

# MO443 - Processamento de Imagem Digital

## Trabalho 03

Patrick de Carvalho Tavares Rezende Ferreira - 175480

27 de maio de 2020

### 1 Introdução

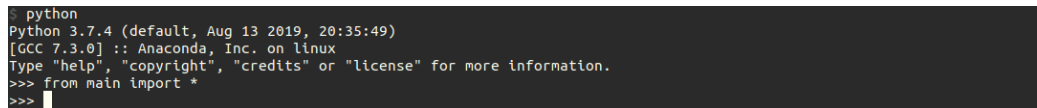
O objetivo deste trabalho é aplicar a transformada rápida de Fourier (do inglês, Fast Fourier Transform - FFT) em imagens digitais, convertendo-as para o domínio de frequência. A filtragem das imagens no domínio de frequência possibilita a alteração de seus valores originais em novas informações, de forma a atenuar o ruído nas imagens, suavizar os dados, aumentar o contraste, realçar detalhes (bordas) das imagens, e operações afins.

Serão aplicados filtro passa-baixa, passa-alta e passa-faixa através do espectro de Fourier em imagens monocromáticas. Também será realizada a compressão destas no domínio da frequência.

### 2 Descrição do código elaborado

O código desenvolvido neste relatório é contido no arquivo “main.py” e foi desenvolvido para que o utilizador possa simplesmente importar todas as suas funções e utilizá-las independentemente. Para fins de demonstração, o código também possui um script dentro da função “main()” que é executado sempre que chamado como arquivo principal pelo interpretador python e cria um diretório “output” contendo as imagens solicitadas neste roteiro. Todas as funções com argumentos numéricos aceitam que estes sejam inteiros positivos, tratando de forma adequada e convertendo para o tipo “uint8” a matriz da imagem antes de salvar o arquivo de saída. O diretório das imagens de entrada é passado também como argumento de função.

As funções criadas para a execução deste roteiro são descritas nas subseções a seguir e podem ser importadas conforme a figura 1.



```
python
Python 3.7.4 (default, Aug 13 2019, 20:35:49)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from main import *
>>>
```

Figura 1: Comando para importar as funções utilizadas para o roteiro.

## 2.1 Função circle-into-matrix

A função “circle-into-matrix” inscreve um círculo preenchido de elementos valendo “1” em uma matriz de elementos valendo “0”, de acordo com seus argumentos “r” - a representar o raio do círculo - e “N”, indicando que desejamos uma matriz  $N \times N$ . Esta função é particularmente útil para determinar os círculos de efeito dos filtros aplicados. Descrição na tabela 1.

Tabela 1: Referência da função circle-into-matrix.

Parâmetro	Descrição
r	Raio do círculo a ser definido nos elementos da matriz
N	Dimensões da matriz quadrado contendo o círculo (default: 512).
retorno	Retorna a matriz de zeros contendo o círculo de elementos “1”.

## 2.2 Função get-spectrum-from-img

Como o nome já indica, esta função é encarregada de receber uma matriz de uma imagem monocromática “uint8” e extrair seu espectro através da transformada de Fourier em 2d. Sua implementação é baseada na função “fft2” do pacote *numpy*, devido à sua otimização, e a componente de frequência-zero é transladada para o centro do espectro através da função “fftshift”, também do *numpy*. Descrição na tabela 2.

Tabela 2: Referência da função get-spectrum-from-img.

Parâmetro	Descrição
img	Matriz de imagem em “uint8”
retorno	Retorna a matriz do espectro de frequência em 2d (os valores são complexos).

## 2.3 Função get-img-from-spectrum

Desempenha o reverso da função “get-spectrum-from-img”, recebendo uma matriz com o espectro do Fourier de uma imagem original, e retornando a imagem reconstruída. Também é suportada pelas funções do pacote *numpy* como “ifftshift” (que desfaz a centralização na componente de frequência zero) e “ifft2”, que desempenha a transformada de Fourier reversa em 2d. Descrição na tabela 3.

Tabela 3: Referência da função get-img-from-spectrum.

Parâmetro	Descrição
img	Matriz de imagem em “uint8”
retorno	Retorna a matriz do espectro de frequência em 2d (os valores são complexos).

## 2.4 Função `apply-fourier-filter`

Esta função realiza a aplicação do filtro no domínio da frequência de Fourier. Recebe uma matriz de espectro de frequência e um raio mínimo e máximo que determinam os valores onde serão preservadas as frequências.


