

# Transformações de intensidade e histograma de imagens

---

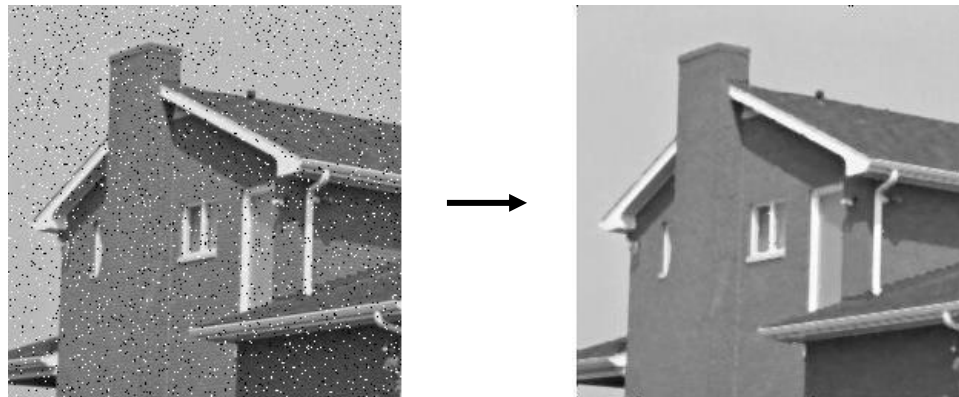
CESAR HENRIQUE COMIN

# Exemplos de transformação

Realce de contraste



Eliminação de ruído



# Transformação de imagens

- Utilizadas para diversos propósitos:
  - Tornar a imagem mais agradável visualmente
  - Facilitar a identificação visual ou automatizada de elementos da imagem
  - Diminuir a informação necessária para armazenar a imagem
  - Diminuir o número de níveis de intensidade para facilitar a aplicação de técnicas de processamento de imagens
  - etc

# Representação de uma imagem em nível de cinza



# Representação de uma imagem em nível de cinza



■ Preto, valor 0

□ Branco, valor 255

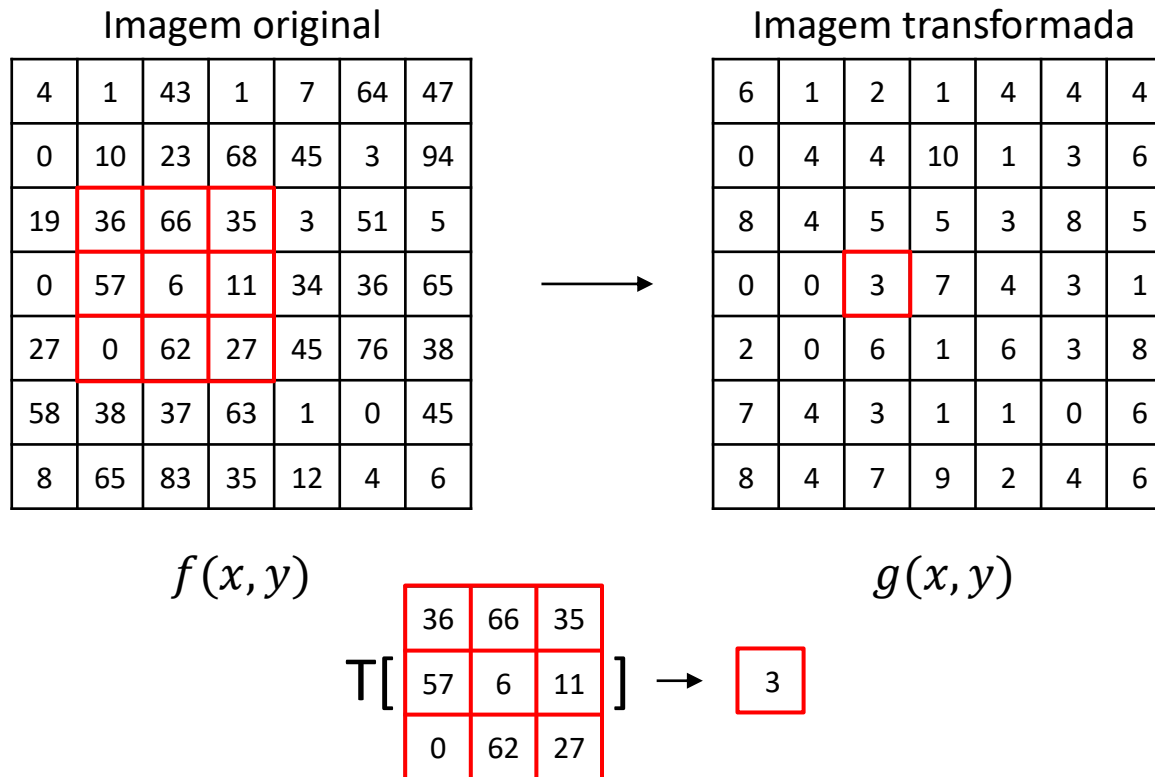


59	19	76	178	169	169	170	167	166	166
47	33	23	167	169	168	165	169	168	168
14	63	16	103	173	169	168	168	165	162
10	51	25	36	170	172	169	165	166	167
9	17	63	18	130	171	170	167	167	168
9	10	67	23	54	172	168	155	161	165
9	9	26	55	19	157	175	171	178	186
9	8	11	74	21	80	193	191	191	193
14	8	8	43	40	20	181	194	190	193
40	9	9	14	62	17	101	194	190	190

# Transformação de imagens

- Matematicamente, representamos a transformação como

$$g(x, y) = T[f(x, y)]$$



# Transformação de intensidade pontual

- Podemos definir uma função de transformação  $s = T(r)$  que é aplicada sobre cada pixel  $r$  da imagem
- Nesse caso, a transformação de cada pixel independe da sua vizinhança

Imagem original

4	1	43	1	7	64	47
0	10	23	68	45	3	94
19	36	66	35	3	51	5
0	57	6	11	34	36	65
27	0	62	27	45	76	38
58	38	37	63	1	0	45
8	65	83	35	12	4	6

