

# Relatório – Projeto 2

1<sup>st</sup> Júlia Yuri Garcia Baba  
Departamento de Ciência da  
Computação  
UnB  
Brasília, Brasil  
190057921@aluno.unb.br

**Keywords**—*Erosão, dilatação, abertura, fechamento, top-hat, hit-or-miss, máscara, morfologia, segmentação, watershed, binarização.*

## I RESUMO

Este trabalho teve como objetivo principal aplicar os conhecimentos obtidos em sala de aula acerca de manipulações morfológicas de imagens para a obtenção de elementos desejáveis, assim como a aplicação do processo de segmentação. Todos os códigos foram desenvolvidos em OpenCv 4.5.1, usando a linguagem de programação Python.

O projeto é dividido em três partes: A primeira consiste em testar a capacidade de retirar o ruído de uma imagem, para que consigamos binarizá-la, afim de obter uma melhor visualização dos dados. Já na segunda parte, foi avaliada a capacidade de, a partir de algoritmos morfológicos de imagens binárias e em níveis de cinza, fazer a retirada de determinados objetos da imagem.

Por fim, na terceira parte do trabalho foi avaliada a capacidade de particionar uma determinada imagem em diferentes regiões, através de aplicação segmentação de watershed.

## II INTRODUÇÃO

### A. Primeira questão: Binarização da imagem

Neste exercício foi dada uma figura, com números escritos em um fundo cinza com uma marca d'água. O objetivo principal dessa etapa é apresentar uma imagem binária em que os números são destacados do fundo branco.

Para que isso fosse possível, usou-se um algoritmo morfológico em níveis de cinza, chamado top-hat, de forma a reduzir as variações no fundo, diminuindo os ruídos e descontinuações para que a binarização de Otsu, que será aprofundada na próxima seção, seja o mais eficiente possível.

### B. Segunda questão: Retirada de objetos de uma imagem

Nessa etapa, recebe-se uma imagem com duas cookies, uma mordida e outra inteira. A finalidade nessa segunda parte é retirar a cookie mordida, deixando apenas a cookie inteira na figura.

Para isso, deve-se, escolhendo um limiar apropriado, binarizar a imagem e através de algoritmos morfológicos em imagem binária, eliminar a cookie mordida da imagem, deixando pelo menos parte da cookie inteira.

Após tais operações, deve-se recuperar a imagem da cookie completa e usar essa como máscara, afim de obter uma imagem final em níveis de cinza com somente a cookie completa

### C. Terceira questão: Segmentação Watershed

Por fim, neste terceiro exercício, temos uma figura com várias células em níveis de cinza. Objetivo principal dessa

questão é obtermos como saída uma imagem segmentada, ou seja, particionada em diferentes regiões.

Inicialmente, usando métodos similares a primeira questão, deve-se binarizar essa imagem, obtendo uma figura onde as células são pretas e o fundo é branco, feito isso, usou-se um algoritmo afim de preencher o buracos deixados na binarização e outro para preencher os espaços desconectados e finalmente foi computado a segmentação watershed.

## III METODOLOGIA

Dado o resumo do objetivo de cada exercício, será feita uma explicação breve de como tais finalidades foram alcançadas. Quais funções foram implementadas para que os algoritmos funcionassem como esperado.

### A. Primeira questão

#### a) Binarização de Otsu

A binarização é o método mais simples de segmentação de imagens, que resumidamente, consiste em separar uma imagem em regiões de interesse e não interesse através de um limiar. Essas regiões podem ser representadas por pixels pretos e brancos.

Como neste problema estamos interessados na imagem binarizada do resultado. Nossas operações vão alterar constantemente os valores de intensidade da imagem, por isso foi escolhido um método de binarização conhecido como binarização de Otsu.

Esse método tem por objetivo, a partir de uma imagem em tons de cinza, determinar o valor ideal de um *threshold* que separe os elementos do fundo e da frente, a partir da distribuição do seu histograma bimodal. A ideia é aproximar o histograma de uma imagem por duas funções Gaussianas e escolher um limiar de forma a minimizar a variância entre as classes fundo e objeto. De forma simplificada, esse algoritmo “pesa” qual dos lados é mais pesado, e remove o peso do lado mais pesado até que este se torne o mais leve, repetindo essa mesma operação até que os dois extremos da balança se encontrem.