Para fazer um filtro passa-baixas, o raio mínimo deve valer zero e o raio máximo deve ser maior que ele. Para um filtro passa-faixa, escolha um raio máximo maior que o raio mínimo, o qual não deve valer zero. Para um filtro passa-alta, use o raio máximo sendo zero e o raio mínimo sendo o valor de corte. Para um filtro rejeita-faixa, use quaisquer valores diferentes de zero para os raios, desde que o mínimo seja maior que o máximo. Descrição na tabela 4.

Tabela 4: Referência da função `apply-fourier-filter`.

Parâmetro	Descrição
<code>img</code>	Matriz de imagem em “uint8”
retorno	Retorna a matriz do espectro de frequência em 2d (os valores são complexos).

## 3 Execução do script demonstrativo e tarefas do roteiro

A geração das saídas solicitadas por este roteiro pode ser realizada disponibilizando-se a imagem “baboon.png” no mesmo diretório que o script “main.py” e utilizando-se do comando exibido na figura 2. As imagens são geradas dentro do diretório “output”.



```
python main.py
```

Figura 2: Comando para gerar as saídas requisitadas pelo roteiro.

### 3.1 Obtenção do espectro de frequências

Neste roteiro serão trabalhadas técnicas baseadas na transformada de Fourier em 2 dimensões, caso particularmente útil para a edição de imagens. A transformada de Fourier é uma operação que se baseia em identificar a correlação do sinal[2] considerado com uma faixa de frequências possíveis, identificando de fato a presença de cada uma daquelas que são chamadas “componentes de frequência” na função considerada.

O roteiro solicita inicialmente que façamos a extração do espectro de frequências presentes em uma imagem monocromática através da transformada rápida de Fourier e façamos sua reconstrução. Este processo é exibido na figura 3, onde o espectro é representado apenas em magnitude, já que seus valores são dados complexos e seria difícil de representá-lo em duas dimensões. Nota-se que as frequências mais baixas possuem maior influência nesta imagem, e que a reconstrução é extremamente fidedigna, já que as possíveis perdas se devem apenas a imprecisões numéricas e esta imagem não possui um grande range de valores possível para cada pixel (“uint8”).

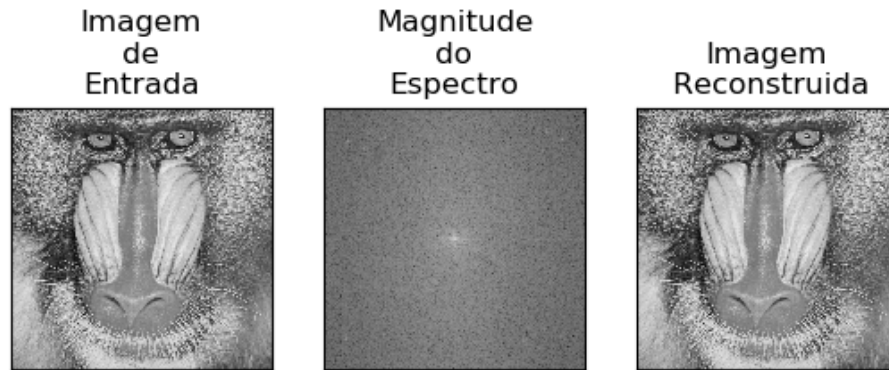


Figura 3: Obtenção do espectro em frequência e reconstrução da imagem.

### 3.2 Filtragem

A primeira manipulação solicitada pelo roteiro na aparência da imagem é realizar a filtragem de frequências altas (através do filtro passa-baixa), mantendo somente as baixas. Para tal, as frequências são radialmente selecionadas em função de sua distância ao centro do espectro (deixando passar apenas as de baixo valor). O núcleo deste filtro é exibido na figura 4.

A imagem resultante da reconstrução do espectro filtrado pelo passa-baixa é exibida na figura 5 e mostra que a imagem parece ter perdido nitidez, exibindo majoritariamente as cores predominantes em cada região da imagem. Isto já era previsto, já que as baixas frequências são as componentes responsáveis por compôr as variações mais sutis em qualquer função.

A segunda manipulação solicitada pelo roteiro é a aplicação do filtro passa-alta em 2 dimensões. As frequências do espectro original são, novamente, selecionadas radialmente, porém se utilizando agora apenas as que estão acima de um certo raio a partir do centro (a frequência de corte selecionada pelo usuário do código). O núcleo deste filtro é exibido na figura 4.