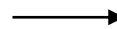
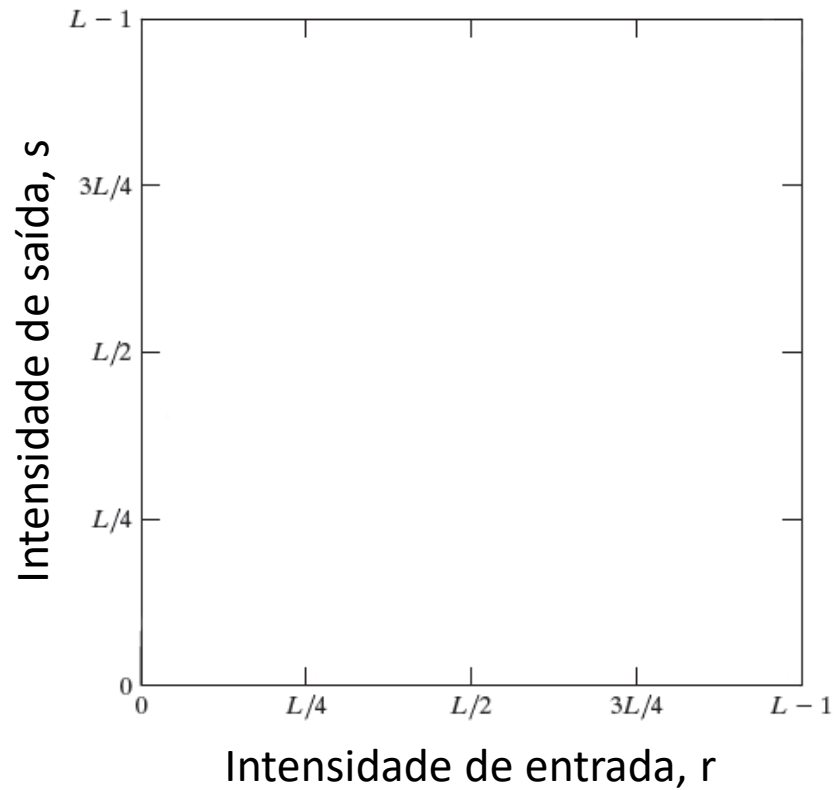
$$f(x, y)$$


Imagem transformada

6	1	2	1	4	4	4
0	4	4	10	1	3	6
8	4	5	5	3	8	5
0	0	3	7	4	3	1
2	0	6	1	6	3	8
7	4	3	1	1	0	6
8	4	7	9	2	4	6

