

Processamento de Linguagem Natural e Imagens

Aula 7

James Moraes de Almeida

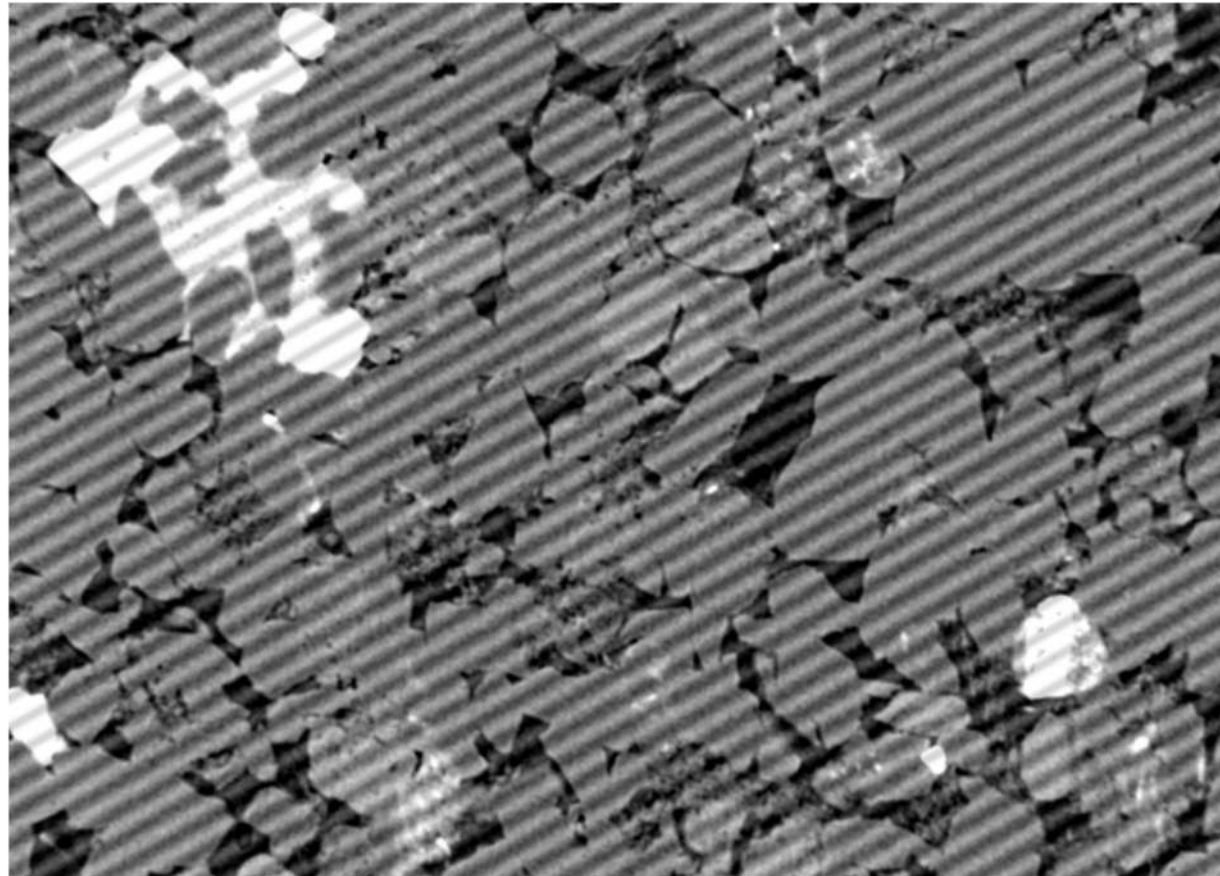


Conteúdo do curso

- Processamento de imagens.
 - 05.10.2026 - Aula 7 — imagem digital, filtros e Fourier.
 - 12.10.2026 - Semana da Primavera.
 - 19.10.2026 - Aula 8 — segmentação e quantificação de microscopia.
 - 26.10.2026 - Aula 9 — aprendizado de máquina para imagens.
 - 02.11.2026 - Feriado.
 - 09.11.2026 - Evento.
 - 16.11.2026 - Aula 10 — redes neurais convolucionais.
 - 23.11.2026 - Fechamento e relatório.
- Material principal: Python for Microscopists.
 - https://github.com/bnsreenu/python_for_microscopists
 - <https://www.youtube.com/c/digitalsreeni>

Um enigma

- Uma micrografia de arenito com listras de interferência.
- Defeito comum em microscopia eletrônica e de força atômica.
- Dá para tirar as listras com um filtro?



O que são imagens?

- Imagens tem um significado muito amplo.
 - Vocês já devem saber muito melhor que eu.
- Vamos focar em imagens digitais.
 - Vetoriais.
 - svg, eps, ps, ...
 - Rasterizadas.
 - png, jpeg, tiff, bmp, ...



O que são imagens?

- Imagens tem um significado muito amplo.
 - Vocês já devem saber muito melhor que eu.
- Vamos focar em imagens digitais.
 - Durante o curso, utilizaremos imagens rasterizadas.

O que são imagens?

- Primeira imagem digital:
 - Standards Eastern Automatic Computer, SEAC, NIST, 1957.
 - 176x176 pixels, foto escaneada.

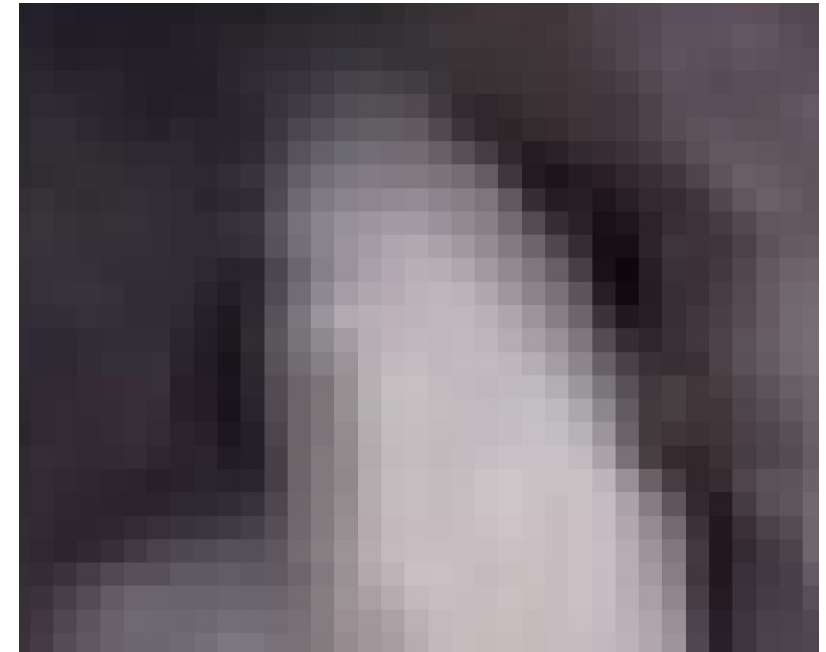


O que são imagens?

- O primeiro sensor digital só foi feito em 1969 (Bell Labs).
 - Charge-Coupled Device (CCD).
- Tecnologia com baixo atraso e ruído só foi inventada em 1980 (NEC).

O que são imagens?