A imagem reconstruída com esta porção do espectro é exibida na figura 5 e revela um interessante efeito de ressaltar apenas os pontos onde há mudança “brusca” de tonalidade da imagem. Isto é

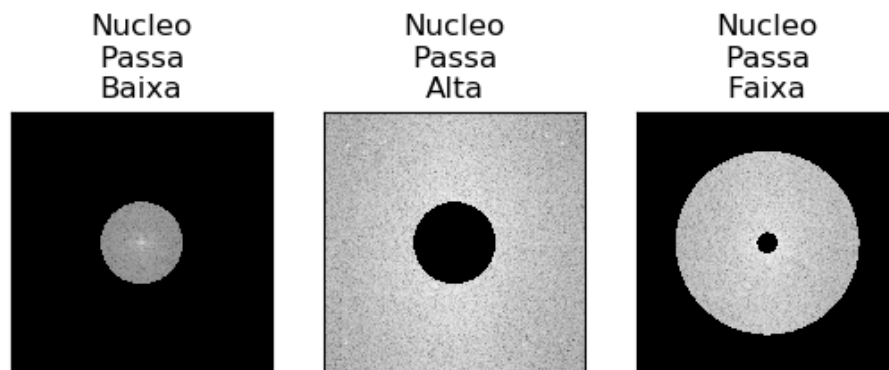


Figura 4: Nucleo de cada filtro utilizado neste roteiro.

previsto, já que as frequências mais elevadas são as componentes necessárias para compôr variações rápidas de qualquer função, seja de 1, 2 ou N dimensões.

Como último elemento de filtragem, implementamos também um filtro passa-faixa, que, como o nome já indica, permite a passagem de frequências apenas dentro de uma faixa especificada. O núcleo do espectro selecionado por este filtro é exibido na figura 4 e a reconstrução desse espectro é exibida na figura 5.

A imagem passa a ter um pouco mais da predominância de tonalidades por região do que o caso do filtro passa-alta, assim como possui mais detalhes do que o filtro passa-baixa. Mas uma importante observação a se notar é que a coloração média da imagem não é recuperada, visto que somente contemplando-se a frequência zero da imagem original isto seria possível, o que só ocorre no caso de estarmos trabalhando com o próprio espectro original ou com uma reconstrução de um passa-baixas.

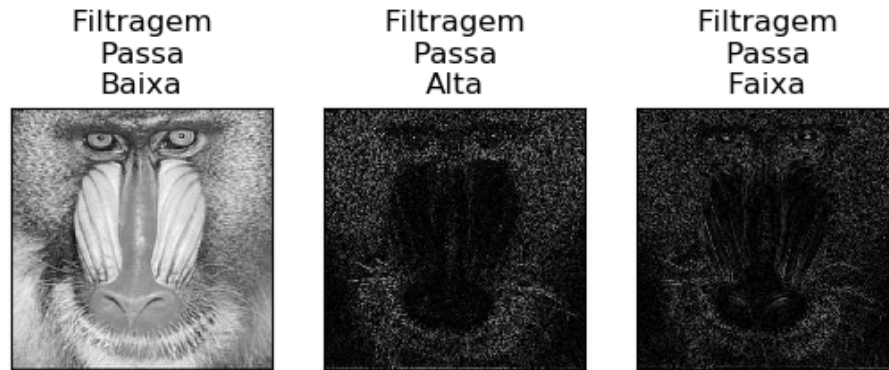


Figura 5: Núcleo de cada filtro utilizado neste roteiro.

### 3.3 Compressão de imagens

A compressão neste caso se dá pela retirada de componentes do espectro cuja magnitude seja menor que um certo limiar. Este limiar foi definido como sendo 70% da componente de maior magnitude e não leva em conta a frequência das componentes. Como a formação de detalhes e da imagem como um todo depende, não somente da frequência e magnitude de cada componente, como também do alinhamento de fases entre elas, o efeito da compressão em alguns pontos da imagem é de perda de detalhes e, em outros, de perda da intensidade daquela região. O efeito no geral é de perda de qualidade na representação, como pode-se notar na figura 6

Caso o limiar de compressão seja elevado para retirar todas as componentes cuja magnitude seja inferior a 80% do maior valor, a imagem reconstruída resultante é mostrada na figura 7. O interessante deste caso é que restam tão poucas componentes formando esta imagem que, principalmente na horizontal, é possível ver um certo caráter periódico na própria imagem, e as manchas da face do babuíno estão um pouco desalinhadas na parte inferior, notavelmente pela falta de componentes que, juntas, iriam representar corretamente a fotografia original.