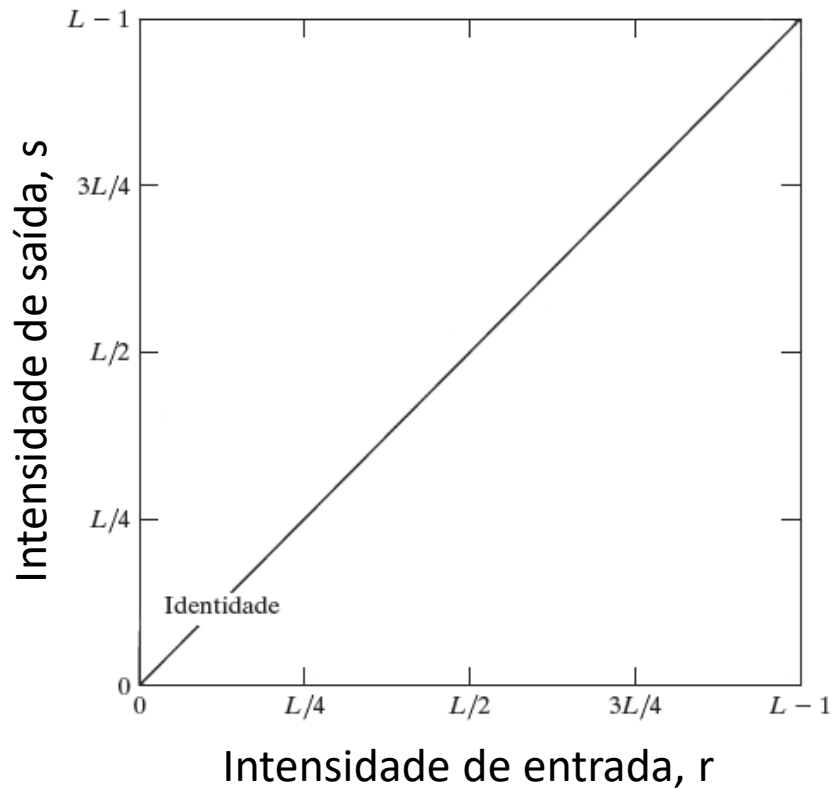
$$g(x, y)$$

# Transformações de intensidade pontuais comumente utilizadas



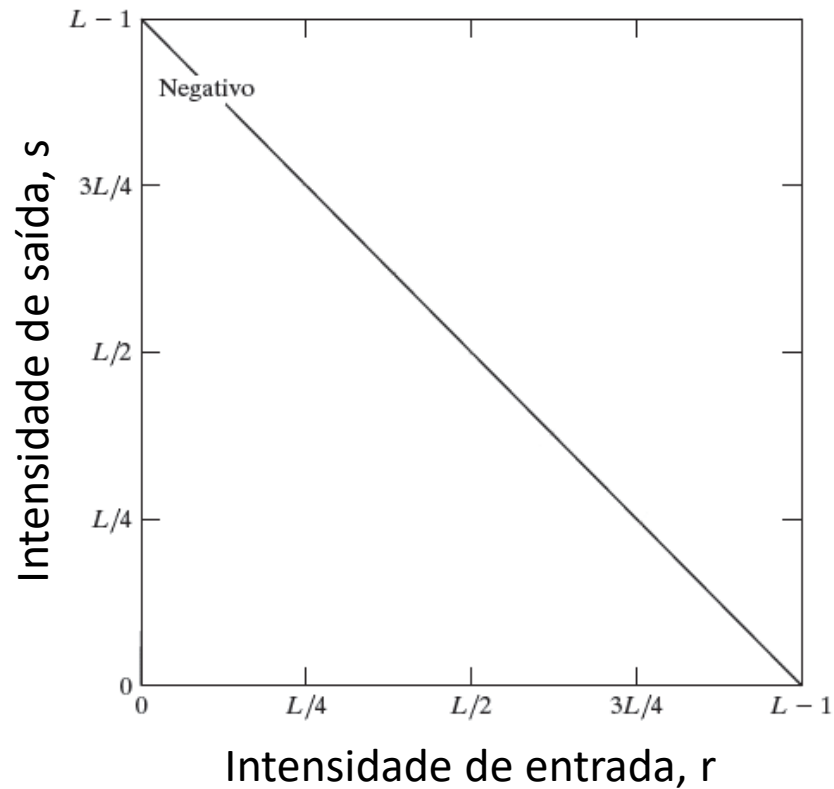


# Transformações de intensidade pontuais comumente utilizadas



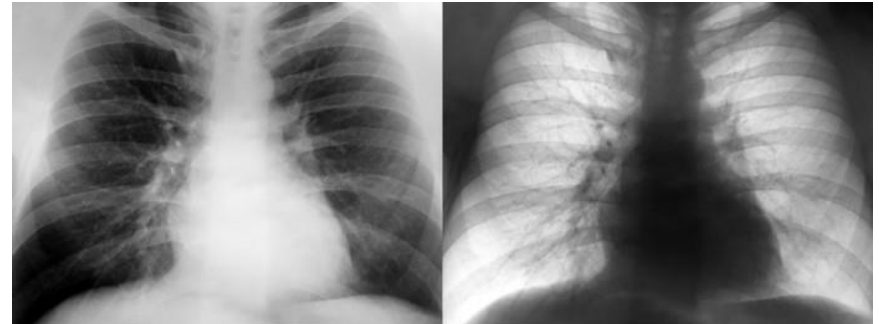
Identidade:  
O valor de saída é igual ao de  
entrada

# Transformações de intensidade pontuais comumente utilizadas

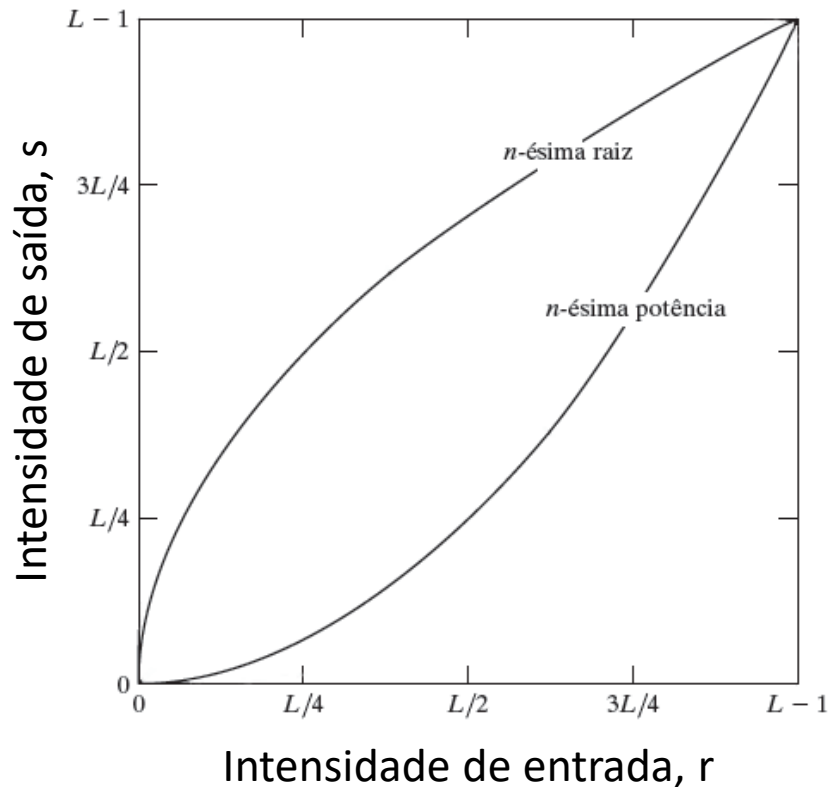


Negativo da imagem:

$$s = L - 1 - r$$



# Transformações de intensidade pontuais comumente utilizadas

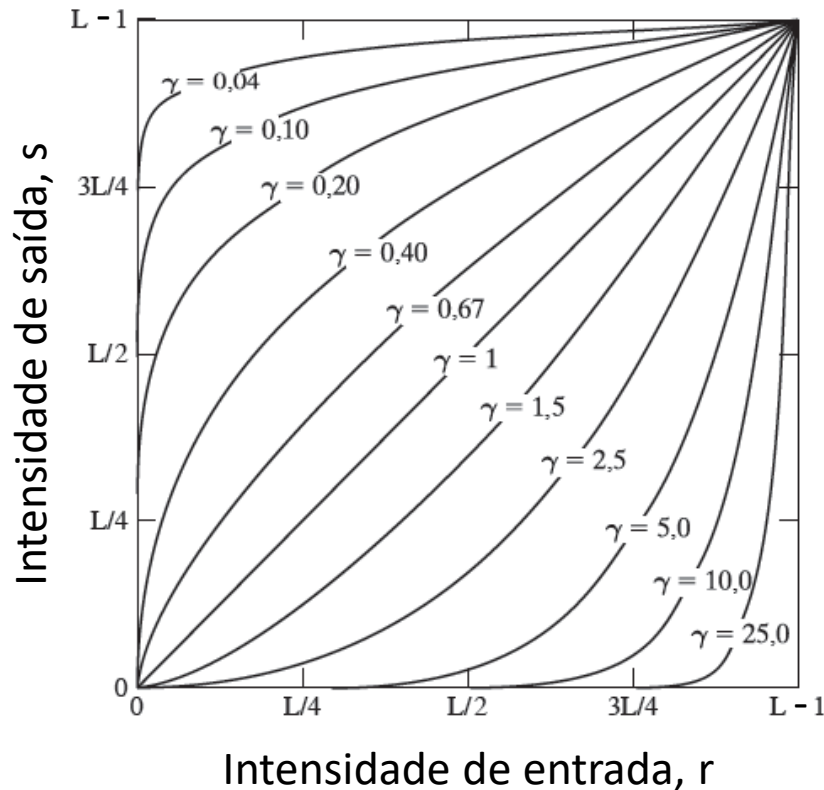


Transformação de potência (gamma)

