Filtragem no Domínio de Frequência

Prof. Jefferson T. Oliva
jeffersonoliva@utfpr.edu.br

Processamento de Imagens
Engenharia de Computação
Departamento Acadêmico de Informática (Dainf)
Universidade Tecnológica Federal do Paraná (UTFPR)
Campus Pato Branco

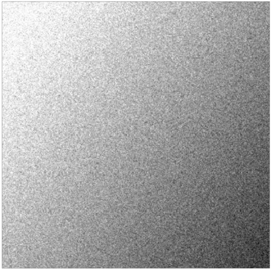
Sumário

- Transformada de Fourier
- Transformada de Fourier 2D
- Filtragem no Domínio da Frequência

Introdução

- Filtragem espacial
  - Podemos aplicar filtros permitindo obter realce sobre certas características da imagem, enquanto outras características são suprimidas ou eliminadas
  - O filtro de média suaviza a imagem removendo detalhes de transição e variações, porém realçando as cores relativas às regiões planas
  - o filtro diferencial extrai apenas as bordas/transições, removendo informações das regiões planas (*e.g.* removendo a cor dos objetos)
  - Essas diferentes características da imagem são descritas pelo comportamento da transição das intensidade
  - Uma região de imagem cujos pixeis vizinhos possuem transição rápida entre intensidades distintas pode indicar: uma borda de um objeto, ruído, ou detalhes de um determinado objeto

- Exemplo: cada pixel é bastante diferente um do outro



[[159, 119, 110, 154, 144, 134, 147, 121, 157, 135], [170, 171, 151, 154, 152, 141, 141, 126, 131, 160], [150, 163, 133, 147, 153, 130, 133, 148, 158, 145], [153, 136, 128, 134, 114, 141, 136, 151, 136, 128], [139, 137, 115, 130, 136, 145, 137, 118, 171, 151], [166, 130, 120, 197, 146, 151, 130, 146, 132, 191], [128, 137, 130, 109, 130, 167, 162, 151, 120, 153], [156, 140, 134, 120, 147, 154, 142, 120, 135, 130], [136, 141, 141, 106, 122, 141, 128, 120, 170, 108], [144, 157, 178, 129, 131, 146, 130, 135, 136, 140]]

Introdução

- Exemplo: uma mesma região possui valores constantes

img[200 : 210, 200 : 210]

[[127, 127, 127, 127, 127, 127, 127, 127, 127, 127], [127, 127, 127, 127, 127, 127, 127, 127, 127, 127],

[127, 127, 127, 127, 127, 127, 127, 127, 127, 127], [127, 127, 127, 127, 127, 127, 127, 127, 127, 127],

[127, 127, 127, 127, 127, 127, 127, 127, 127, 127], [127, 127, 127, 127, 127, 127, 127, 127, 127, 127],

[127, 127, 127, 127, 127, 127, 127, 127, 127, 127], [127, 127, 127, 127, 127, 127, 127, 127, 127, 127],

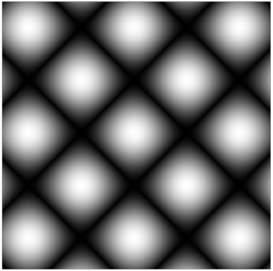
[127, 127, 127, 127, 127, 127, 127, 127, 127, 127], [127, 127, 127, 127, 127, 127, 127, 127, 127, 127]]

- Exemplo: outra região contém mais transições

img[120:130, 120:130]

[[178, 179, 179, 179, 180, 181, 181, 181, 182, 182], [178, 179, 179, 180, 180, 181, 181, 182, 182, 183],
[179, 180, 180, 181, 181, 182, 182, 182, 183, 183], [180, 180, 181, 182, 182, 182, 183, 184, 183, 184],
[180, 181, 182, 182, 183, 183, 184, 184, 184, 185], [181, 182, 183, 182, 183, 184, 184, 184, 185, 186],
[182, 183, 183, 183, 184, 184, 185, 186, 186, 186], [182, 183, 183, 184, 185, 185, 185, 186, 186, 187],
[183, 183, 185, 185, 185, 185, 186, 187, 187, 188], [184, 185, 185, 185, 186, 186, 186, 188, 187, 188]]

- Exemplo: nesse caso há variação, porém essa é suave, lenta, entre um pixel e seus vizinhos



img[120:130, 120:130]

