution_time.txt`). A execução da técnica de limiarização global duraram menos de 10ms, enquanto é possível verificar alguns tempos da limiarização local acima de 10s, e alguns casos mais extremos superior a 20s. Estes valores são compreensivos, uma vez que a limiarização global foi implementada de forma vetorizada.

Apêndice



Figura 6: Método de limiarização global com os valores 105 e 128, respectivamente.

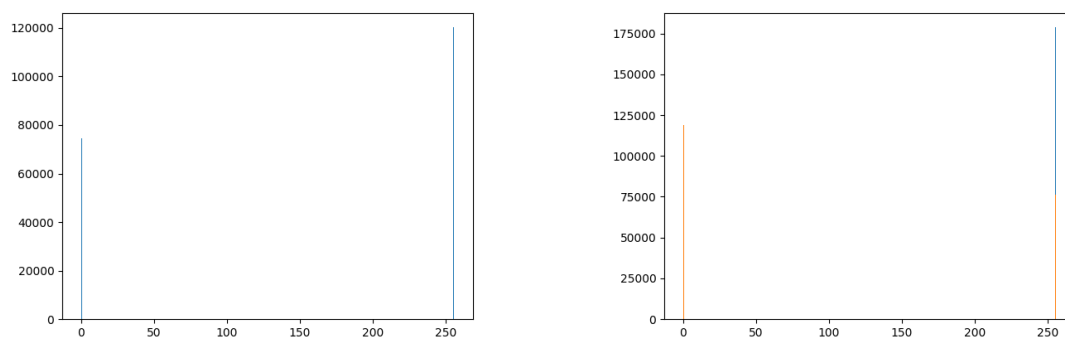


Figura 7: Histograma dos métodos de limiarização global com os valores 105 e 128.

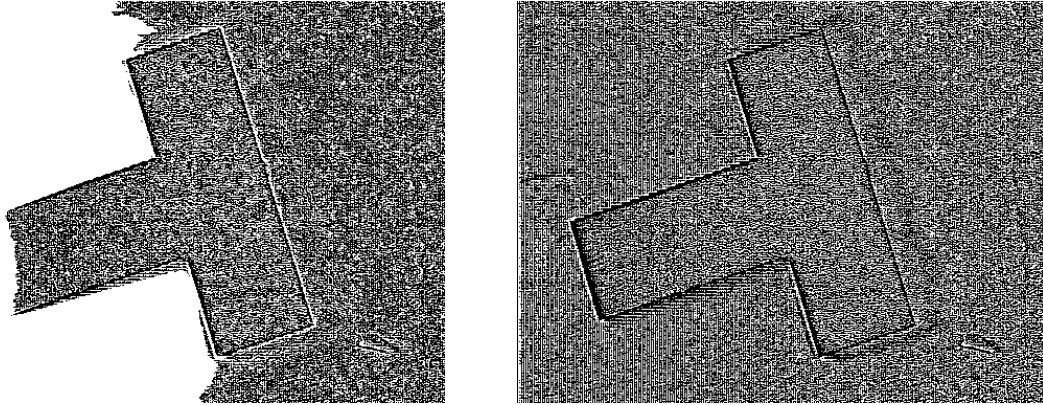


Figura 8: Método de Bernsen e método de Niblack.

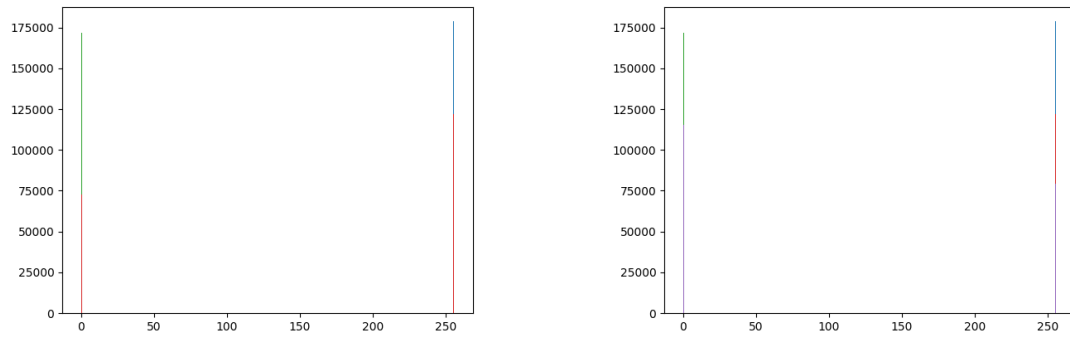


Figura 9: Histogramas dos método de Bernsen e de Niblack.