$$s = (L - 1) \frac{r^\gamma}{(L - 1)^\gamma}$$

Pode ser ajustada através do parâmetro  $\gamma$

# Transformações de intensidade pontuais comumente utilizadas



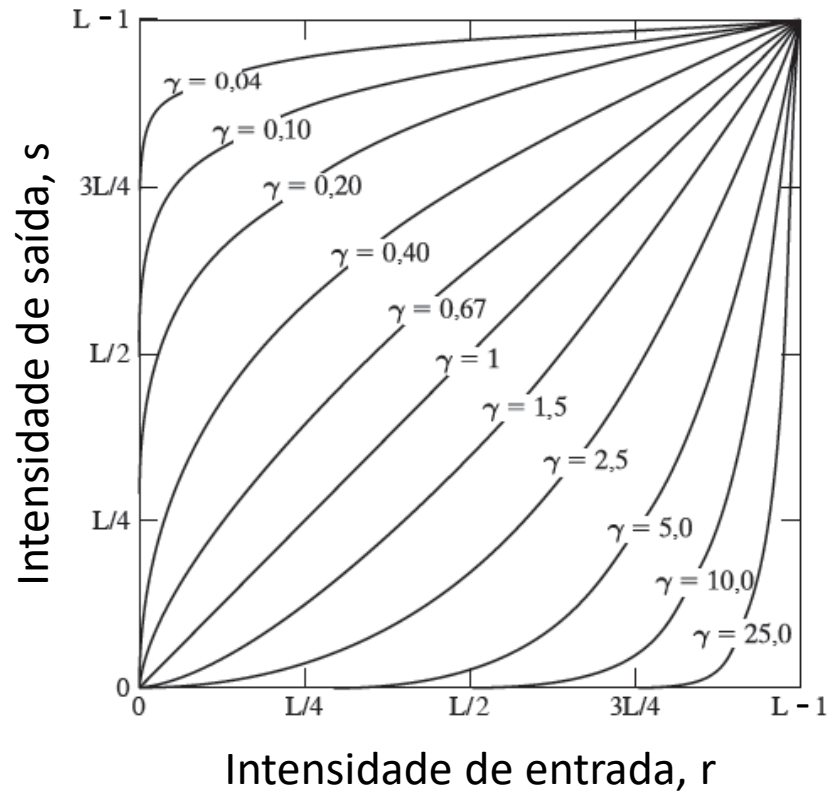
Transformação de potência (gamma)

$$s = (L - 1) \frac{r^\gamma}{(L - 1)^\gamma}$$

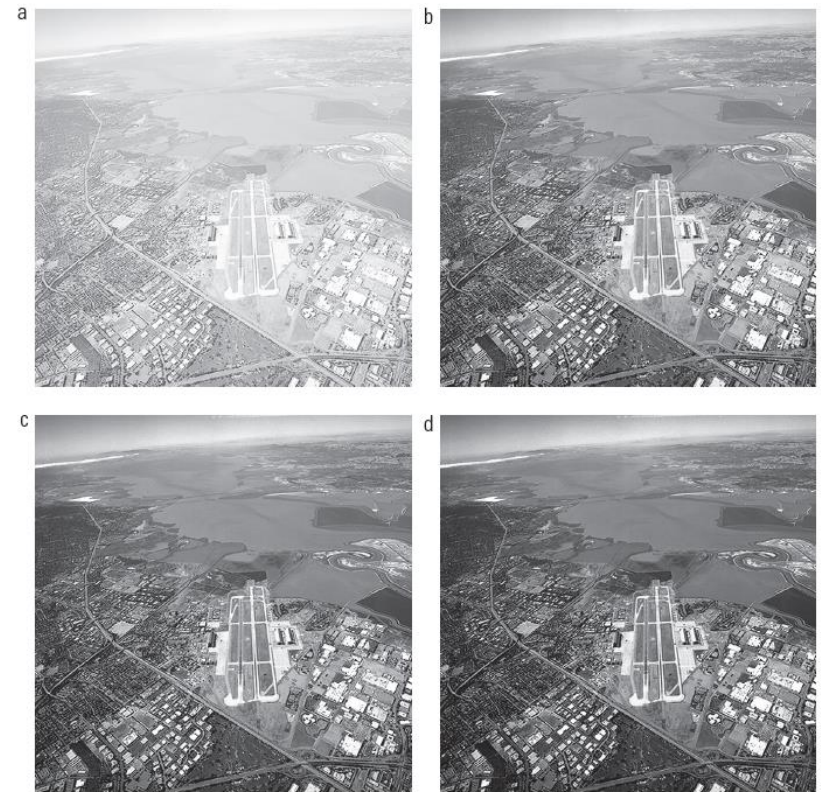
Pode ser ajustada através do parâmetro  $\gamma$

Usualmente utilizada para corrigir  
imagens de displays

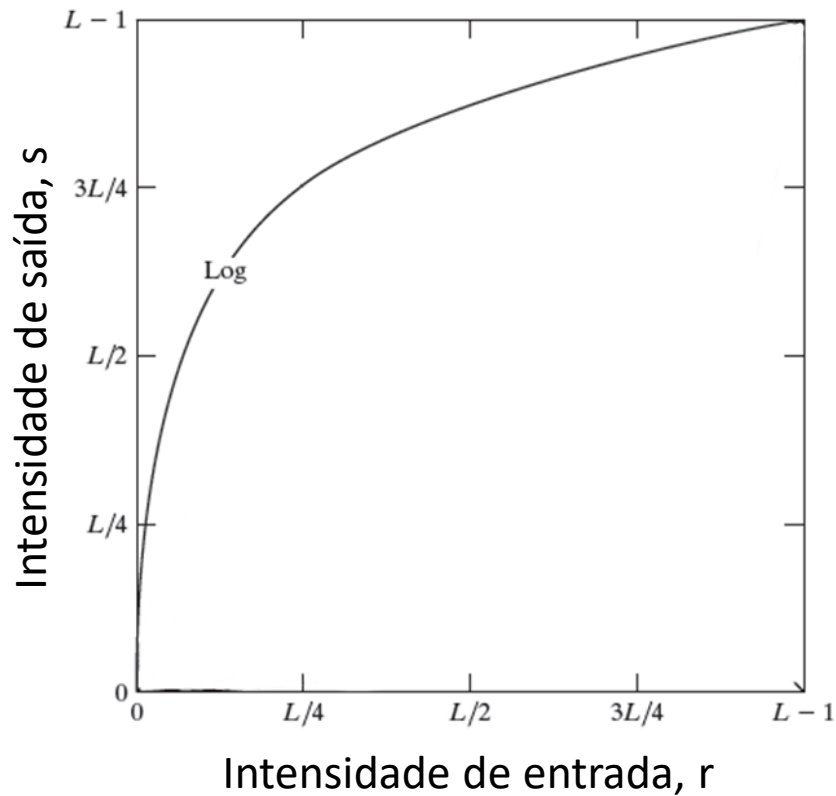
# Transformações de intensidade pontuais comumente utilizadas