Essa técnica foi primeiro aplicada diretamente a imagem, usando uma função pré definida do OpenCv (*cv.threshold*), para fins de comparação posterior.

#### b) Top-hat

Como estamos lidando com uma imagem com números num fundo claro, com uma marca d'água. Dessa forma, deve-se aplicar um pré-processamento, afim de minimizar as variações de fundo, para que a binarização seja o mais eficiente possível, assim foi escolhido o método de *top-hat*, já que este é comumente usado para extrair pequenos detalhes da imagem..

Essa técnica consiste basicamente na diferença entre a imagem de entrada e a operação morfológica de abertura da

imagem, que pode ser descrita como:

$$A \circ B = (A \ominus B) \oplus B \quad (1)$$

Essa operação, em uma imagem binária pode ser definida como a união de todas as translações de B que se encaixam em A. Geralmente, tende a suavizar contornos e aliminar saliências finas. Em níveis de cinza, a abertura vai diminuir a intensidade dos objetos claros da imagem, pouco afetando a os objetos escuros.

Sendo assim, essa operação combinada com a diferença da imagem original se encarrega de reduzir consideravelmente a marca d'água, aumentando a eficiência da binarização de Otsu e atingindo o resultado esperado, no qual a letras são evidenciadas em um fundo branco.

É importante salientar que para que o *top-hat* possa ser aplicado, a imagem deve ser negatizada antes da transformação e desnegatizada após a binarização, já que o objetivo principal da questão é realçar os apenas os dígitos. Além de que todas as operações foram feitas usando funções pré definidas do OpenCv e da biblioteca numpy

### B. Segunda questão

#### a) Transformada hit-or-miss

Neste problema temos uma foto de dois biscoitos, um inteiro e outro mordido, o objetivo é isolar a cooky inteira utilizando uma máscara plana. Para isso, a imagem foi binarizada escolhendo-se experimentalmente um limiar apropriado, assim pode-se obter as formas gerais dos cookies.

Dessa forma, devemos eliminar os elementos indesejados da imagem, nesse caso os ruídos e a cooky mordida. Para os ruídos, foram feitas uma operação de abertura (equação 1) e outra de fechamento, descrita como:

$$A \bullet B = (A \oplus B) \ominus B \quad (2)$$

Que em uma imagem binária pode ser definida como o conjunto formado quando B tenta contornar A. Geralmente funde descontinuidades, eliminada pequenos buracos e preenche lacunas em contornos. Em níveis de cinza, o fechamento vai aumentar a intensidade dos objetos escuros da imagem. A combinação da abertura com um fechamento em níveis de cinza é uma suavização da imagem pois remove os picos escuros e claros.

Com a eliminação das imperfeições do fundo, resta agora a remoção da cooky mordida, para isso foi feita uma transformada hit-or-miss. Contudo, antes deve-se obter uma máscara em forma do objeto que será mantido na imagem, obtemos esse elemento recortando a área da imagem binarizada que corresponde a cooky inteira. Obtendo-se a máscara, realiza-se a transformada hir-or-miss, que através de operações básica, pode ser descrita como:

$$A \circledast X = (A \ominus X) - (A \oplus (W - X)) \quad (3)$$

Essa operação é útil para para localizar padrões em imagens binárias, já que ela pega uma forma X e encontra na imagem A aonde ela ocorre, assim obtemos os pontos no qual o objeto aparece. Sendo assim, fazemos uma dilatação para obter a máscara final que irá isolar a cooky fazendo uma simples operação AND na imagem original, recuperando a figura da cooky inteira.

### C. Terceira questão

#### a) Segmentação watershed

Nesta questão temos como entrada uma imagem de diversas células, o nosso objetivo principal é segmentar a imagem, ou seja, separar as diferentes células, através da segmentação watershed.