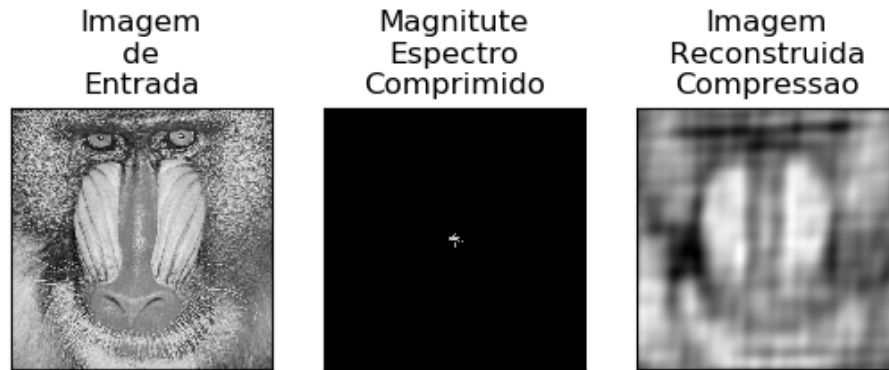


Figura 6: Magnitude do espectro comprimido a 70% e reconstrução.

## 4 Conclusão

A transformada de Fourier é uma importante ferramenta para diversas áreas da engenharia e da ciência em geral. Nos casos abordados, executamos a transformada em 2 dimensões sobre imagens para identificar as frequências presentes nestas e poder realizar algumas operações.

Trabalhar sobre o espectro obtido pela transformada de Fourier se mostrou como uma poderosa ferramenta para a filtragem de frequências e, conseqüentemente, características das imagens que podem então ser trabalhadas de maneira intuitiva e até visual, como apresentado ao longo do roteiro. A compressão de imagens a partir de suas componentes de frequência se mostrou efetiva e também de fácil aplicação, embora existam outras técnicas de mais fácil aplicação para o âmbito específico de compressão, tal como a redução de matrizes decompostas em valores singulares[1].

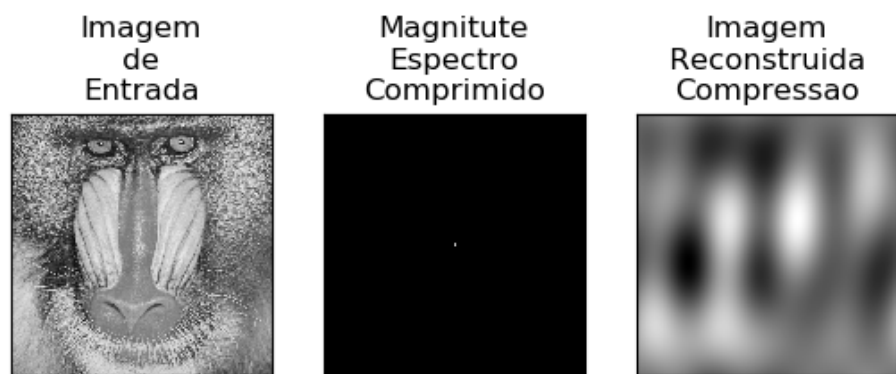


Figura 7: Magnitude do espectro comprimido a 80% e reconstrução.



## Referências

- [1] Wikipédia. *Decomposição em valores singulares* — *Wikipédia, a enciclopédia livre*. [Online; accessed 31-março-2020]. 2020. URL: [https://pt.wikipedia.org/w/index.php?title=Decomposi%C3%A7%C3%A3o\\_em\\_valores\\_singulares&oldid=57926065](https://pt.wikipedia.org/w/index.php?title=Decomposi%C3%A7%C3%A3o_em_valores_singulares&oldid=57926065).
- [2] Wikipédia. *Transformada de Fourier* — *Wikipédia, a enciclopédia livre*. [Online; accessed 24-maio-2020]. 2020. URL: [https://pt.wikipedia.org/w/index.php?title=Transformada\\_de\\_Fourier&oldid=58339526](https://pt.wikipedia.org/w/index.php?title=Transformada_de_Fourier&oldid=58339526).