## Transformação de potência



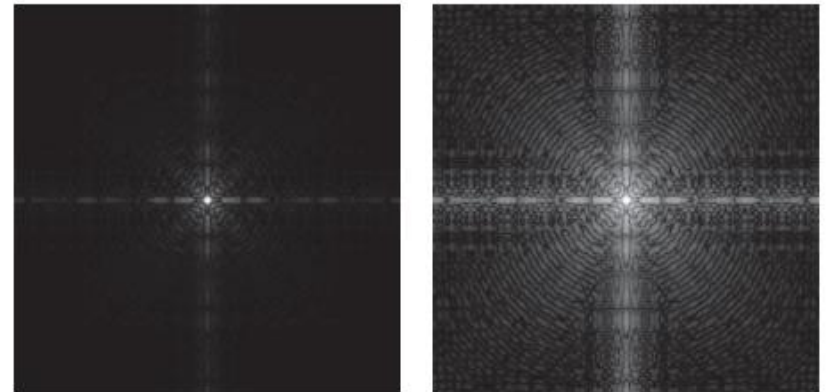
# Transformações de intensidade pontuais comumente utilizadas



Logaritmo da imagem:

$$s = (L - 1) \frac{\log(1 + r)}{\log(L)}$$

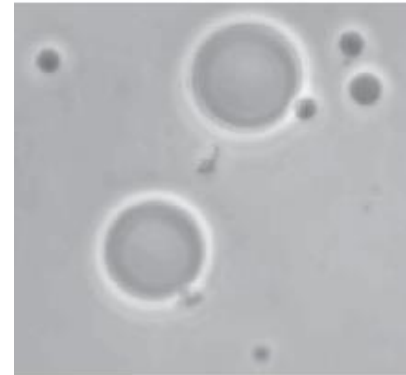
Comumente utilizada quando a imagem possui valores muito diferentes



# Transformações de intensidade pontuais comumente utilizadas

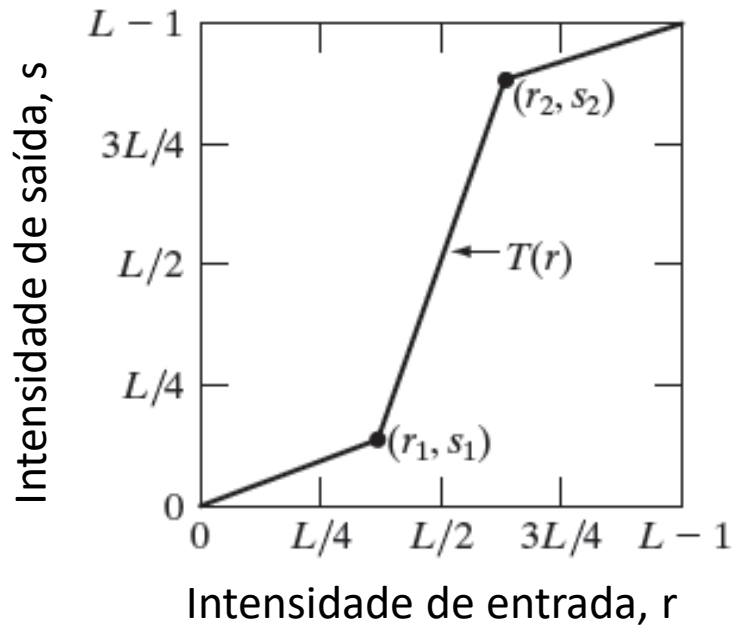


Limiarização



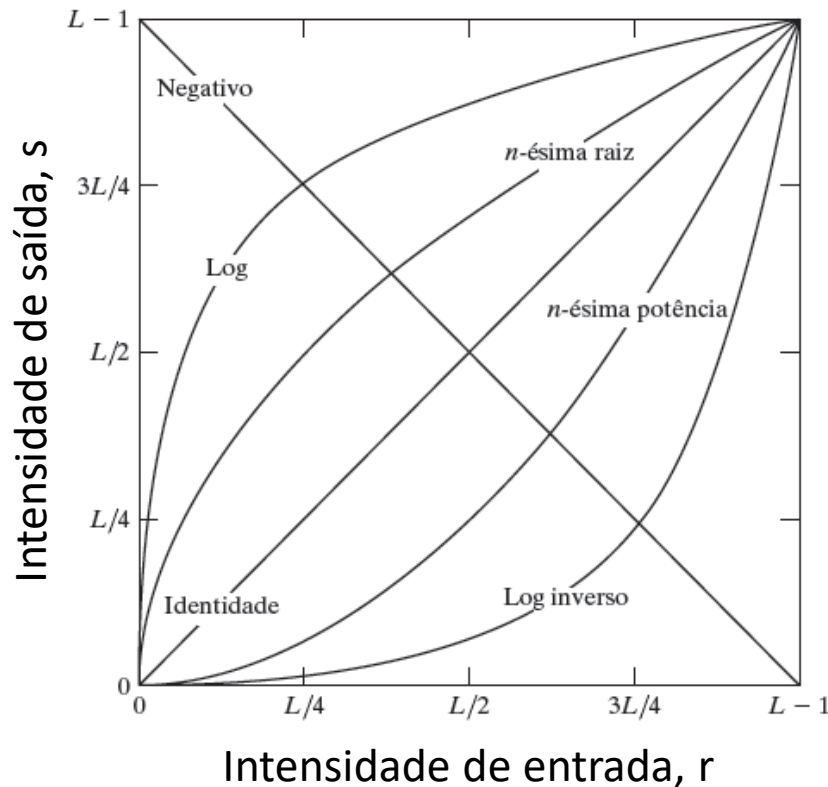
# Transformações de intensidade pontuais comumente utilizadas

- Transformação por partes lineares





# Transformações de intensidade pontuais comumente utilizadas



- Transformações de intensidade pontuais possuem implementação simples
- Elas também são bem eficientes

# Implementação da transformação de intensidade pontual

- Podemos implementar transformações de intensidade através de uma tabela de associação (comumente chamada de *lookup table*)
- Para uma imagem de 8 bits, a tabela possui 256 linhas
- A posição  $r$  da tabela possui o valor que o nível de cinza  $r$  irá adquirir após a transformação
- A implementação por *lookup table* possibilita calcular de forma muito eficiente a imagem transformada
- Em muitos casos, é possível transformar a imagem de forma interativa

$s$	0	5	13	30	53	78	90	...	255
$r$	0	1	2	3	4	5	6	...	255

# Linguagem Python

# Linguagem Python - Instalação

Obtida em

<https://www.python.org/downloads/>

A grande vantagem de utilizar Python para processamento de imagens está na enorme variedade de bibliotecas desenvolvidas na área.