[[149, 152, 155, 158, 161, 164, 166, 169, 172, 174], [152, 155, 158, 161, 164, 167, 170, 172, 175, 177], [155, 158, 161, 164, 167, 170, 173, 175, 178, 181], [158, 161, 164, 167, 170, 173, 176, 179, 181, 184], [161, 164, 167, 170, 173, 176, 179, 181, 184, 187], [164, 167, 170, 173, 176, 179, 182, 184, 187, 189], [166, 170, 173, 176, 179, 182, 184, 187, 190, 192], [169, 172, 175, 179, 181, 184, 187, 190, 192, 195], [172, 175, 178, 181, 184, 187, 190, 192, 195, 198], [174, 177, 181, 184, 187, 189, 192, 195, 198, 200]]

Introdução

- Uma forma de ver as características descritas nos exemplos anteriores é na forma de frequências

- Para a análise dos componentes de frequência presentes em um sinal (ou imagem), a Transformada de Fourier é a abordagem mais utilizada

Sumário

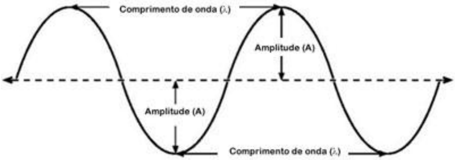
Transformada de Fourier

Transformada de Fourier

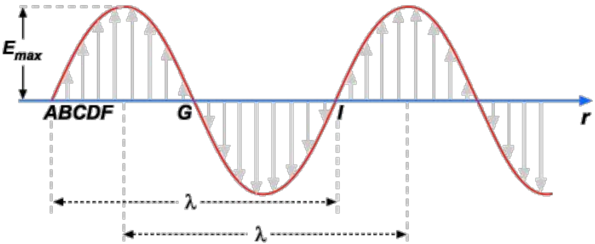
- Jean-Baptiste Joseph
  - Contribuição para ferramentas de análise de frequência
  - Estudava transferência de calor e afirmou que uma função de uma variável poderia ser expandida em uma série de senóides de múltiplos da variável

Transformada de Fourier

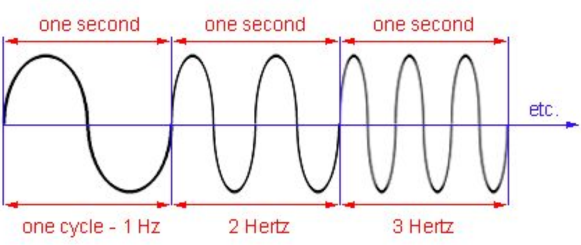
- Série de Fourier: qualquer função periódica pode ser expressa como a soma de senos e/ou cossenos de diferentes frequências, cada uma multiplicada por um coeficiente diferente
- Transformações matemáticas são aplicadas a sinais para obter informações não disponíveis (ou não visíveis) diretamente no sinal original
- Um sinal unidimensional está geralmente no domínio do tempo em sua forma original
  - Quando exibimos o sinal temos uma representação tempo-amplitude



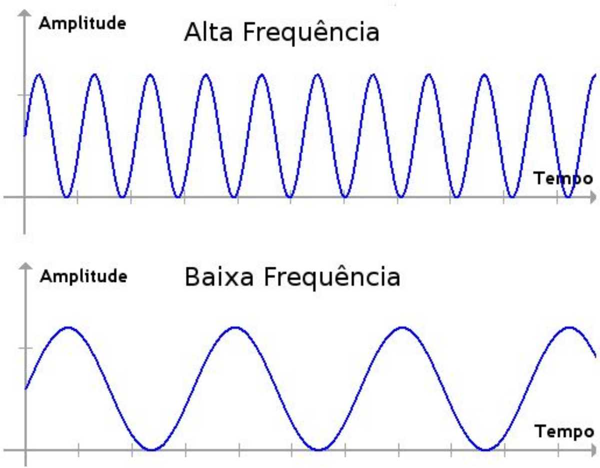
- Definições básicas
  - Comprimento de onda é a distância entre valores repetidos sucessivos em um padrão de onda



- Definições básicas
  - Frequência indica o número de ocorrências de um evento em um determinado intervalo de tempo
    - É medida em ciclos por segundo (Hertz)
    - É inversamente proporcional ao comprimento de onda

