

# Documentação Trabalho 2 - IPI

Bruna Azambuja  
Departamento de Ciência da  
Computação  
UnB  
Brasília, Brasil  
bubu.azambuja@hotmail.com

**Keywords - processamento, imagens.**

## I. RESUMO

Este trabalho teve como objetivo principal aplicar os conhecimentos obtidos em sala de aula sobre processamento e características gerais de uma imagem; tais como algoritmos de morfologia matemática, filtros no domínio da frequência e manipulação de imagens. Os códigos foram desenvolvidos em MatLab 2018, e testados com imagens distintas, como será explicado melhor na seção de Resultados.

O exercício consiste em três partes: A primeira testa a capacidade de retirar ruído de uma imagem, para que consigamos binarizá-la para ter uma melhor visualização dos dados. Já a segunda parte do trabalho tem objetivo de retirar o Padrão Moiré de uma imagem, um processo de filtragem rejeita-Notch, mediante a utilização de filtros passa-altas Butterworth.

Por fim, na terceira parte do trabalho 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.

## II. INTRODUÇÃO

### A. Primeiro Exercício: Binarização

Neste exercício foi dada uma fotografia da página de um livro, com números escritos num fundo cinza com marca d'água.

A saída gerada pelo programa deve ser uma imagem binária em que os números são destacados do fundo branco. Para isso deve-se usar algoritmos morfológicos para que tanto o fundo quanto as letras fiquem livres de possíveis ruídos e descontinuações e a binarização seja o mais eficiente possível.

### B. Segundo Exercício: Filtro Rejeita-Notch Butterworth

Já na segunda parte do trabalho, tem-se uma imagem contendo padrões Moiré, e deve-se utilizar de filtros passa-altas Butterworth com  $n$  igual a 4, para que o processo rejeita-notch seja realizado.

Para isso deve-se transformar a imagem para o domínio da frequência utilizando a transformada rápida de Fourier e eliminando, mediante filtros Butterworth, as frequências específicas que causam a distorção.

### C. Terceiro Exercício: Retirada de Objetos de uma Imagem

Finalmente, neste terceiro exercício, recebe-se uma imagem com dois cookies em níveis de cinza, um mordido e o outro inteiro. Nosso trabalho nessa última parte é retirar o cookie mordido da imagem, deixando apenas o cookie inteiro em níveis de cinza.

Para isso, deve-se primeiramente binarizar a imagem, em seguida, mediante algoritmos morfológicos, eliminar por completo o cookie mordido, deixando pelo menos parte do cookie inteiro.

Após recuperado o formato do cookie inteiro, deve-se, a partir da imagem binária com apenas um cookie e a imagem original, obter a imagem de saída em níveis de cinza com apenas um cookie.

## III. METODOLOGIA

Dado o resumo do objetivo de cada exercício, será feita a explicação sucinta de como foi alcançado tal objetivo. Quais funções foram usadas e quais algoritmos foram implementados para que o programa funcionasse do jeito esperado.

Lembrando aqui que os nomes de funções utilizadas são referentes ao MatLab 2018.

### A. Primeiro Exercício:

Foi recebida uma imagem no formato .png de uma página de livro com números num fundo claro com marca d'água. Como pré-processamento, aplicou-se um filtro Gaussiano para que as letras fiquem mais suaves.

A partir da imagem já pré-processada, utilizou-se morfologia matemática para a criação de uma imagem que seja só 'fundo', ou seja, sem as letras. Para isso, foi feito um fechamento (dilatação + erosão) a partir da função *imclose*, com um elemento estruturante disco de raio 5.

Essa imagem só 'fundo' foi subtraída da imagem original, obtendo uma imagem apenas com os números, tal processo seria o equivalente a utilizar a função *imbothat*, que aplica a transformada Bottom Hat na imagem – realça objetos escuros em fundo claro (que é o caso dos números escuros em página clara).

Então, aplicou-se a binarização da imagem, a partir da função *imbinarize* utilizando um limiar calculado pela função *graythresh* que nos fornece o melhor limiar de binarização para aquela imagem específica.