A forma padrão de instalar pacotes em Python é digitar no terminal:

```
pip install <nome do pacote>
```

Entretanto, a instalação de pacotes (bibliotecas) em Python pode apresentar problemas em algumas situações, especialmente se o pacote possuir funções compiladas em C.

# Linguagem Python - Instalação

Bibliotecas Python utilizadas na disciplina:

numpy, scipy, matplotlib, notebook (jupyter)

Forma simples de instalar todas as bibliotecas:

Baixar e instalar o gerenciador conda:

<https://docs.conda.io/en/latest/miniconda.html>

No terminal:

```
conda install -c conda-forge numpy scipy matplotlib notebook
```

\*O Google Colab já possui as bibliotecas que serão utilizadas na disciplina

# Linguagem Python

Algumas IDEs conhecidas de Python:

- VSCode
- PyCharm
- Jupyter notebook ou JupyterLab

Utilizaremos o Jupyter notebook. Para inicializar o notebook, basta digitar no terminal:

```
>>> jupyter notebook
```

# Linguagem Python

Um bom “crash course” contendo diversos comandos Python, Numpy e Matplotlib:

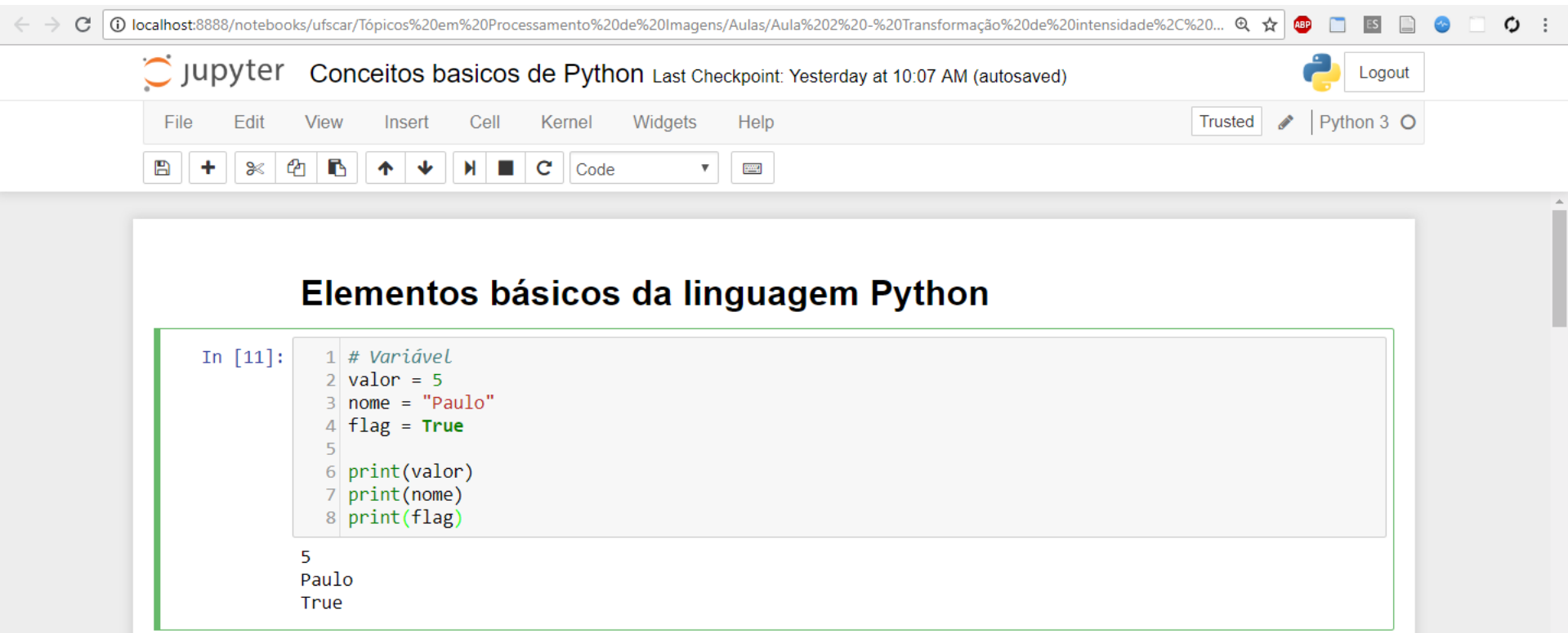
<https://cs231n.github.io/python-numpy-tutorial/>

# Iniciando o Jupyter

The screenshot shows the Jupyter web interface in a browser window. The address bar indicates the URL is `localhost:8888/tree`. The interface includes a top navigation bar with the Jupyter logo, a 'Quit' button, and a 'Logout' button. Below this, there are tabs for 'Files', 'Running', and 'Clusters'. The 'Files' tab is active, displaying a message: 'Select items to perform actions on them.' Below this message is a file browser interface showing a directory structure with a file named '0' and a folder named '/'. A 'Name' dropdown menu is open, showing options for creating new items: 'Notebook: Python 3' (highlighted), 'Other: Text File', 'Folder', and 'Terminal'. The 'Upload' button is also visible. The status bar at the bottom shows `localhost:8888/tree#`.