- Vamos entender como o computador interpreta imagens rasterizadas.



O que são imagens?

- Vamos entender como o computador interpreta imagens rasterizadas.

Vermelho



Verde

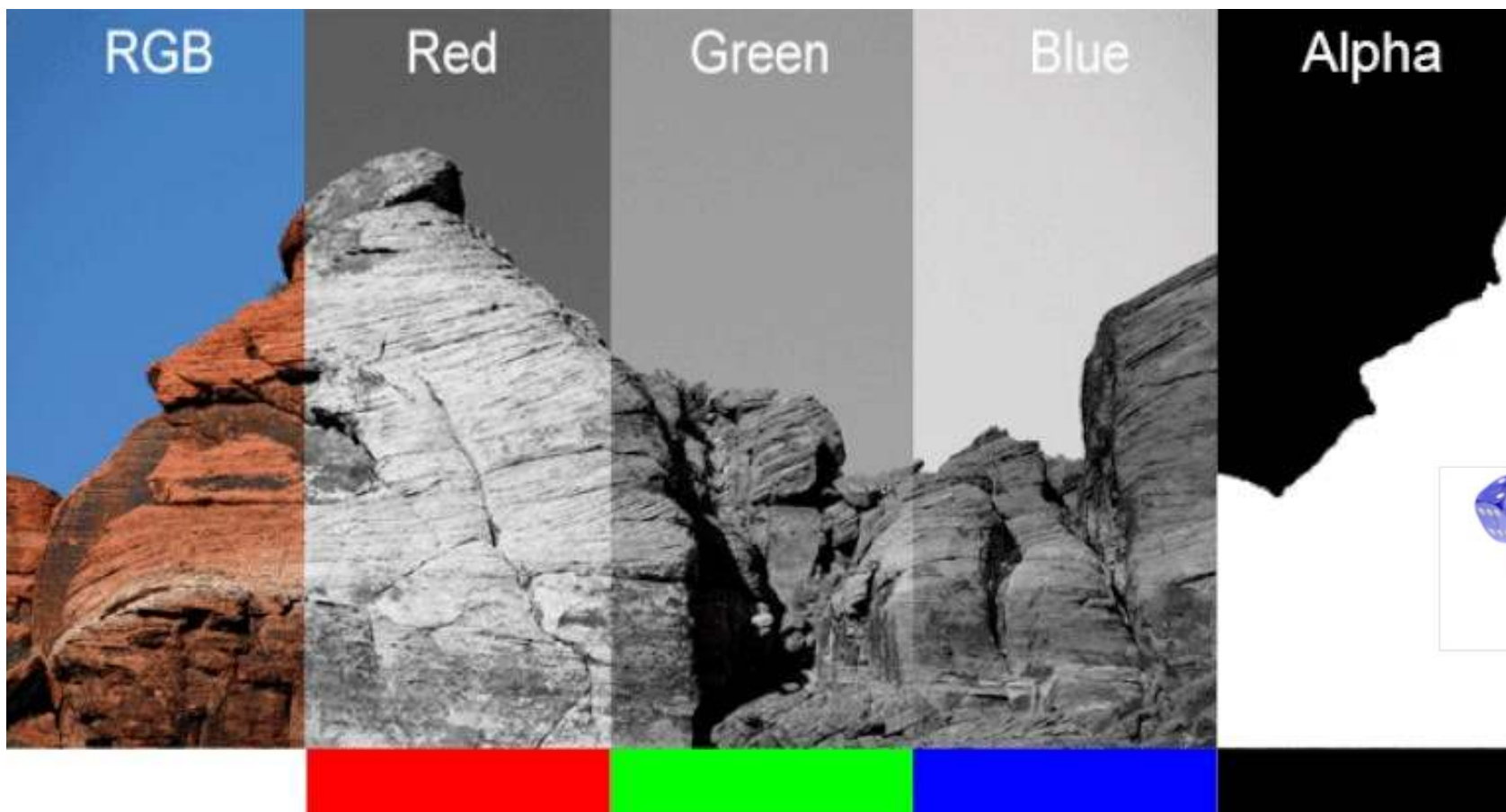


Azul



O que são imagens?

- Vamos entender como o computador interpreta imagens rasterizadas.
- Algumas imagens tem um canal de transparência.



1



2

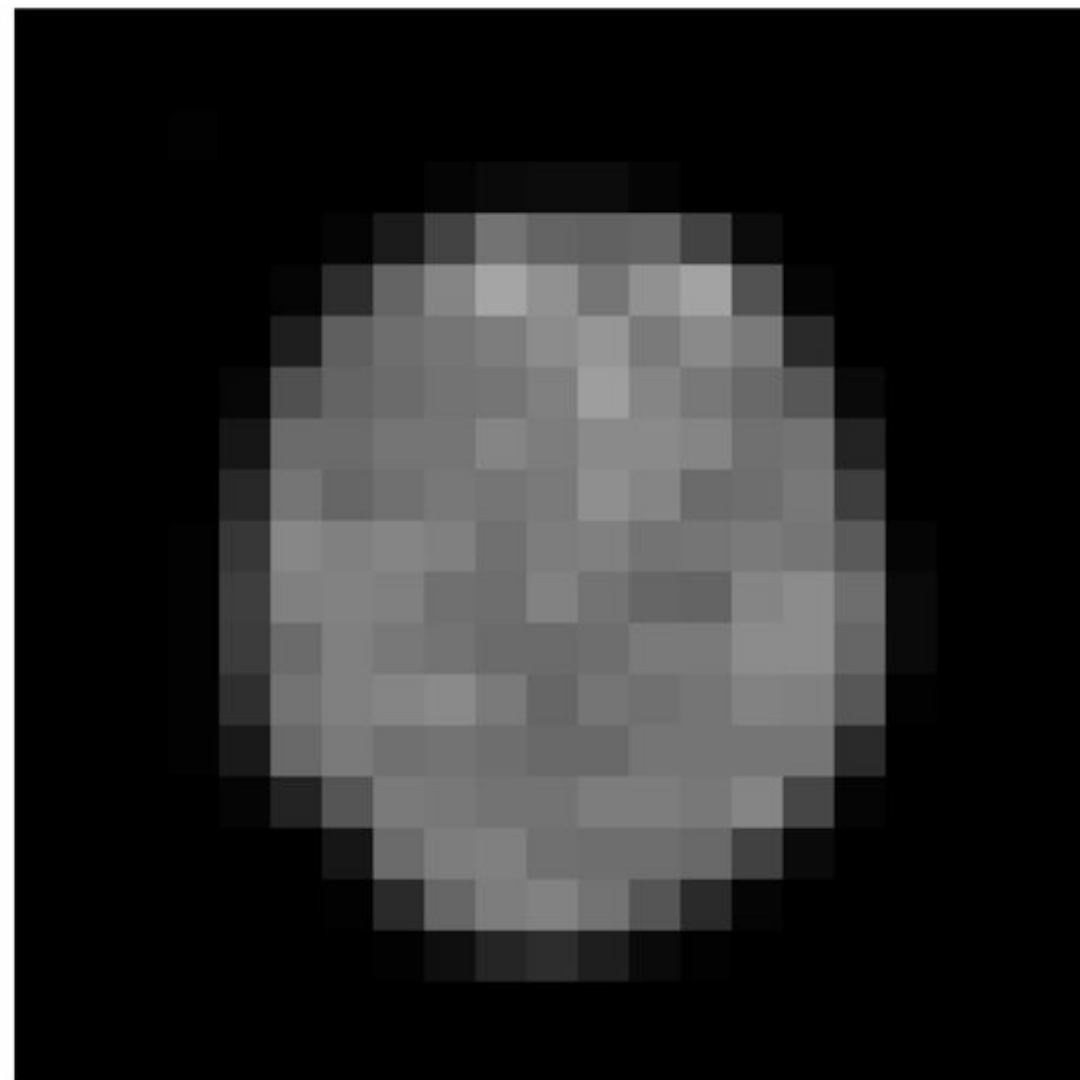


3

O que são imagens?