Após a binarização, foi feito um pós-processamento da imagem: um fechamento com elemento estruturante quadrado de lado 2. Então, finalmente, foi feito o complemento da imagem, apenas para que possa se ter melhor visualização.

### B. Segundo Exercício

Nesta questão nos foi dada uma imagem que contém Padrões Moiré, como já dito nosso objetivo era retirar esses padrões por meio de filtros Butterworth.

Para isso, primeiramente, utilizou-se de padding para que a filtragem seja exata – processo em que a imagem em subplotada numa matriz maior que o tamanho dela, para que as bordas fiquem bem filtradas. Depois, utilizou-se da função *fft* e *fftshift* para gerar a transformada de Fourier da imagem, já transladada para o centro.

Foi aplicada também a função logarítmica para que a imagem esclareça e possamos ver melhor o que se passa nas partes escuras da transformada de Fourier.

Após o tratamento com a imagem, deve-se criar a máscara que será utilizada para a filtragem; para isso, utiliza-se da fórmula do filtro de Butterworth:

$$D_k(i,j) = [(i - u_k)^2 + (j - v_k)^2]^{1/2} \quad (1)$$

$$H(i,j) = [1/(1 + D_{ok}/D_k(i,j)^{2n})] \quad (2)$$

onde  $(u_k, v_k)$  são as coordenadas do centro do filtro-altas Butterworth,  $D_{ok}$  é o raio do filtro e 'n' é o número já especificado pelo exercício (4 - número de pares de frequências que queremos retirar).

Após criado o formato de um filtro rejeita-Notch Butterworth, basta posicioná-lo nas frequências que queremos tirar – ou seja, as frequências que causam o ruído da imagem. Para isso, foi criada uma máscara do tamanho da imagem (usando padding), e subplotar o filtro butterworth nas posições específicas dessas frequências.

A coordenada de cada uma das frequências que queremos tirar foi adquirida utilizando a função *impixelinfo* do matlab, que nos dá a coordenada de determinados pixels.

Após criada a máscara, basta aplicá-la na imagem da transformada de Fourier, desse modo, estaremos retirando as frequências que causam as distorções na imagem.

Feita a filtragem, basta aplicar a transformada inversa de Fourier utilizando as funções *ifft* e *ifftshift* que obtemos a imagem no domínio do tempo sem os padrões Moiré.

### C. Terceiro Exercício

Finalmente, neste terceiro exercício, como já foi explicado, recebemos uma imagem em níveis de cinza de dois cookies, um mordido e outro inteiro, e nosso trabalho é retirar o cookie mordido da imagem mediante algoritmos morfológicos.

Primeiramente, binariza-se a imagem utilizando a função *imbinarize* e *graythresh* para encontrar o limiar exato. Após a binarização, aplicou-se abertura da imagem com elemento estruturante disco de raio 60, ou seja, suficientemente pequeno para caber dentro do cookie inteiro, e suficientemente grande para não caber no cookie mordido.

Dessa forma, a imagem resultante que obtemos é uma imagem com o formato incompleto do cookie inteiro, porém sem nada do cookie mordido. Depois, aplicando dilatação por um quadrado de lado 20 (grande o suficiente para preencher o cookie inteiro sem alcançar o mordido), dessa forma, teremos uma figura maior que o cookie inteiro, então basta multiplicá-la pela imagem binária dos dois cookies que

recupera-se o cookie inteiro no seu formato original, porém sem o mordido.

Finalmente, temos uma máscara com apenas o cookie inteiro; então basta multiplicá-la com a imagem original que obteremos apenas o cookie inteiro em níveis de cinza.

## IV. RESULTADOS

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

### A. Primeiro Exercício:

A imagem de entrada em nível de teste foi:

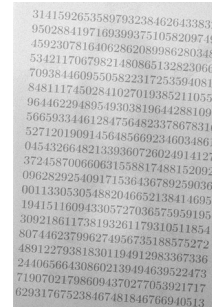


Fig.1. Imagem de entrada

Feita a aplicação do filtro Gaussiano e um fechamento com um disco de raio 5, obtêve-se uma imagem só 'fundo':



Fig.2. Imagem só fundo – note que podemos ver a marca d'água

Subtraindo o fundo da imagem original, obteve-se a imagem com apenas os números destacados (objetos escuros num fundo claro – Transformada Bottom-Hat):