Para isso, primeiro devemos binarizar diretamente a imagem utilizando o mesmo método da primeira questão, a binarização de Otsu. Logo em seguida, foi usado um algoritmo para preencher os espaços vazios deixados na imagem após a binarização, esse método consiste em, resumidamente, inverter a imagem, encontrar seus contornos por meio de uma função pré definida, preencher esses com preto e por fim inverter a imagem novamente, esse processo poderia ser comparado com a função *imfill()* do matlab.

Agora precisamos remover todos os pequenos ruídos brancos da imagem, para isso podemos usar a abertura morfológica (equação 1). Contudo, apesar das modificações anteriores, nota-se que a imagem ainda possui uma área desconhecida, impedindo que a segmentação watershed seja aplicada de forma eficiente. Portanto, precisamos extrair uma área que temos certeza que são células, dessa forma, primeiramente aplicou a dilatação, que pode ser descrita como:

$$A \oplus B = \{z | (B \cap z) \cap A \neq \emptyset\} \quad (4)$$

Que em uma imagem binária, pode ser definida como o conjunto de todos os deslocamentos de z de forma que B e A se sobrepõem em pelos menos um elemento. Em níveis de cinza com um elemento estruturante plano, o pixel sendo avaliado é substituído pelo maior valor coberto por seu elemento estruturante, fazendo com que o haja um aumento no limite do objeto para o fundo, desse modo, podemos ter certeza de que qualquer região no fundo do resultado é realmente um fundo, uma vez que a região limite é removida.

No entanto, como as células estão se tocando, calculou-se uma função de distância, aplicando-se um limite adequado e usando funções pré-definidas. Nesse operador as intensidades do nível de cinza dos pontos dentro das regiões de primeiro plano são alteradas para distanciar suas respectivas distâncias do valor zero mais próximo. Sendo assim, para obtermos a área desconhecida subtraímos as imagens desses dois passos, já que, respectivamente, é aonde o primeiro plano e o fundo se encontram.

Portanto, como sabemos quais são as regiões das moedas e qual o fundo, podemos começar a criar o marcador, por meio de uma função pré-definida, e rotular as regiões dentro dele, sendo este um array de mesmo tamanho da imagem, porém com outro tipo de dado. As regiões conhecidas são rotuladas com quaisquer inteiros positivos e diferentes entre si, já a área desconhecida é marcada com zero. Contudo, deve-se fazer uma última modificação no marcador, já que que se o fundo estiver marcado com zero, o watershed classificará como área desconhecida, logo, esse será marcado com um número inteiro.

Por fim, aplicou-se a segmentação de watershed e modificou-se a imagem do marcador, sendo a região limite marcada por -1, obtendo-se a imagem final.

## IV RESULTADOS

Após uma noção geral de cada função dos exercícios e como foi implementada cada uma, será dada nessa seção a saída gerada por cada um dos programas com imagens pré-definidas

## A. Primeira questão

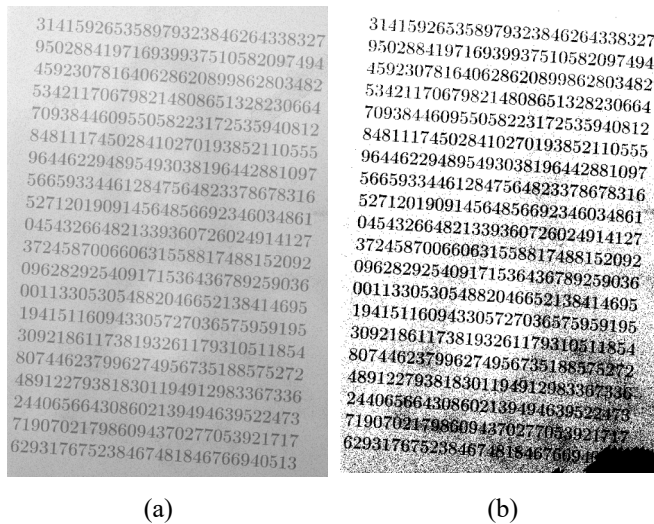


Fig. 1. (a) Imagem original; (b) Imagem binarizada diretamente com seu threshold baseado no seu histograma, conhecido como binarização de otsu.