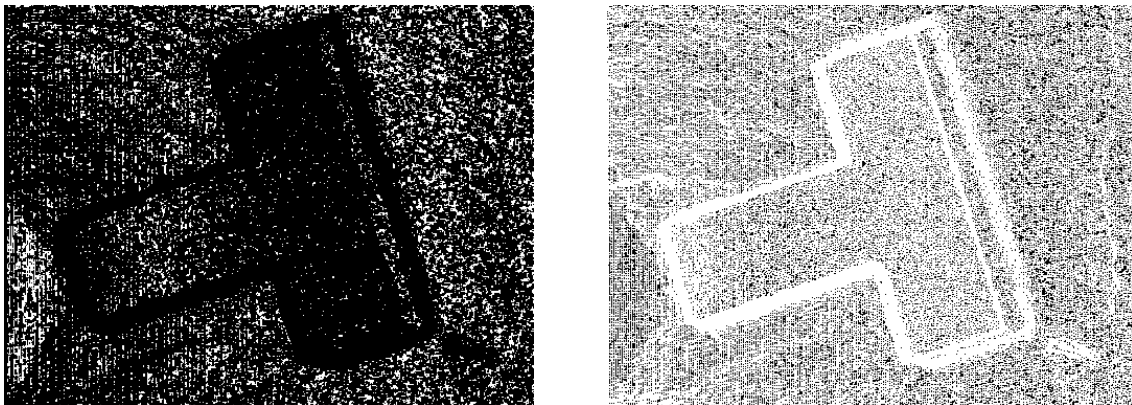


Figura 10: Método de Phansalskar More e Sabale, e método de contraste.

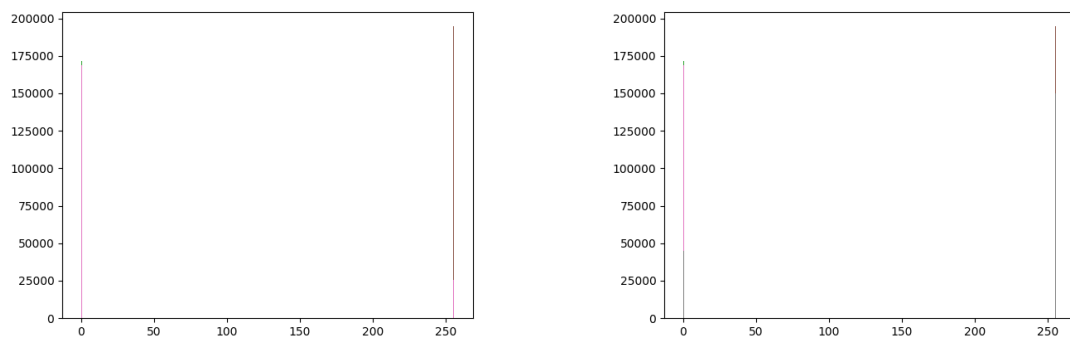


Figura 11: Histograma dos métodos de Phansalskar More e Sabale e do método de contraste.

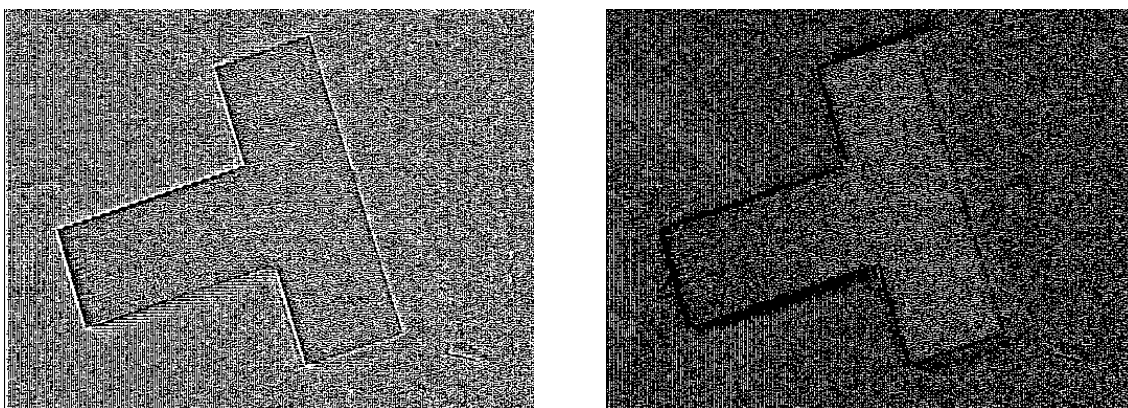


Figura 12: Método da média e método da mediana.

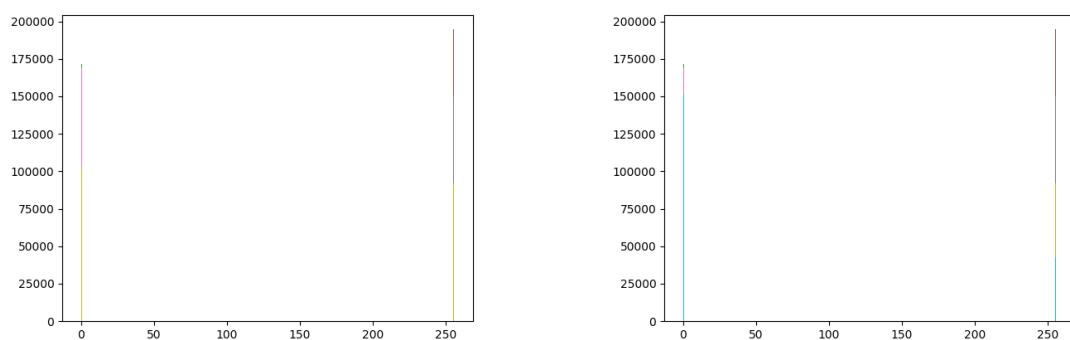


Figura 13: Histograma dos métodos da média e do método da mediana.