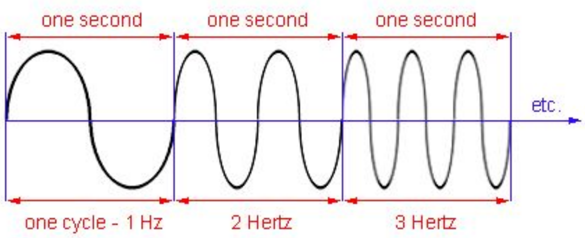


- Definições básicas

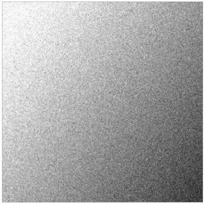


- Definições básicas
  - Período: intervalo de tempo necessário para uma onda percorrer o seu comprimento
    - Inverso da frequência

$$T = \frac{1}{f}$$



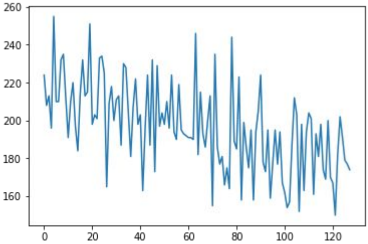
- Considere um fragmento de uma imagem ruidosa



img[100, 64:192]

[224, 208, 213, 196, 255, 210, 210, 232, 235, 212, 191, 210, 220, 197, 184, 214, 232, 213, 215, 251, 198, 203, 201, 233, 234, 225, 165, 209, 218, 200, 211, 213, 187, 230, 228, 205, 181, 207, 222, 198, 203, 163, 196, 224, 187, 232, 173, 229, 197, 204, 198, 210, 196, 224, 194, 190, 219, 195, 193, 192, 191, 191, 190, 246, 182, 215, 193, 186, 200, 213, 155, 235, 186, 177, 181, 166, 175, 164, 244, 189, 185, 223, 158, 199, 186, 175, 195, 158, 194, 205, 224, 178, 173, 195, 159, 177, 195, 177, 194, 167, 162, 154, 157, 187, 212, 203, 152, 198, 163, 194, 204, 201, 161, 193, 181, 198, 174, 169, 200, 170, 167, 150, 182, 202, 191, 179, 177, 174]

- Ao plotar o vetor do slide anterior, note que há uma transição/modificação brusca entre os valores de um valor e seu vizinho
  - Para estudar esses diferentes padrões de transição, podemos pensar em termos das funções matemáticas mais simples possível e que permitam representar variação: seno e cosseno



Transformada de Fourier

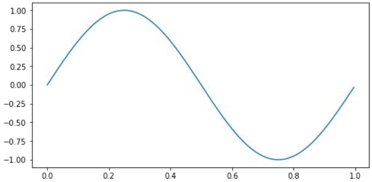
- Função senoidal
  - Uma função senoidal com baixa frequência tem variação suave, enquanto uma função com alta frequência, variação brusca
    - Considere uma função seno avaliada no intervalo entre 0 e 2
  - Como queremos avaliar as características de frequência (em termos de ciclos por segundo - Hz) usamos 2π no argumento e então a variável t abaixo passa a significar o tempo em segundos

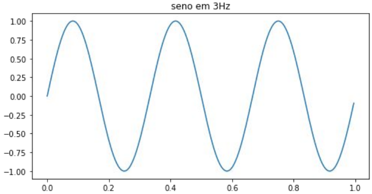
$$np.sin(t * (2 * pi))$$

- Função senoidal

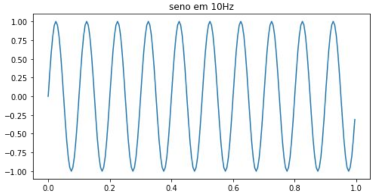
```
#Exemplo 1 Hz
t = np.arange(0, 1, 0.005)
pi = 3.1415
fsin1 = np.sin(t * (2*pi))
```

```
Fs = 3 # Parametro de Frequencia
fsin3 = np.sin(t * (2*pi) * Fs)
```



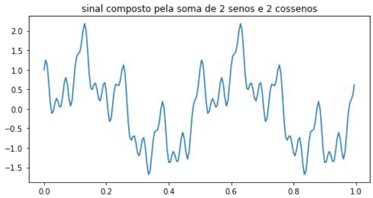


- Função senoidal
  - Exemplo para $Fs = 10$



  - Na análise de Fourier é calculada a a contribuição das diferentes frequências (em Hz) para a formação de um sinal
    - O sinal pode ser escrito pela soma de senos e cossenos de diferentes frequências

- Podemos obter sinais bastante complexos apenas pela combinação linear de senos e cossenos em diferentes frequências
    - Exemplo: sinal = 1 * np.sin(t * (2 * pi) * 2) + 0.6 * np.cos(t * (2 * pi) * 8) + 0.4 * np.cos(t*(2 * pi) * 16) + 0.3*np.sin(t*(2*pi) * 32)