- Então. é uma matriz de intensidades. para cada cor.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	1	1	2	1	1	1	2	1	2	1	2	1	1	1	1	1	1	1	1	1
2	2	1	2	3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
3	2	1	1	2	2	2	2	2	4	6	7	7	4	2	2	1	1	1	1	1	1
4	2	2	2	2	2	2	4	13	29	48	42	41	42	29	7	2	1	1	1	1	1
5	2	2	2	2	2	4	20	42	55	68	60	49	60	67	35	4	1	1	1	1	1
6	1	2	2	2	2	14	40	46	49	52	58	62	51	57	51	19	2	1	1	1	1
7	1	1	2	2	5	34	42	45	48	49	53	65	55	50	44	37	6	1	1	1	1
8	2	2	2	2	11	45	45	49	49	55	52	57	57	55	47	48	16	2	1	1	1
9	1	1	2	2	18	49	43	47	50	49	51	59	55	45	46	50	27	2	1	1	1
10	1	1	2	3	24	56	53	55	53	47	52	53	48	49	51	49	38	4	1	1	1
11	1	2	2	3	27	53	54	53	47	46	54	48	43	42	55	58	46	6	1	1	1
12	1	2	2	3	26	45	53	50	48	45	45	46	51	51	58	58	43	6	2	1	1
13	1	1	2	3	21	48	53	55	57	51	43	49	47	49	54	53	37	3	1	1	1
14	1	1	2	3	12	44	51	47	48	46	44	44	49	49	49	49	19	2	2	1	1
15	2	1	1	2	4	16	36	51	50	48	48	52	52	50	55	30	4	2	2	2	1
16	1	1	1	2	2	2	11	45	52	53	47	46	46	44	28	6	2	2	2	1	1
17	1	1	1	2	2	1	3	19	43	52	54	48	36	19	4	2	2	1	1	1	1
18	1	1	1	1	2	1	2	3	8	17	20	14	6	3	2	1	1	1	2	1	1
19	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	2	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

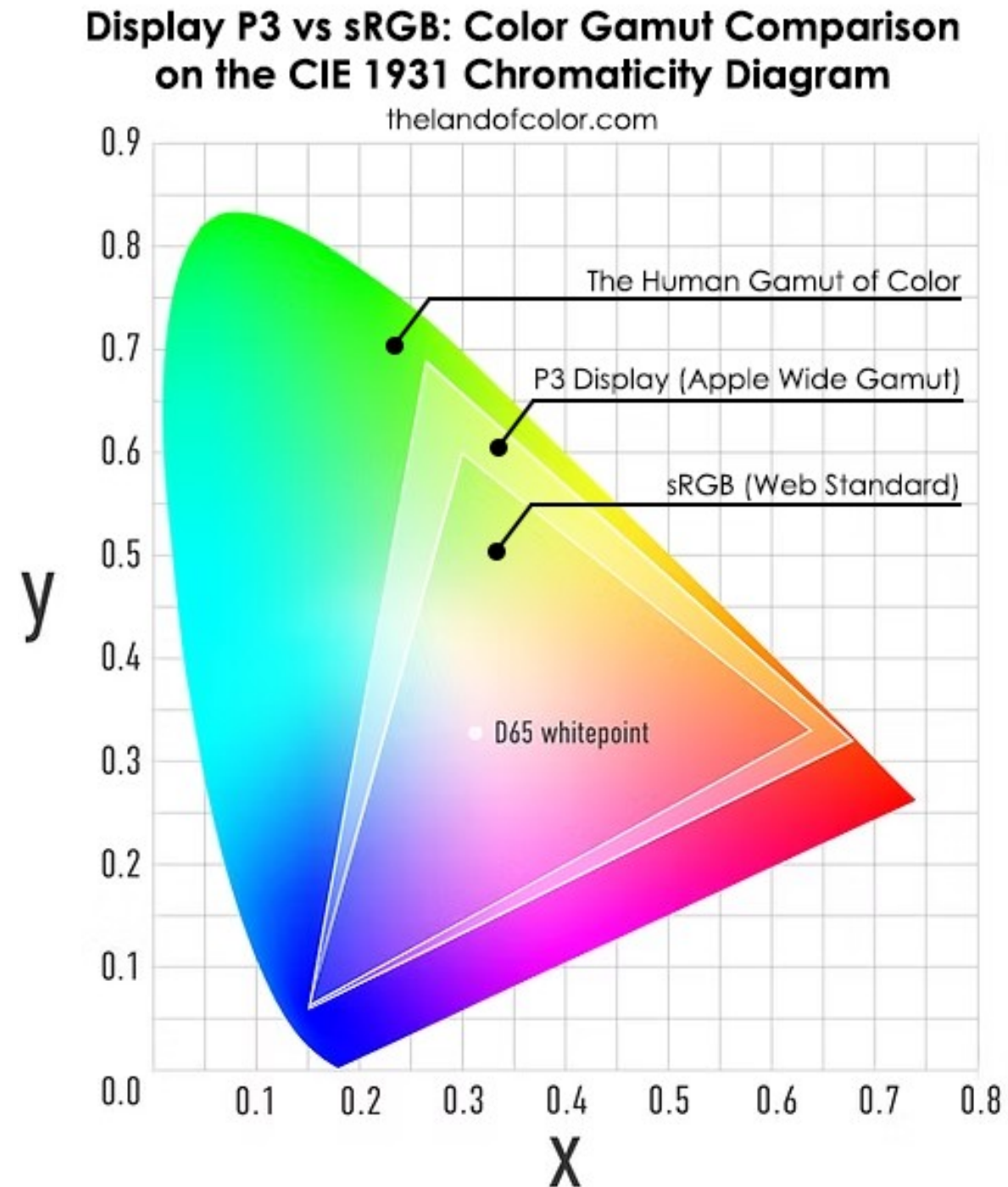


O que são imagens?

- Então, é uma matriz de intensidades, para cada cor.
- A escala varia:
 - 8 bit color, essa escala vai de 0 a 255 ($2^8 = 256$ níveis).
 - 16 bit high color, 65.536 níveis.
 - 24 bit true color, 16.777.216.
 - São na verdade 8 bits por cada canal de cor.
 - É o mais usado na maioria das imagens e telas.
 - 30-48 bit deep color, 1.073.741.824 - 281.474.976.710.656.

O que são imagens?