Primeiramente, como ponto de referência para os resultados fizemos uma binarização sem nenhum tipo de tratamento anterior, afim de mostrar a efetividade da binarização de Otsu.

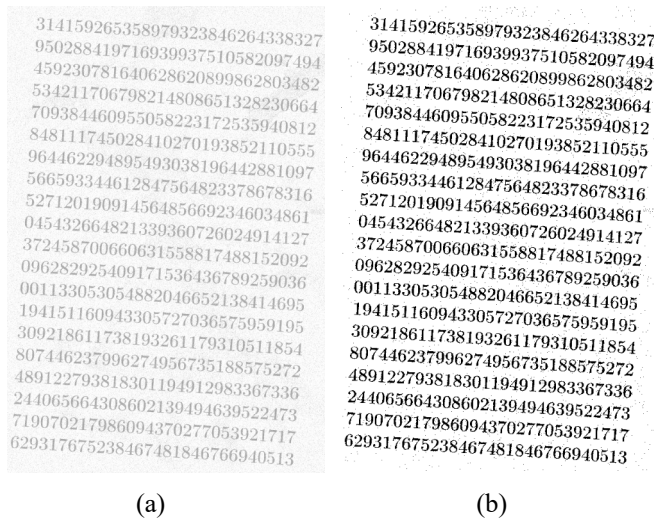


Fig. 2. (a) Imagem com o top-hat de máscara quadrada 7x7, aplicado antes da binarização; (b) Imagem 2.(a) após a binarização.

Agora, aplicamos um filtro de tratamento feito pela transformação morfológica top-hat e em seguida novamente a binarização de otsu. Nota-se que em comparação com a imagem 1.(b) a binarização se mostrou significativamente mais eficiente na última etapa, deixando os números nítidos, removendo completamente a marca d'água e sem perda de dados.

## B. Segunda questão

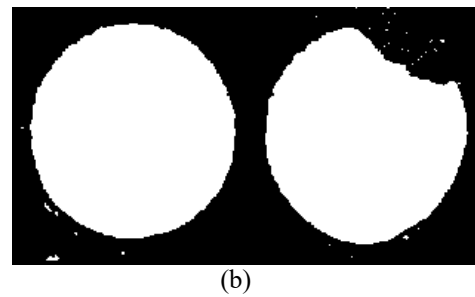
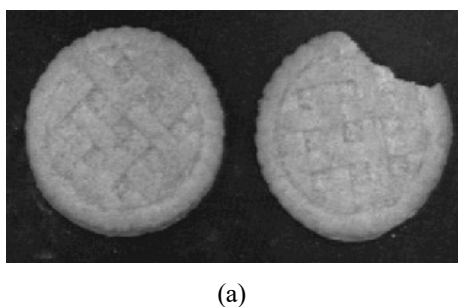


Fig. 3. (a) Imagem original; (b) Imagem binarizada com um threshold ajustado para 55 de intensidade.

Fazendo diversos testes, o melhor resultado da binarização que mantém a forma do biscoito com poucas interferências foi com um limiar de 55 de intensidade

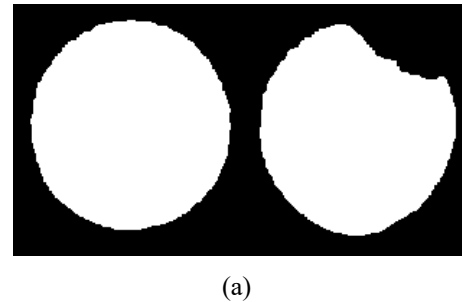


Fig. 4. (a) Imagem após a realização das operações de abertura e fechamento com uma máscara quadrada 3x3

Contudo, a binarização deixa certas imperfeições na imagem, esses pequenos elementos foram removidos fazendo uma operação de abertura e fechamento com uma máscara quadrada 3x3. Sendo assim, obteve-se uma imagem mais “limpa” com menos interferências que poderiam atrapalhar as etapas seguintes.