# Jupyter



The screenshot shows a Jupyter Notebook interface in a web browser. The browser's address bar displays the URL: `localhost:8888/notebooks/ufscar/Tópicos%20em%20Processamento%20de%20Imagens/Aulas/Aula%20-%20Transformação%20de%20intensidade%2C%20...`. The Jupyter logo and the notebook title "Conceitos basicos de Python" are visible, along with a "Last Checkpoint: Yesterday at 10:07 AM (autosaved)" message. A "Logout" button is in the top right. The menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the menu is a toolbar with icons for saving, adding cells, undo, redo, and running code. The "Code" dropdown menu is open. The main content area displays the title "Elementos básicos da linguagem Python" and a code cell. The code cell is labeled "In [11]:" and contains the following Python code:

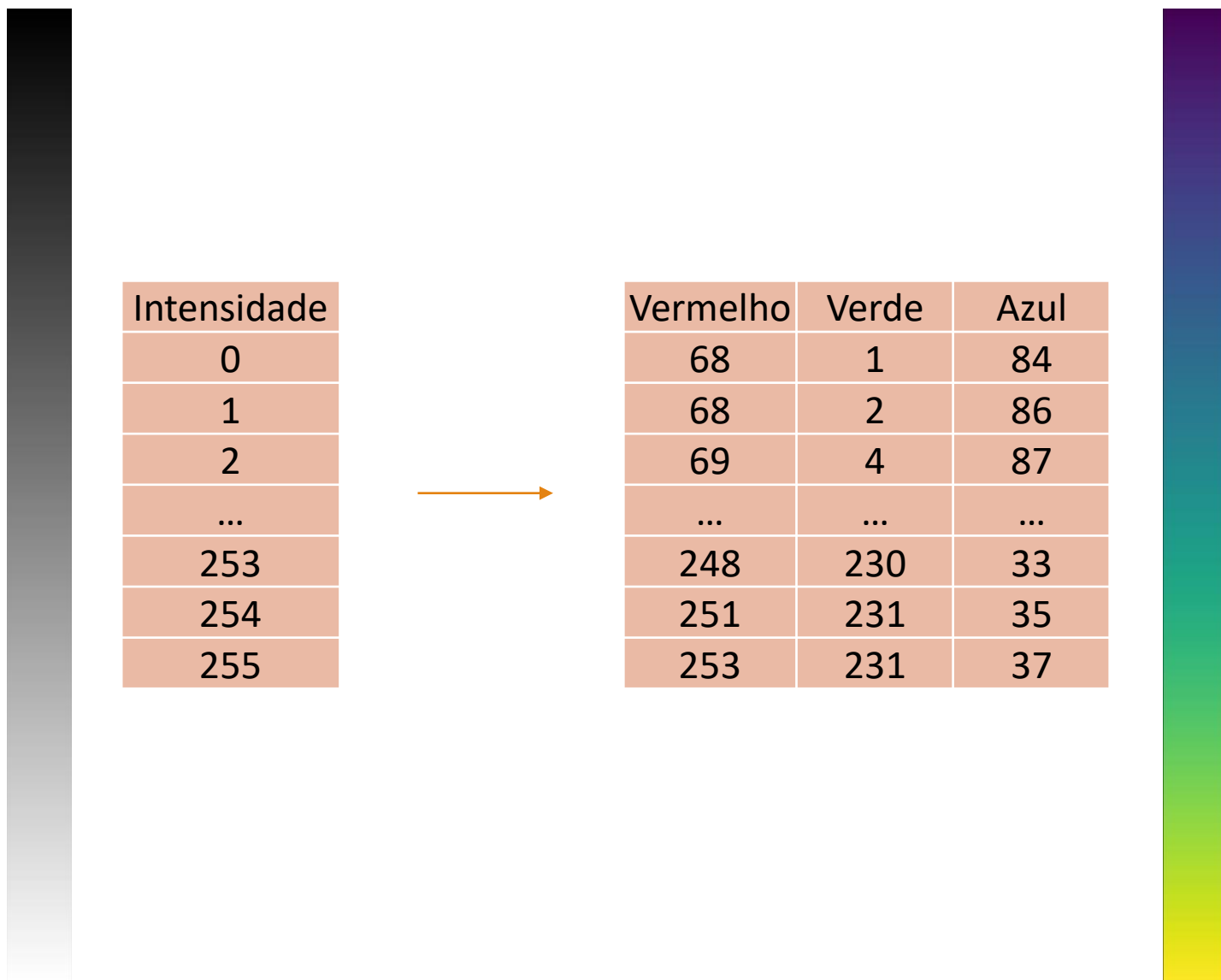
```
1 # Variável
2 valor = 5
3 nome = "Paulo"
4 flag = True
5
6 print(valor)
7 print(nome)
8 print(flag)
```

The output of the code cell is displayed below the code:

```
5
Paulo
True
```

shift+Enter para executar o código em uma célula

## Detalhe: como o matplotlib mapeia as cores das imagens na função imshow



# Aplicação dos conceitos

Notebooks “**numpy e matplotlib**”, “**Leitura de imagem**” e  
“**Transformacoes pontuais em imagens**”

# Histograma de imagens

# Histograma

- O histograma de uma imagem possui informações sobre os níveis de intensidade presentes na imagem. A informação sobre a localização espacial dessas intensidades é perdida.
- Muita informação sobre a imagem é perdida na representação por histograma. Ainda assim, histogramas permitem a análise de diversas técnicas de processamento de imagem de forma intuitiva.

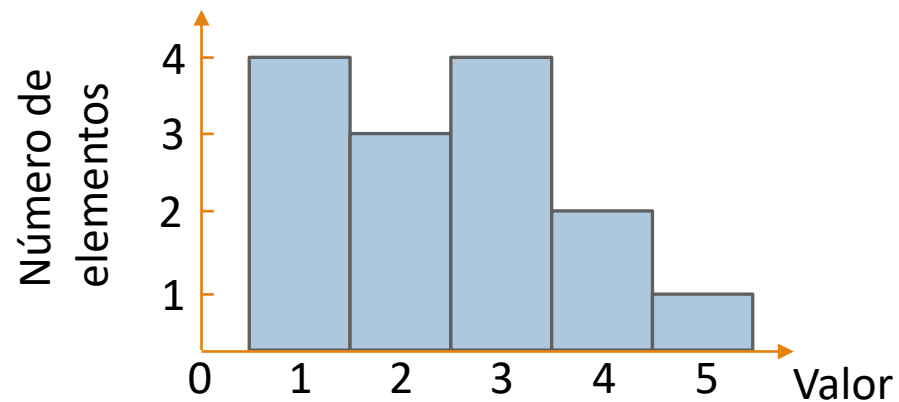
# Histograma

- O histograma de uma imagem possui informações sobre os níveis de intensidade presentes na imagem. A informação sobre a localização espacial dessas intensidades é perdida.
- Muita informação sobre a imagem é perdida na representação por histograma. Ainda assim, histogramas permitem a análise de diversas técnicas de processamento de imagem de forma intuitiva.
- Vamos assumir que as imagens possuem intensidades no intervalo  $[0, L-1]$ .
- Um histograma é a informação sobre quantos pixels na imagem possuem o valor de intensidade  $k$ , para todos os valores de intensidade (todos os valores de  $k$ ).