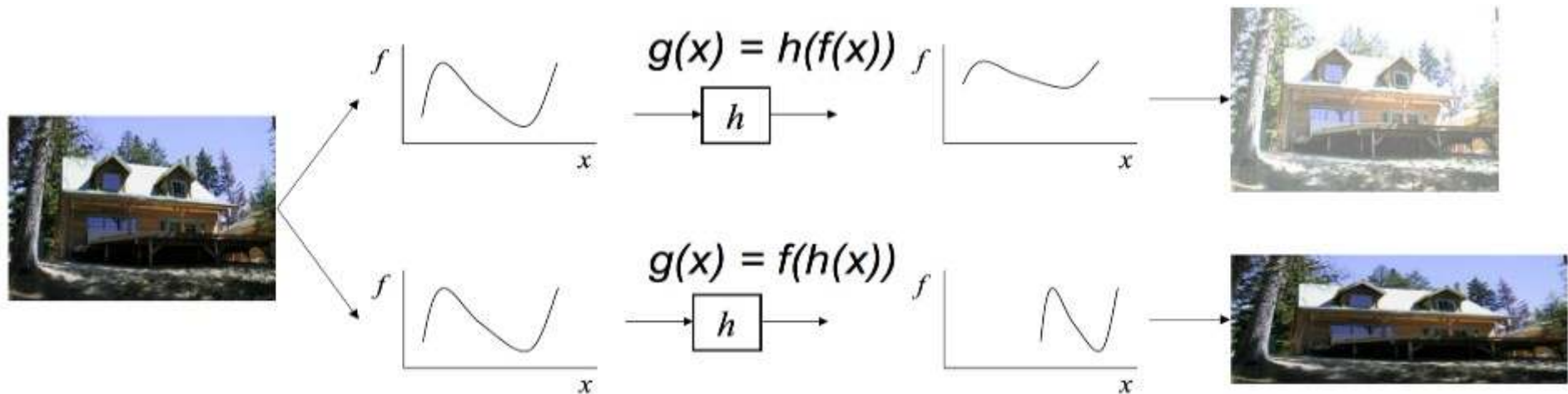
- O que os dispositivos mostram?



Filtros de imagens.

- Basicamente é uma função que atua nas matrizes das imagens.

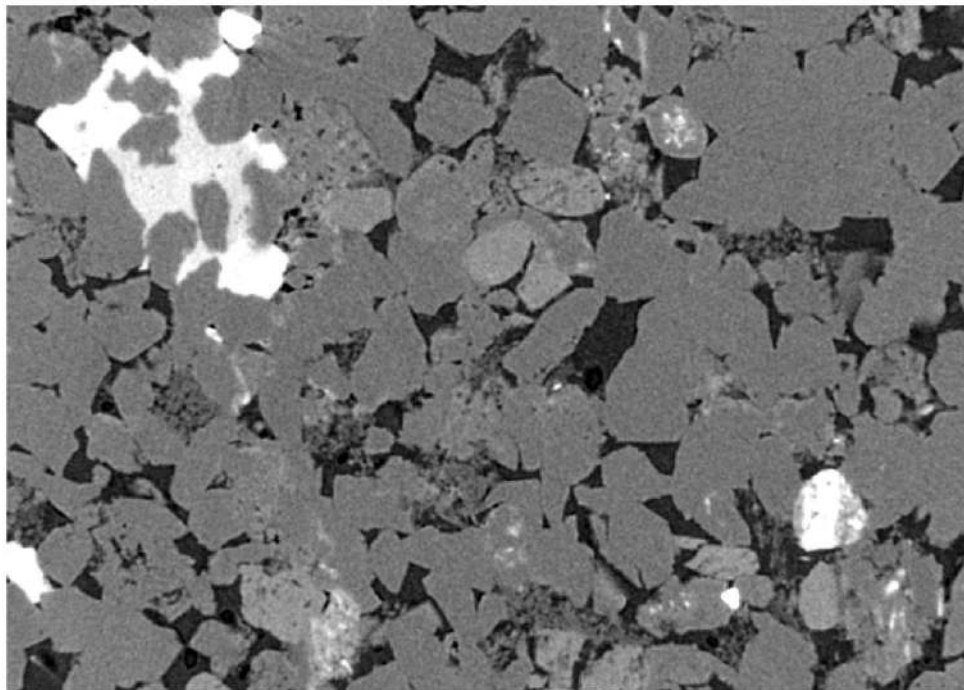
$$f(x, y) = \begin{bmatrix} r(x, y) \\ g(x, y) \\ b(x, y) \end{bmatrix}$$



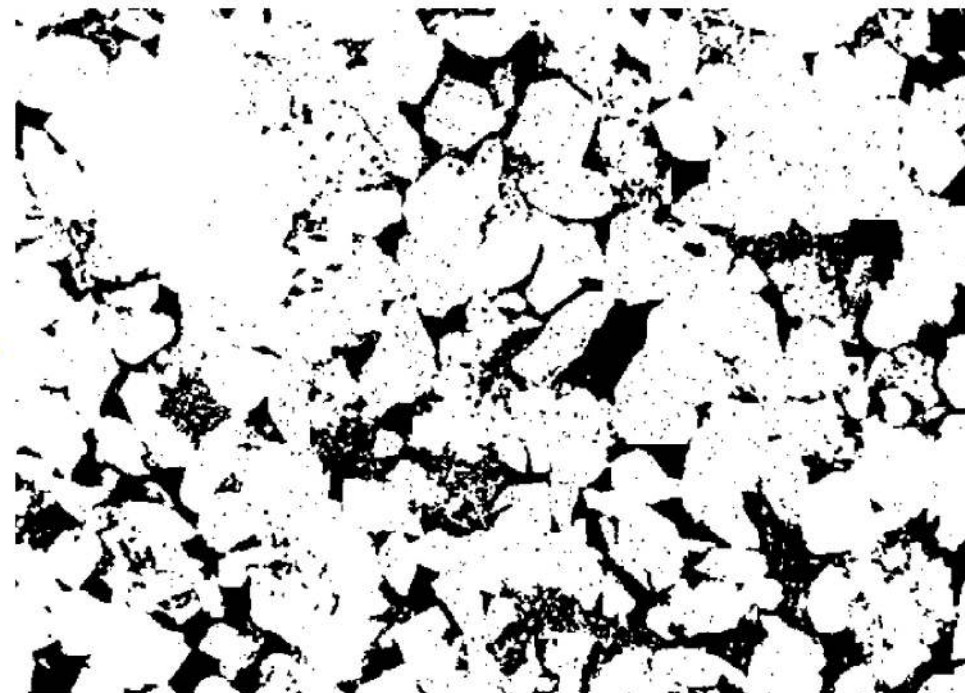
Filtros de imagens.