- Para verificar qual tipo de função compõe o sinal podemos multiplicar ponto a ponto e somar o resultado
  - Por exemplo, vamos multiplicar o sinal 1 com senos e cossenos nas frequências 1, 2, 3 e 4, e ver o resultado

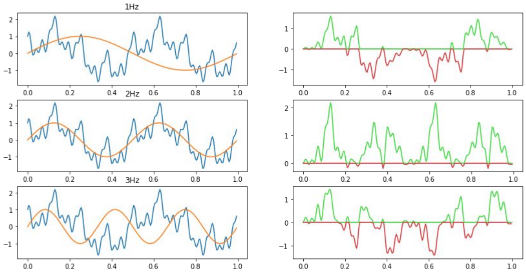
  - Ao fazer a soma dos produtos estamos tentando encontrar o quão parecidos os sinais são

```
for Freq in range(1,5):
    SomaProdSeno = np.sum( sinal * np.sin(t * 2 * pi * Freq))
    SomaProdCoss = np.sum( sinal * np.cos(t * 2 * pi * Freq))
    print("Frequencia %d" % Freq)
    print("\tSeno = %.5f" % SomaProdSeno)
    print("\tCosseno = %.5f" % SomaProdCoss)
```

```
Frequencia 1
        Seno = 0.00008
        Cosseno = -0.00475
Frequencia 2
        Seno = 100.00311
        Cosseno = -0.00475
Frequencia 3
        Seno = 0.00024
        Cosseno = -0.00474
Frequencia 4
        Seno = 0.00033
        Cosseno = -0.00474
```

Transformada de Fourier

- Note nos gráficos abaixo que a correspondência entre o sinal e a frequência 1 ocorre em alguns pontos (produz valores positivos), se cancela em outros, e gera valores negativos



- Ao somar, temos um valor baixo pois os negativos cancelam os positivos
- No lado direito, em verde, mostramos os valores maiores que zero e, em vermelho, os valores menores do que zero

- Testando com frequências de 1 a 32

```
print("Freq.\tSeno\tCosseno")

for Freq in range(1,33):
    SomaProdSeno = np.sum(sinal * np.sin(t * 2 * pi * Freq))
    SomaProdCoss = np.sum(sinal * np.cos(t * 2 * pi * Freq))
    print("%d\t%.3f\t%.3f" % (Freq, SomaProdSeno, SomaProdCoss))
```

```
Freq.   Seno    Cosseno
1       0.000   -0.005
2       100.003 -0.005
3       0.000   -0.005
4       0.000   -0.005
5       0.000   -0.005
6       0.000   -0.005
7       0.001   -0.005
8       0.001   59.997
9       0.001   -0.005
10      0.001   -0.005
11      0.001   -0.005
12      0.001   -0.005
13      0.001   -0.005
14      0.001   -0.005
15      0.001   -0.005
16      0.001   39.997
17      0.001   -0.005
18      0.001   -0.005
19      0.002   -0.005
20      0.002   -0.005
21      0.002   -0.005
22      0.002   -0.005
23      0.002   -0.004
24      0.002   -0.004
25      0.002   -0.004
26      0.002   -0.004
27      0.002   -0.004
28      0.002   -0.004
29      0.002   -0.004
30      0.002   -0.004
31      0.003   -0.004
32      30.003  -0.004
```

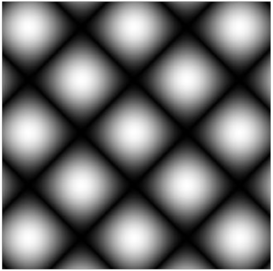
- Implementação de uma função que retorna a frequência relacionada à maior amplitude de um sinal

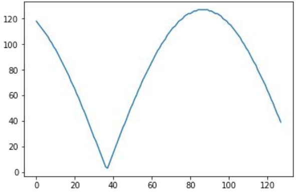
```
def maximum_frequencies(A):
    # vamos computar frequencias ate metade do tamanho do sinal
    n = int(A.shape[0] / 2)
    t = np.linspace(0, 1, n * 2)
    magnit = np.empty(n)

    for f in range(2, n):
        SomaProdSeno = np.sum(A * np.sin(t * 2 * pi * f))
        SomaProdCoss = np.sum(A * np.cos(t * 2 * pi * f))
        magnit[f] = SomaProdSeno + SomaProdCoss

    return np.argmax(magnit)
```