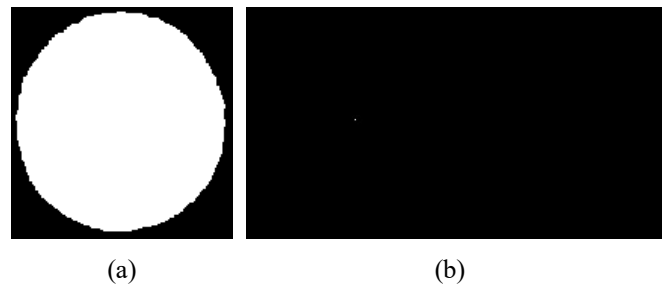
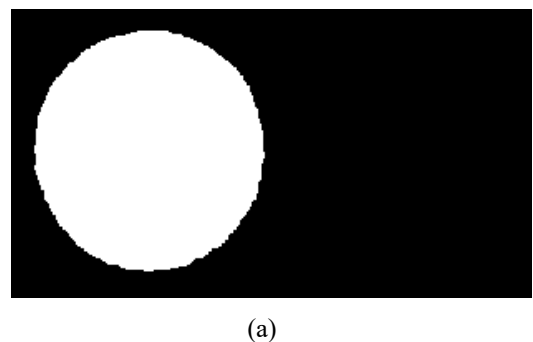
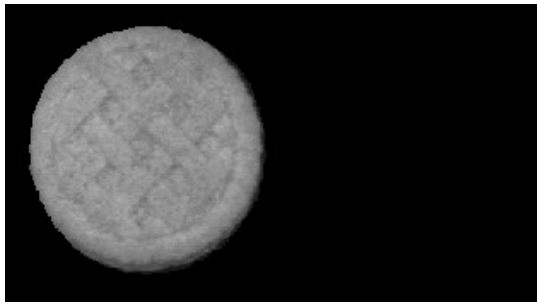


Fig. 5. (a) Máscara utilizada na transformação hit-or-miss; (b) Resultado da transformada hit-or-miss

Com a imagem binarizada pronta, obtemos uma seção com o formato do biscoito inteiro e usamos a transformada hit-or-miss, que gera um único pixel, indicando o local do biscoito.



(a)

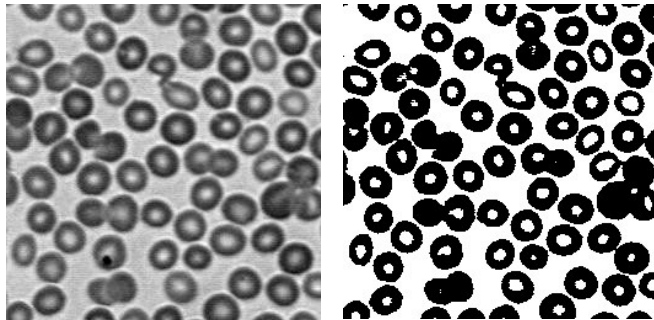


(b)

Fig. 6. (a) Máscara obtida após a dilatação; (b) Imagem com apenas a cookie inteira, obtida fazendo um AND entre a imagem original e a máscara.

Aplicando a dilatação obtemos a máscara final e a utilizamos para isolar a cookie através de uma operação AND com a imagem original.

### C. Terceira questão

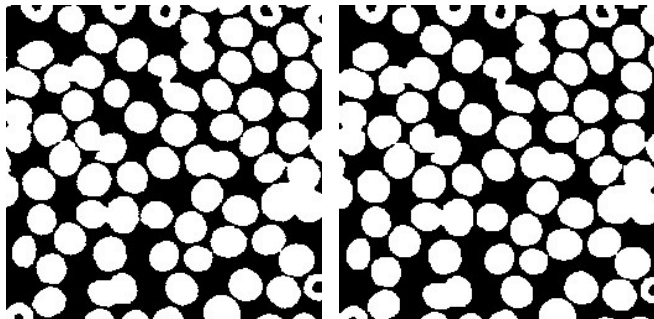


(a)

(b)

Fig. 7. (a) Imagem original; (b) Imagem binarizada diretamente com seu threshold baseado no seu histograma, conhecido como binarização de otsu.