- O mais simples: decidir pixel a pixel, por um valor de corte.

arenito



pixels acima de 100



$$g = \begin{cases} 255, & f > 100 \\ 0, & \text{caso contrário} \end{cases}$$

Um filtro é uma janela que desliza

arenito: linha 410, coluna 224

49	85	131	171	196	200
0	16	93	165	203	211
0	20	91	168	206	208
48	43	108	167	190	192
70	61	119	160	177	182
76	61	113	153	166	169

kernel de Sobel

-1	0	1
-2	0	2
-1	0	1

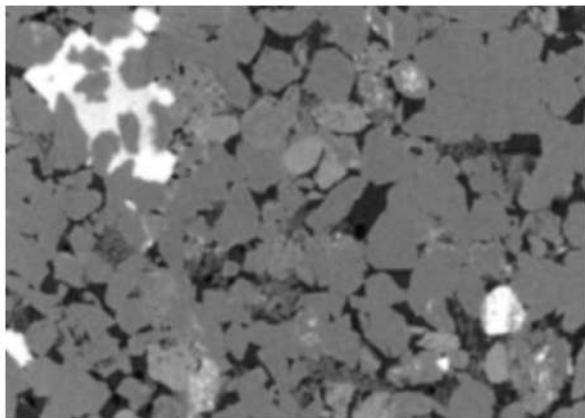
saída

Troca só os pesos: menos à esquerda, mais à direita. Agora a conta mede uma diferença.

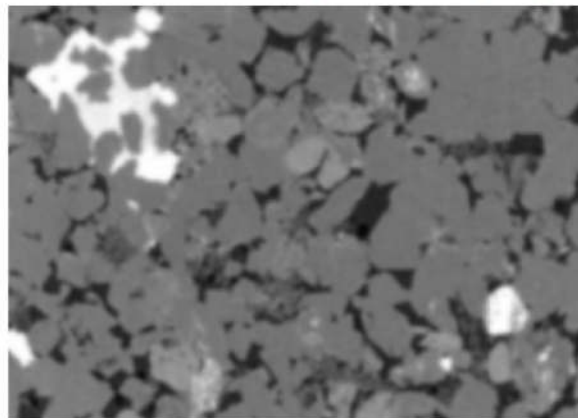
Filtros de imagens.

- Realce: o filtro unsharp soma de volta o detalhe que o gaussiano tirou.

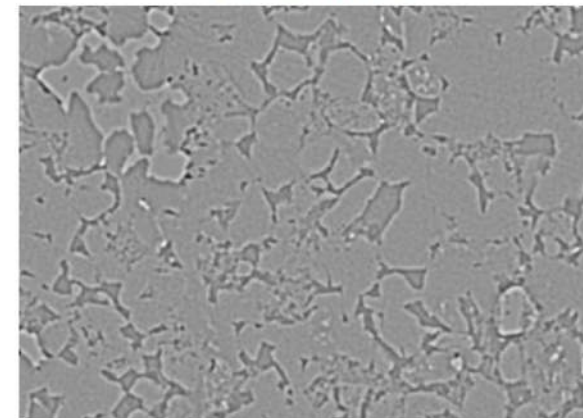
original



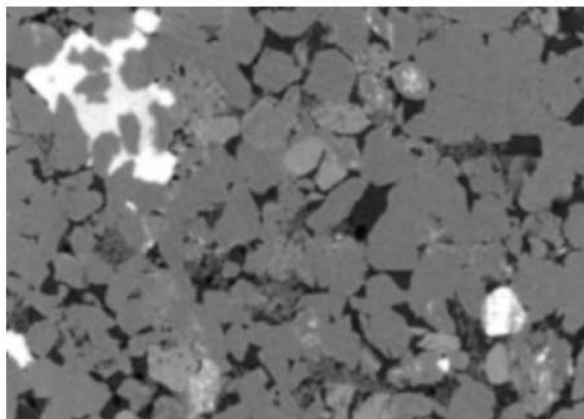
suavizada ($\sigma = 2$ px)