- Exemplo

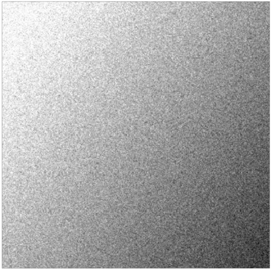


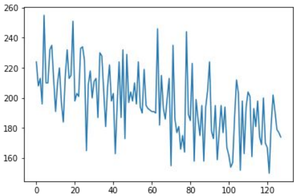


vet1 = img[100, 64:192]

[118, 116, 114, 112, 110, 108, 106, 103, 101, 98, 96, 93, 90, 87, 84, 81, 78, 75, 71, 68, 65, 61, 58, 54, 50, 47, 43, 39, 35, 31, 27, 24, 20, 16, 12, 8, 4, 3, 7, 11, 15, 19, 23, 27, 31, 35, 38, 42, 46, 50, 53, 57, 60, 64, 67, 71, 74, 77, 80, 83, 86, 89, 92, 95, 97, 100, 102, 104, 107, 109, 111, 113, 114, 116, 118, 119, 120, 122, 123, 124, 124, 125, 126, 126, 127, 127, 127, 127, 127, 127, 126, 126, 125, 124, 124, 123, 122, 120, 119, 118, 116, 115, 113, 111, 109, 107, 105, 102, 100, 97, 95, 92, 89, 86, 84, 80, 77, 74, 71, 68, 64, 61, 57, 54, 50, 46, 43, 39]

- Exemplo





vet2 = img[100, 64:192]

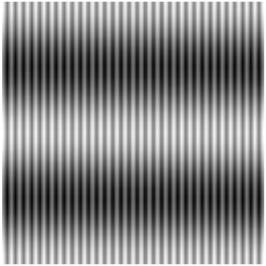
[224, 208, 213, 196, 255, 210, 210, 232, 235, 212, 191, 210, 220, 197, 184, 214, 232, 213, 215, 251, 198, 203, 201, 233, 234, 225, 165, 209, 218, 200, 211, 213, 187, 230, 228, 205, 181, 207, 222, 198, 203, 163, 196, 224, 187, 232, 173, 229, 197, 204, 198, 210, 196, 224, 194, 190, 219, 195, 193, 192, 191, 191, 190, 246, 182, 215, 193, 186, 200, 213, 155, 235, 186, 177, 181, 166, 175, 164, 244, 189, 185, 223, 158, 199, 186, 175, 195, 158, 194, 205, 224, 178, 173, 195, 159, 177, 195, 177, 194, 167, 162, 154, 157, 187, 212, 203, 152, 198, 163, 194, 204, 201, 161, 193, 181, 198, 174, 169, 200, 170, 167, 150, 182, 202, 191, 179, 177, 174]

Transformada de Fourier

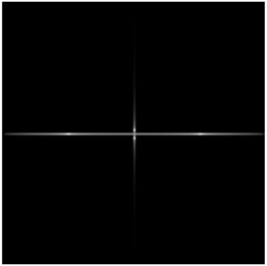
- A Transformada de Fourier muda (qualquer sinal ou dado) do domínio do tempo ou espaço para o domínio da frequência
  - Em sinais domínio do tempo: $f(t)$
  - Em imagens domínio do espaço: $f(x, y)$
- A Transformada Inversa de Fourier converte o domínio da frequência de volta ao domínio do tempo ou espaço
- A teoria de Fourier diz que qualquer sinal, em nosso caso as imagens, podem ser expressadas como uma soma de senoides
- No caso das imagens, são variações senoides do brilho na imagem

Transformada de Fourier

- A transformada de Fourier em imagens tem um único componente
  - Representado por 2 valores pontos brilhantes simetricamente localizados em relação à parte central da imagem da transformada
- O centro da imagem é a origem do sistema de coordenadas da frequência
- A transformada de Fourier pode ser vista como um "prisma matemático" que separa uma função em vários componentes de frequência



cos(32*x) cos(2*y)