Fig.3. Transformada Bottom-Hat

Finalmente, utilizando a função de binarização *imbinarize*, binarizamos a imagem e fazemos um pós-processamento, aplicando um fechamento com quadrado de lado 2, para que as letras fiquem mais contínuas, com menos ruído. Foi feito também o complemento da imagem, para que obtivéssemos letras pretas em fundo branco:

314159265358979323846264338327  
 950288419716939937510582097494  
 459230781640628620899862803482  
 534211706798214808651328230664  
 709384460955058223172535940812  
 848111745028410270193852110555  
 964462294895493038196442881097  
 566593344612847564823378678316  
 527120190914564856692346034861  
 045432664821339360726024914127  
 372458700660631558817488152092  
 096282925409171536436789259036  
 001133053054882046652138414695  
 194151160943305727036575959195  
 309218611738193261179310511854  
 80744623796274956735188575272  
 489122793818301194912983367336  
 244065664308602139494639522473  
 719070217986094370277053921717  
 629317675238467481846766940513

Fig.4. Imagem de Saída - binarização

### B. Segundo Exercício

A imagem de entrada em nível de teste foi:



Fig.5. Imagem de Entrada

Após feita a transformada de Fourier e o shift e utilizando log para que possamos enxergar:

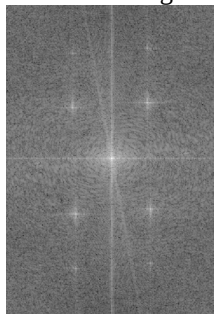


Fig.6. Transformada de Fourier da Imagem de Entrada

Calculado o filtro passa-altas de Butterworth, e posicionando-o nas frequências específicas, obteve-se:



Fig.7. Filtro Butterworth



Fig.8. Máscara Final

Aplicado o filtro e fazendo a transformada inversa de Fourier obteve-se:



Fig.9. Imagem de entrada ----- Imagem de saída

### C. Terceiro Exercício

A imagem de entrada em nível de teste foi:



Fig.10. Imagem de Entrada

Após feita a binarização, foi feita uma abertura com um disco de raio 60, obtendo uma parte do cookie inteiro e nada do cookie mordido:



Fig.11. Imagem Binária com parte do cookie inteiro

Fez-se uma dilatação com um quadrado de lado 20:

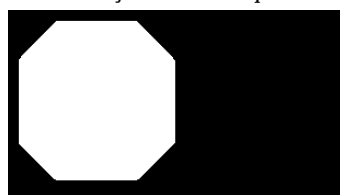


Fig.12. Cookie dilatado

Depois, para recuperar o resto do cookie inteiro, fizemos uma multiplicação dessa máscara com a imagem binária original, resultando numa máscara perfeita do cookie inteiro:



Fig.13. Máscara cookie

Para recuperarmos os tons de cinza, basta multiplicar essa máscara pela imagem original:

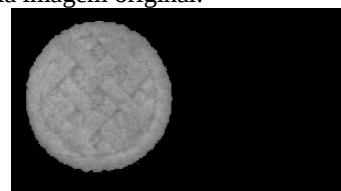


Fig.14. Imagem de Saída

### CONCLUSÃO

Finalmente, concluímos por esse trabalho, que de fato Morfologia Matemática pode nos ajudar a resolver muitos problemas do dia-a-dia. Obtém-se soluções relativamente simples para questões que aparentemente são complexas, como tirar um objeto de uma imagem ou tirar o ruído dela.

Porém, percebe-se também, que apresenta um aspecto negativo, pois as soluções são muito específicas e dependem diretamente do dado de entrada.

Por exemplo, caso seja solicitado a retirada de outro objeto na imagem, que não o definido anteriormente, o algoritmo não funcionará mais! Ou até mesmo retirar os padrões Moiré de outra imagem, o filtro utilizado não será o mesmo, e etc.

Por isso, pôde-se concluir que Morfologia Matemática é muito útil e interessante, porém bastante específica e dependente dos dados de entrada.

#### REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Slides do professor Bruno Macchiavello.
- [2] Site de documentação do MatLab – MathWorks.