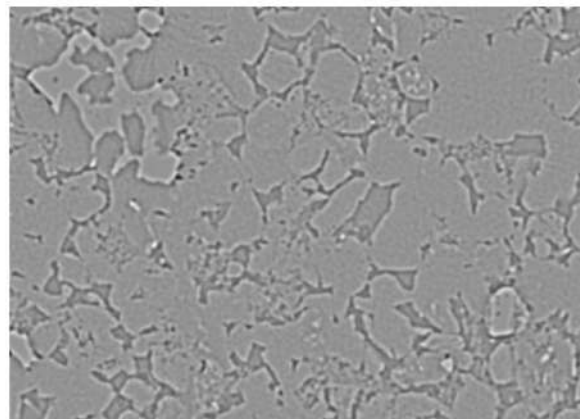
detalhe



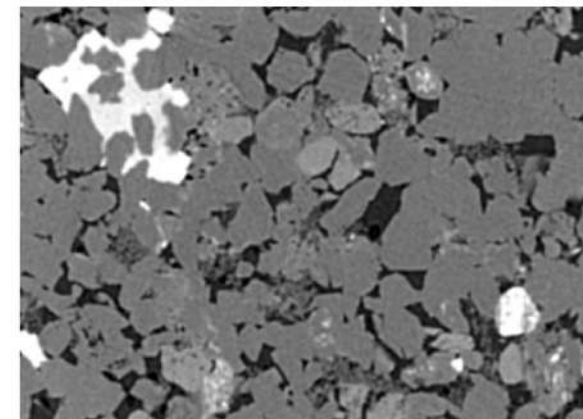
original



detalhe



realçada

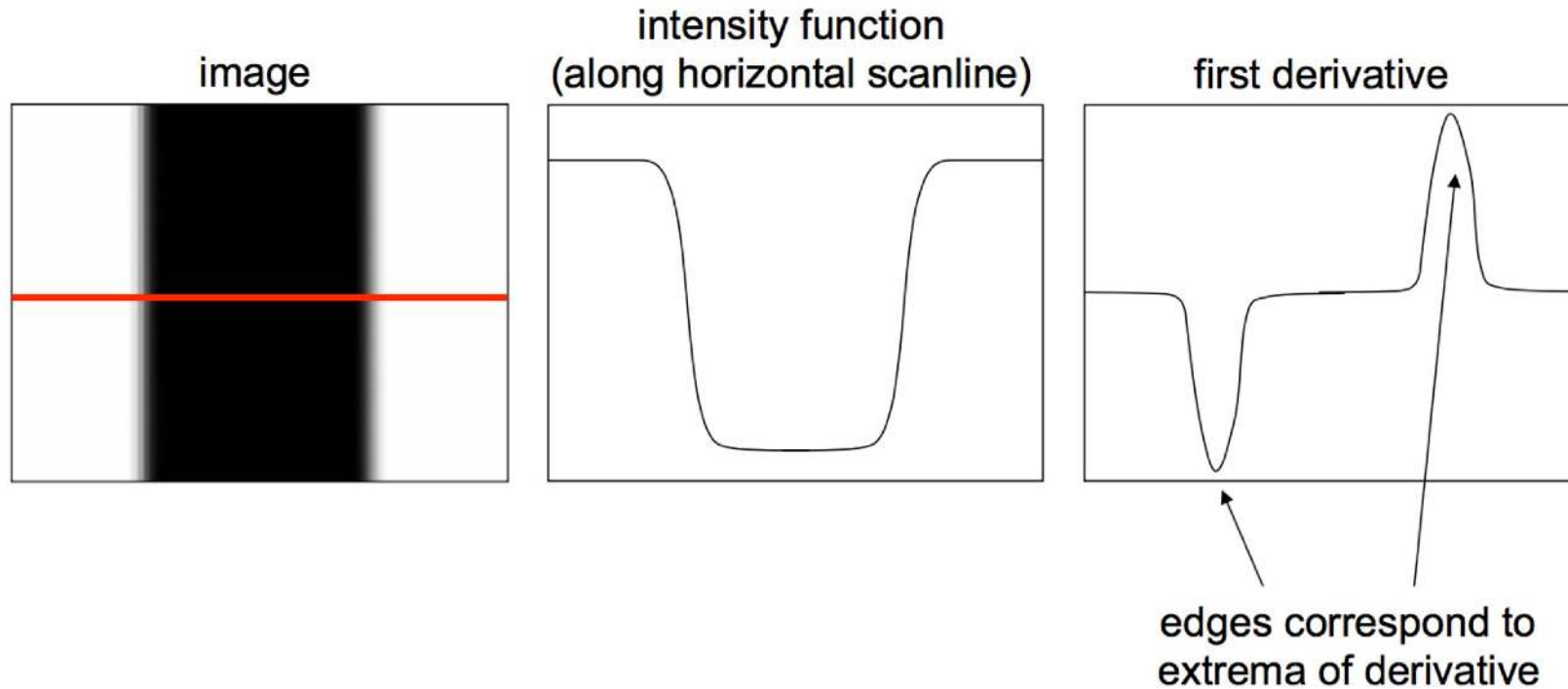


+ 2 ×

=

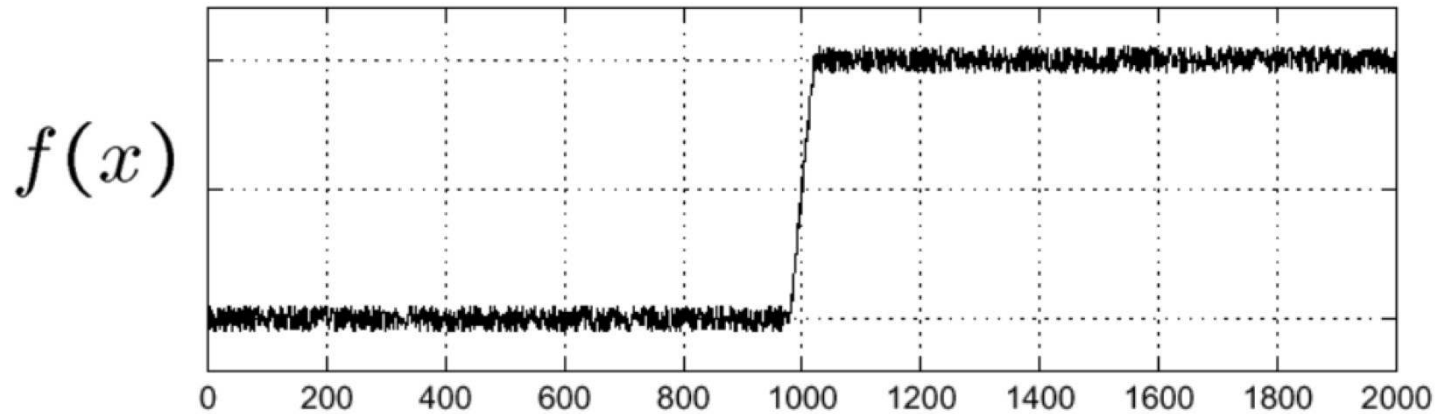
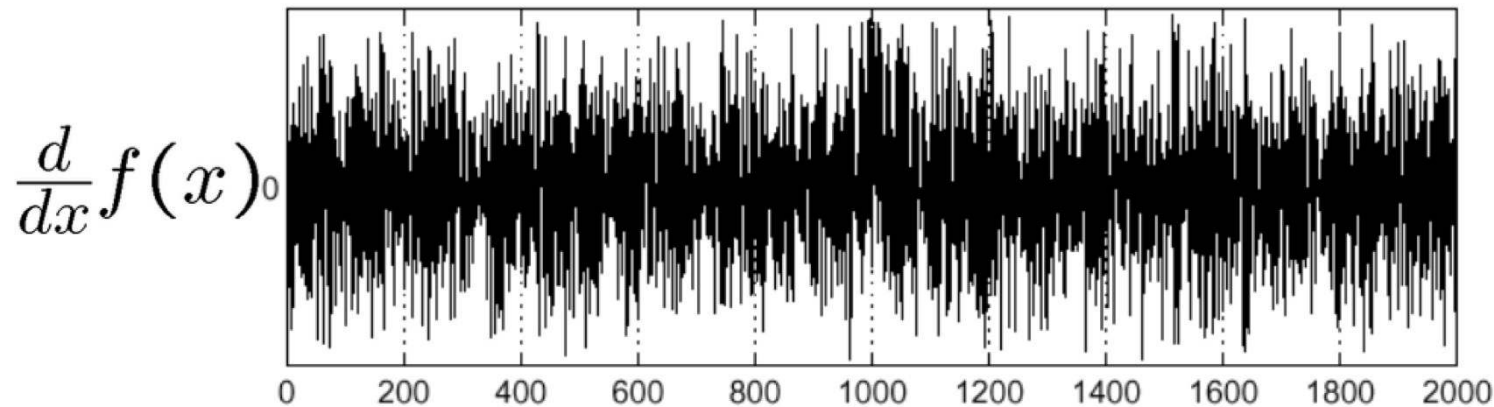
Filtros de imagens.

- Obtendo detalhes da imagem (bordas).
- Existem muitos métodos para detectar bordas.
 - Por exemplo, fazendo sua derivada.



Filtros de imagens.

- Obtendo detalhes da imagem (bordas).
- Existem muitos métodos para detectar bordas.
- Mas as imagens costumam ter ruído.



Filtros de imagens.

- Obtendo detalhes da imagem (bordas).
- Existem muitos métodos para detectar bordas.
- Mas as imagens costumam ter ruído.
- Então filtros mais complexos são utilizados.
- Desfoque Gaussiano:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

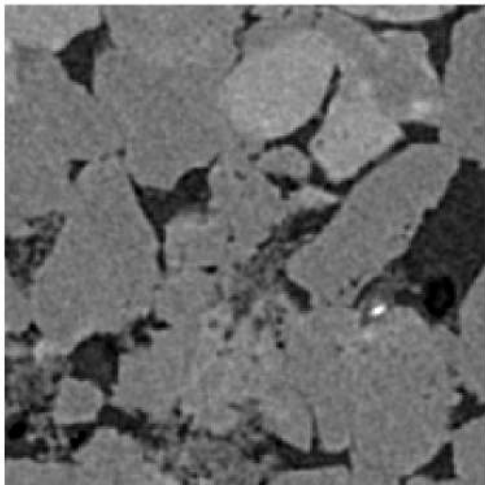
- Kernel 3x3:

$\frac{1}{16}$	1	2	1
	2	4	2
	1	2	1

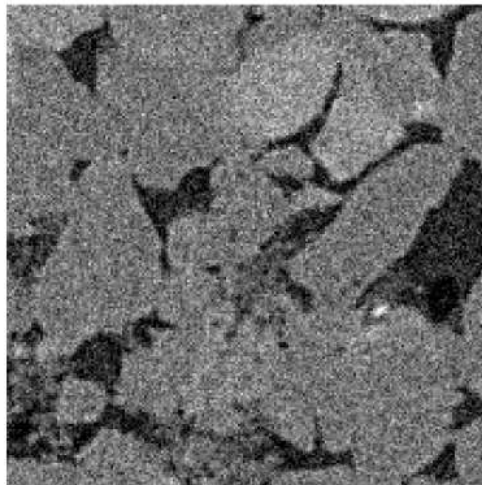
Filtros de imagens.