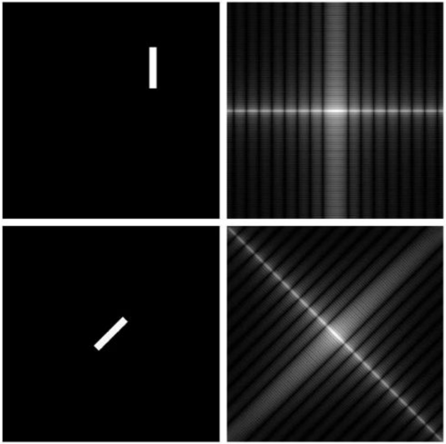
Transformada de Fourier

- A transformada de Fourier sempre trata a imagem como se fosse parte de um vetor replicado periodicamente de imagens idênticas estendendo-os vertical e horizontalmente ao infinito

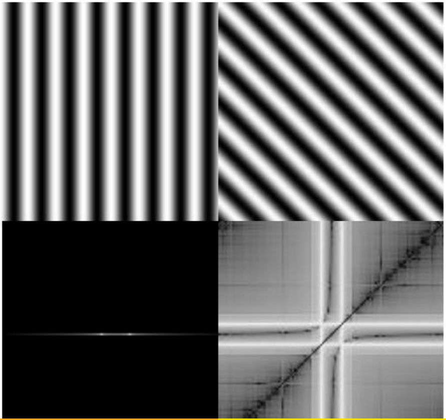
- Visualização do espectro de Fourier
  - No centro do espectro estão concentradas as frequências mais baixas, que compõem regiões da imagem uniformes ou com mudanças suaves de intensidade/cor
  - As frequências mais altas estão afastadas do centro, sendo referentes a regiões da imagem com mudanças abruptas de intensidade/cor

Transformada de Fourier

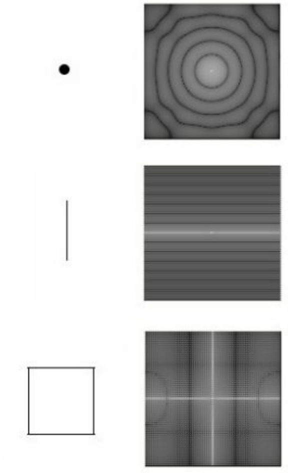
- A rotação de uma imagem resulta também na rotação da correspondente transformada de Fourier



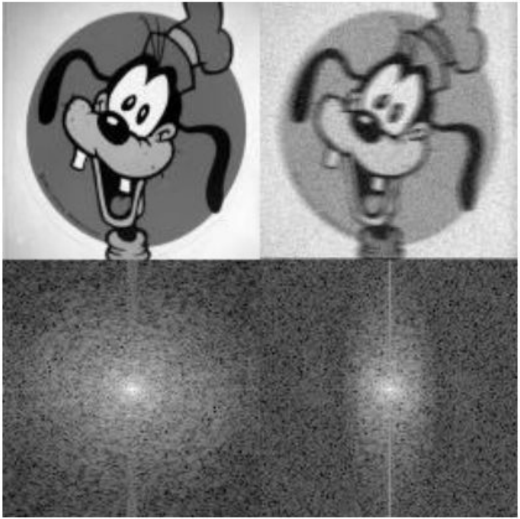
- Em um exemplo abaixo, o cosseno horizontal tem uma transformada normal e simples
- No outro exemplo, o cosseno rotacionado tem uma transformada complexa, com um componente diagonal forte e também um componente horizontal e vertical



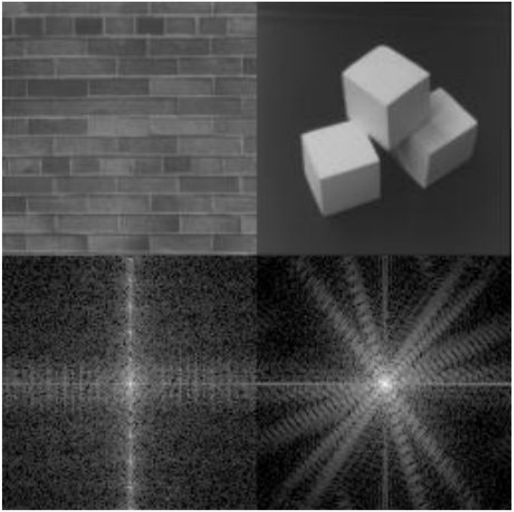
- Algumas imagens representadas como funções bidimensionais e seus espectros de Fourier