Similar a questão 1, foi aplicado a binarização de Otsu sem nenhum tipo de tratamento anterior.



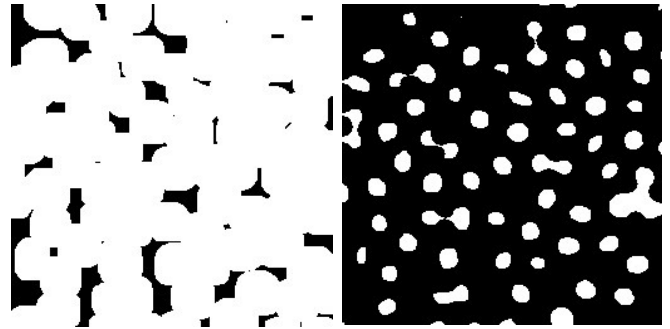
(a)

(b)

Fig. 8. (a) Imagem após o preenchimento de buracos; (b) Imagem após a realização das operações de abertura com uma máscara quadrada 4x4.

Como pode ser notado na figura 7.(b) a binarização deixa diversas falhas, sendo assim foi preciso preencher os buracos e tratar as imperfeições, usando um função de preencher espaços desconectados e uma operação morfológica de abertura de máscara quadrada 4x4, respectivamente.

É importante salientar que o algoritmo usado para preencher espaços não se mostrou totalmente eficiente, deixando algumas células com buracos, o que acarretou para que o resultado final do processo tivesse certas imperfeições.

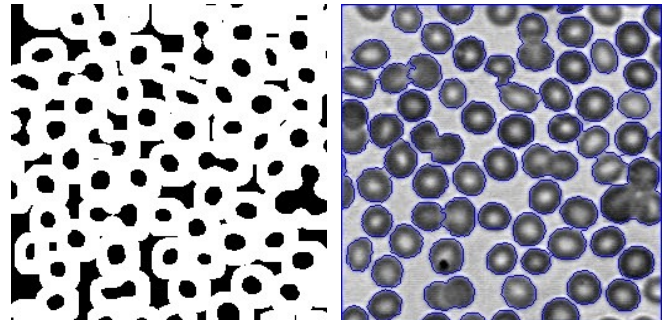


(a)

(b)

Fig. 9. (a) Resultado da dilatação. (b) Resultado da aplicação da função de distância.

Com o objetivo de identificar as área desconhecida, foi aplicada uma dilatação com máscara 4x4 e uma função de distância com limiar 5.



(a)

(b)

Fig. 10. (a) Imagem que representa a área desconhecida; (b) Imagem final, resultante da segmentação watershed

Por fim, obteve-se a imagem da área desconhecida por meio da subtração das figuras 9.(a) e 9.(b) e aplicou-se a ela a segmentação watershed chegando ao resultado final esperado.

### CONCLUSÃO

Por fim, pode-se concluir, no geral, que de fato a morfologia matemática pode nos ajudar a resolver muitos problemas, com soluções relativamente simples para soluções que são aparentemente complexas. Entretanto, na extração do texto pode-se ver uma certa limitação das operações, já que a binarização direta levou a uma perda de dados da imagem e não trouxe o resultado esperado, o que pode ser consertado posteriormente com pré-tratamento da imagem, mas que ainda poderia ser melhorado se pudessemos utilizar técnicas de de realce no domínio espacial.

Já na aplicação hit-or-miss, conseguimos demonstrar o uso das operações morfológicas para isolar elementos de uma imagem e nos fornecer um pequeno ensaio de como a geometria das figuras pode ser aplicada para discretizar e selecionar regiões de interesse. Porém, percebe-se um aspecto negativo, como as soluções são ,de certa forma, muito específicas e dependem diretamente do dado de entrada, caso fosse solicitado a retirada de outro elemento, que não especificado anteriormente, o programa não cumpriria com seu objetivo.

Desta forma, a execução da segmentação watershed se mostrou eficiente se feitas a devidas operações morfológicas na imagens antes da aplicação da mesma, conseguindo ultrapassar as limitações do algoritmo de preencher espaços, gerando a saída esperada.