- Desfoque gaussiano: tira o ruído, mas σ grande demais leva o detalhe.

limpa

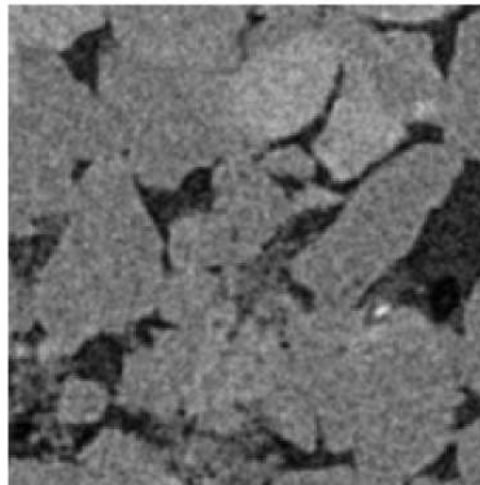


com ruído



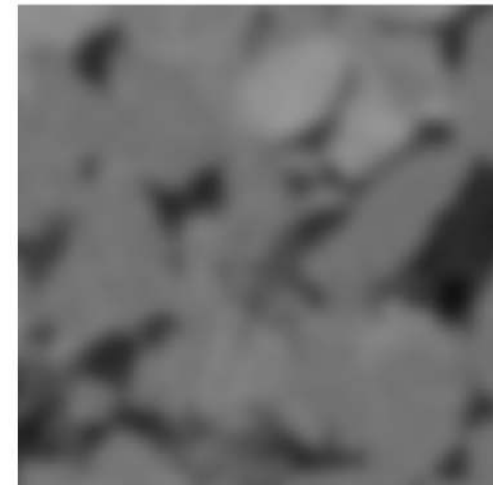
erro
24,6 níveis de cinza

kernel 3×3



erro
10,4 níveis de cinza

$\sigma = 4$ px



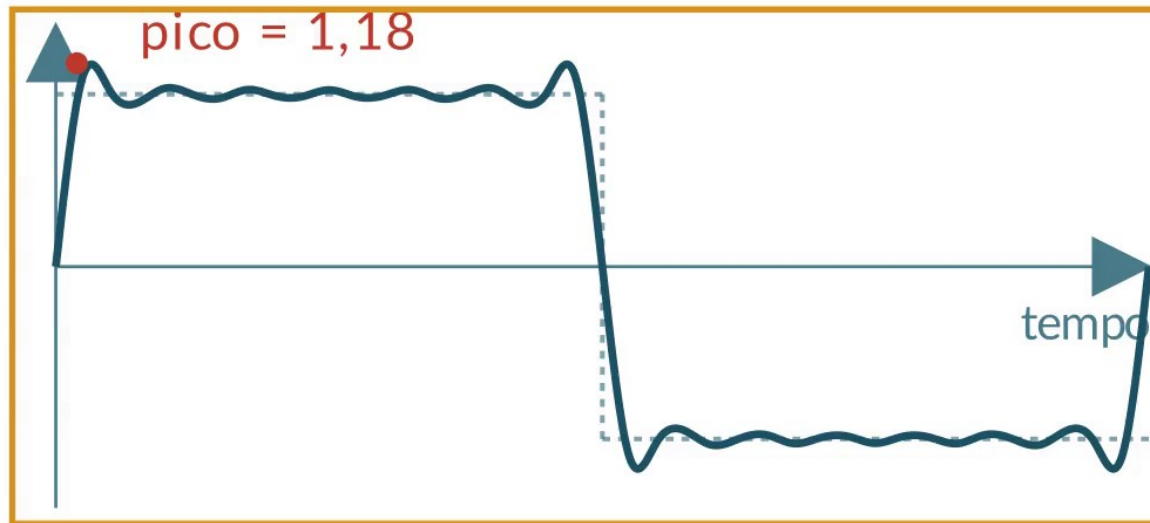
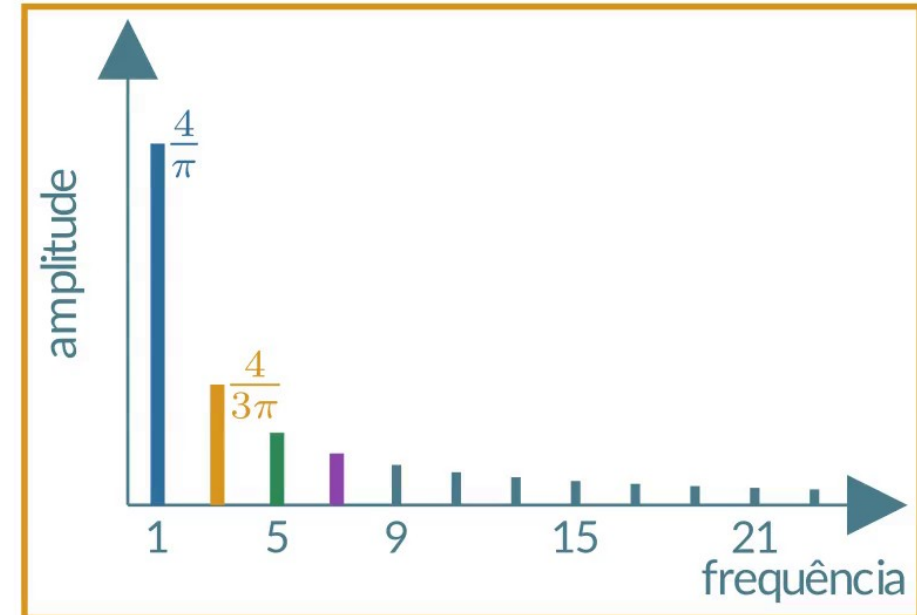
erro
20,0 níveis de cinza

ruído gaussiano de 25 níveis de cinza · erro: RMS contra a limpa, na imagem inteira (481×674 px)

Transformada de Fourier

- De onde vem?
 - Fourier estava tentando resolver a equação do calor.
 - Primeira publicação em 1807, teoria analítica em 1822.
- Propôs escrever uma função arbitrária como soma de senoides.
 - Para poder resolver a equação diferencial do calor.
- Hoje é a base de quase todo processamento de sinais e imagens.

Uma onda quadrada é uma soma de senoides



A mesma onda, dita de dois jeitos: no tempo, ou como uma lista de amplitudes.

Transformada de Fourier

- Uma função periódica escrita como soma de senoides:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(\omega_n t) + b_n \sin(\omega_n t)]$$

$$\omega_n = \frac{2\pi n}{T} \quad a_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \cos(\omega_n t) dt \quad b_n = \frac{2}{T} \int_{-T/2}^{T/2} f(t) \sin(\omega_n t) dt$$

Transformada de Fourier

- Forma exponencial, pelas relações de Euler:

$$\cos \theta = \frac{e^{i\theta} + e^{-i\theta}}{2} \quad \text{sen } \theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}$$

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{i\omega_n t}$$

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-i\omega_n t} dt$$

Transformada de Fourier

- E se a função não for periódica? O período tende ao infinito,
 - e a soma vira uma integral: a transformada de Fourier.