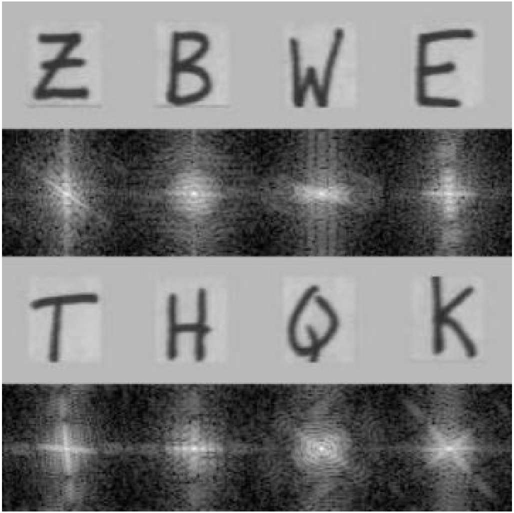
- Exemplo (1)



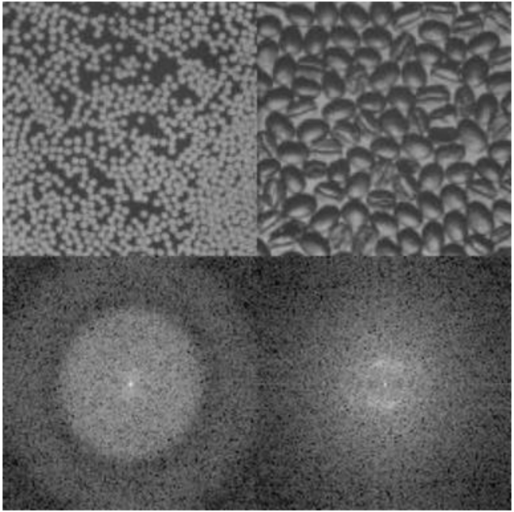
- Exemplo (2)



- Exemplo (3)



- Exemplo (4)



- Definir um filtro no domínio de frequência corresponde a encontrar uma máscara para ser utilizada em conjunto com a imagem transformada e assim obter a imagem filtrada desejada

Sumário

Transformada de Fourier 2D

Transformada de Fourier 2D

- A transformada discreta de Fourier 2-D (DFT) é dada por:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M+vy/N)}$$

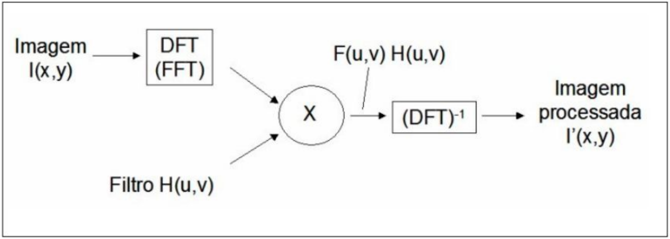
onde $f(x,y)$ é uma imagem digital de tamanho $M \times N$

- Dada a transformada $F(u,v)$, podemos obter $f(x,y)$ usando a transformada inversa discreta de Fourier (IDFT):

$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) e^{j2\pi(ux/M+vy/N)}$$

- Passo a passo da filtragem no domínio da frequência
  a) Dada uma imagem $f(x, y)$ de tamanho $M \times N$, obter uma imagem com entorno $fp(x, y)$ de tamanho $P \times Q$ onde $P = 2M$ e $Q = 2N$ preenchida com zeros no entorno
  b) Multiplicar $fp(x, y)$ por $(-1)^{(x+y)}$ para centralizar a transformada
  c) Aplicar a transformada de Fourier na imagem obtida em b)
  d) Criar um filtro simétrico $H(u, v)$ de tamanho $P \times Q$ com centro em $(P/2, Q/2)$
  e) Fazer o produto $H(u, v) * F(u, v)$ (resultado da transformada em c))
  f) Obter a imagem no domínio espacial fazendo o inverso da transformada de Fourier gp(x,y)
  g) Multiplicar $gp(x, y)$ por $(-1)^{(x+y)}$
  h) Remover o entorno da imagem obtendo $g(x, y)$ que é a imagem filtrada

- Algoritmos
  - FFT (*Fast Fourier Transform* – Transformada Rápida de Fourier)
    - É o algoritmo mais eficiente
    - Existem diversas implementações e variações
  - IFFT (*Inverse Fast Fourier Transform* – Transformada Rápida de Fourier Inversa)

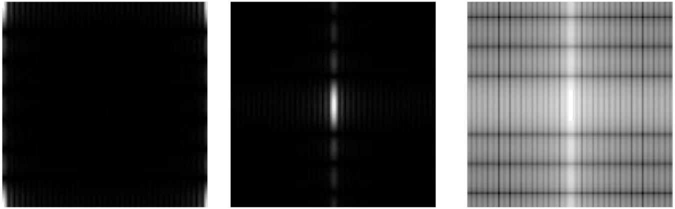


Transformada de Fourier 2D

- Devido às propriedades, a visualização fica melhor se efetuarmos algumas transformações

- Reposicionar os quadrantes das frequências (FFT *shift*) multiplicando a imagem f por $-1 ** (x + y)$ antes do FFT ou trocando os quadrantes de posição após o FFT

- A transformação logarítmica atenua a diferença com o componente de frequência zero

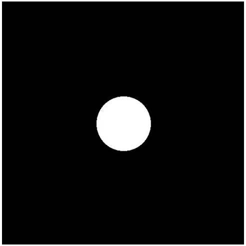


$|F(u,v)|$ $\quad$ $Shift(|F(u,v)|)$ $\quad$ $\log(1 + Shift(|F(u,v)|))$

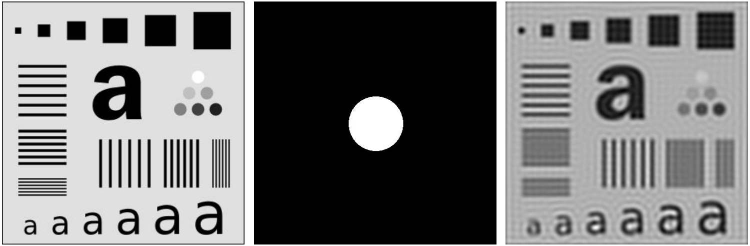
Sumário

Filtragem no Domínio da Frequência

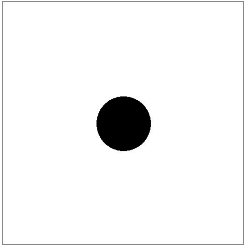
- Filtro passa-baixa (*low-pass filter*):
  - Remove frequências altas, deixando passar as mais baixas
  - O filtro passa-baixa "ideal" remove todas as frequências além de um certo limiar



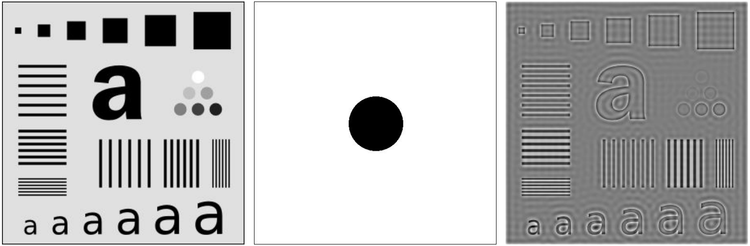
- Filtro passa-baixa (*low-pass filter*)



- Filtro passa-alta (*high-pass filter*):
  - Remove frequências baixas, deixando passar as mais altas
  - O filtro passa-alta "ideal" remove todas as frequências abaixo de um certo limiar

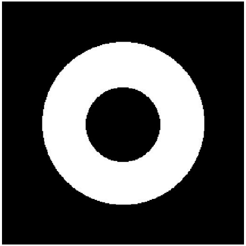


- Filtro passa-alta (*high-pass filter*):

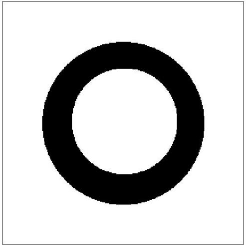


Filtragem no Domínio da Frequência

- Filtro passa-banda (*bandpass filter*):
  - Seleciona apenas uma fração das frequências que serão mantidas



- Filtro rejeita-banda (*band-stop filter*):
  - Remove apenas uma fração das frequências



Referências I

Borges, L. E.
Python para desenvolvedores.
Novatec, 2017.

Bradski, G.
The openCV library.
Dr. Dobb's Journal: Software Tools for the Professional Programmer, v. 25, n. 11, p. 120-123, 2000.

Gonzalez, Rafael C., e Richard E. Woods.
Processamento de imagens digitais.
Editora Blucher, 2000.

Referências II

Oliveira, M. M.
Notas de aula – Fundamentos de Processamento de Imagens.
Porto Alegre, 2010.
Disponível em: http://www.inf.ufrgs.br/~oliveira/Cursos/INF01046/INF01046_descricao_2010_2.html.
Acesso em: 10/12/2021.

Villán, A. F.
Mastering OpenCV 4 with Python.
Packt Publishing, 2019.

Wiggers, K. L.
Notas de aula – Processamento de Imagens: filtragem no domínio de frequência. Pato Branco, PR, 2024.