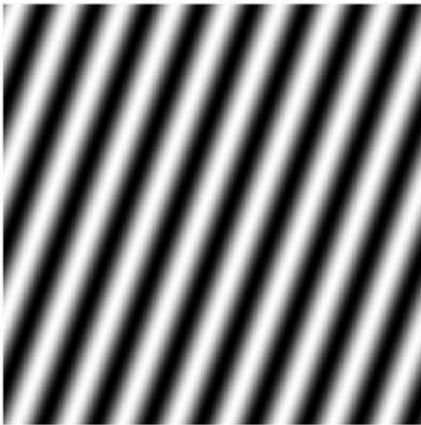
$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad \text{do tempo para a frequência}$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega \quad \text{e de volta}$$

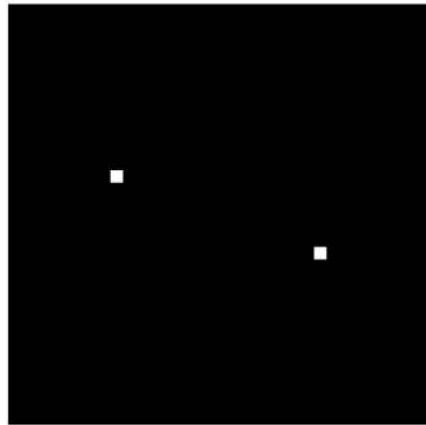
Transformada de Fourier

- Em imagens, a transformada é em duas dimensões.

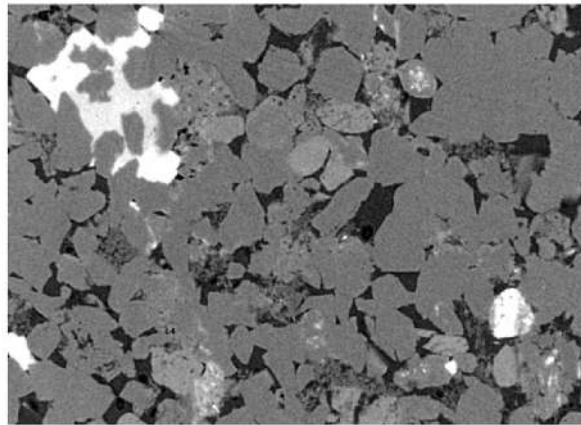
uma senoide



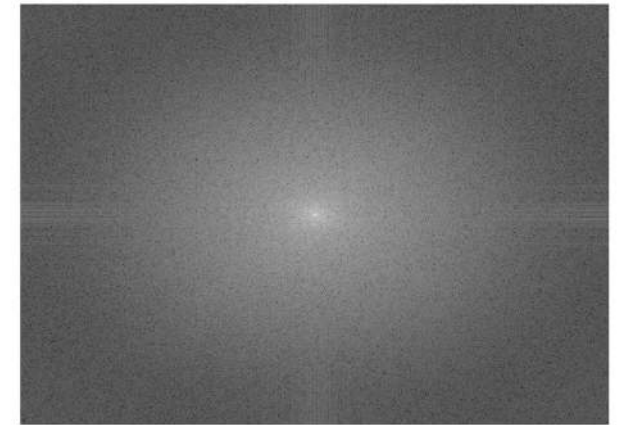
o espectro (ampliado)



arenito



o espectro dele

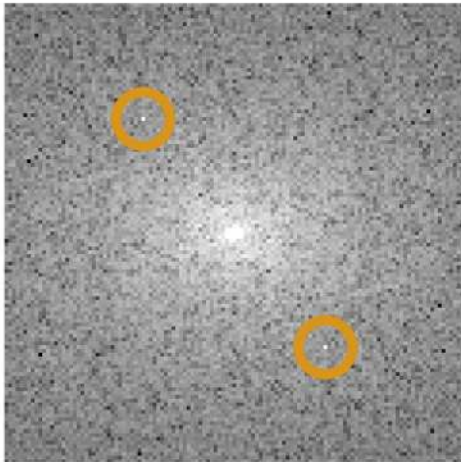


centro = brilho médio • longe do centro = detalhe fino

De volta ao enigma

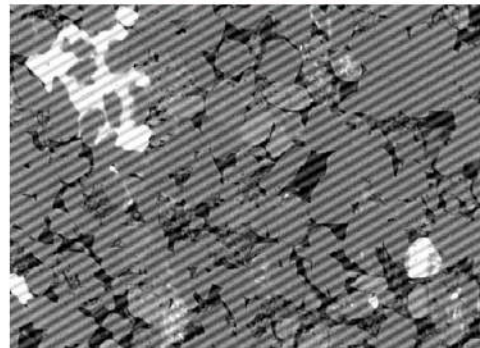
- A listra inteira mora em dois coeficientes do espectro.

centro do espectro (ampliado)



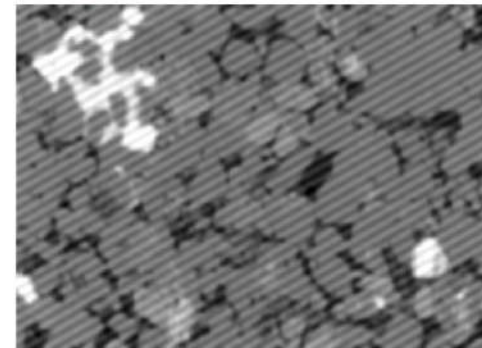
zerar 2 de 324.194 coeficientes

com listras



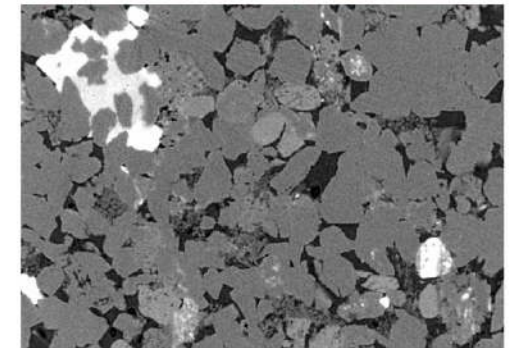
erro 28,3

gaussiano $\sigma = 3$ px



erro 20,5

2 coeficientes zerados



erro 0,41

erro RMS contra a imagem limpa, em níveis de cinza (0-255)

amplitude lida no espectro: 40,2 · injetada: 40 (níveis de cinza)

Vamos ao Python!

- Processamento de imagens: o que são imagens, filtros e Fourier.
- O notebook da aula está no Teams da disciplina.
 - Aula07_imagens_filtros_fourier.ipynb, com a pasta images/.
- Material original encontra-se em:
 - github.com/bnsreenu/python_for_image_processing_APEER

Vamos ao Python!

- No terminal (Anaconda prompt, no Windows):
 - `conda create -n img_proc -c conda-forge python=3.12 scikit-image opencv matplotlib scipy pandas ipykernel`
 - `conda activate img_proc`
 - `pip install bm3d` (opcional)

OBRIGADO

James Moraes de Almeida
james.almeida@ilum.cnpem.br

ilum.cnpem.br