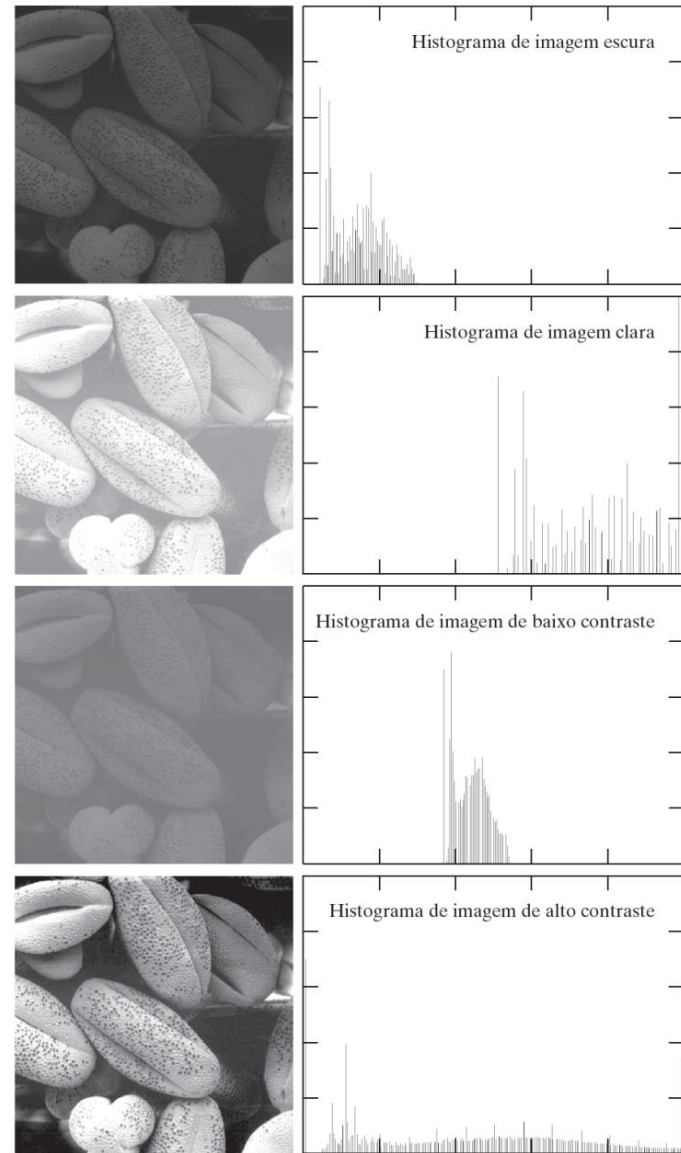
# Histograma

Exemplo para um sinal 1D (ou uma imagem de uma única linha).

3	1	1	2	3	4	5	2	1	3	4	2	3	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---



# Exemplos de histogramas



Intensidade ( $r_k$ )

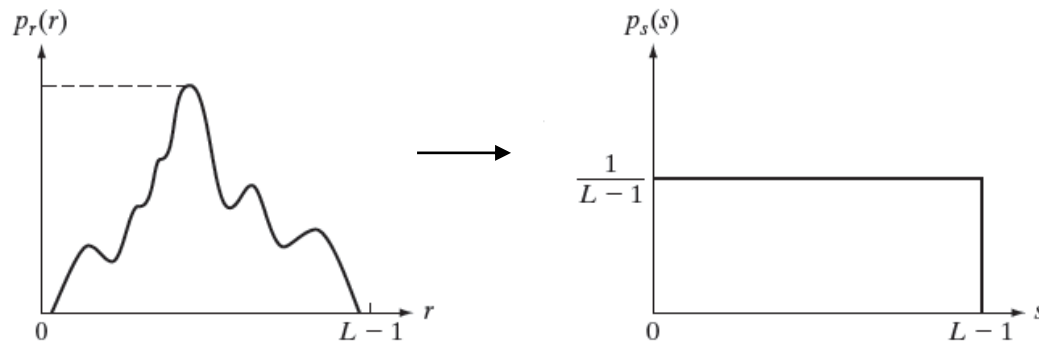


# Histograma

Notebook “**Calculo de histograma**”

# Equalização de histograma

- Metodologia utilizada para tornar o histograma uniforme (“flat”). Isto é, fazer com que todos os níveis de intensidade estejam presentes em igual número.



- Este método aumenta o contraste da imagem e, em alguns casos, melhora a aparência da imagem.

# Equalização de histograma

Vamos definir  $n_k$  como sendo o número de pixels possuindo o  $k$ -ésimo nível de intensidade, e que a imagem possui  $M$  linhas e  $N$  colunas.

Usualmente:

- $n_0$ : número de pixels possuindo intensidade 0.
- $n_1$ : número de pixels possuindo intensidade 1.
- ...
- $n_{255}$ : número de pixels possuindo intensidade 255.

É possível demonstrar que se utilizarmos a seguinte transformação

$$s_k = T(k) = \frac{(L - 1)}{MN} \sum_{j=0}^k n_j$$

A imagem resultante terá um histograma aproximadamente uniforme!


# Equalização de histograma - Exemplo

Vamos supor que temos uma imagem de 3 bits de profundidade (8 níveis de intensidade) de tamanho  $64 \times 64$  pixels ( $MN=4096$ ). Essa imagem possui os seguintes números de pixels com cada intensidade  $k$ :

$k$	$n_k$
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

# Equalização de histograma - Exemplo

0.0017


$$s_k = T(k) = \frac{(L-1)}{MN} \sum_{j=0}^k n_j$$

$k$	$n_k$
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

# Equalização de histograma - Exemplo

0.0017



$$s_k = T(k) = \frac{(L-1)}{MN} \sum_{j=0}^k n_j$$

$$s_0 = T(0) = 0.0017 \sum_{j=0}^0 n_j = 0.0017 * n_0 = 1.34$$

$$s_1 = T(1) = 0.0017 \sum_{j=0}^1 n_j = 0.0017 * (n_0 + n_1) = 3.08$$

$k$	$n_k$
0	790
1	1023
2	850
3	656
4	329
5	245
6	122
7	81

# Equalização de histograma - Exemplo

Como queremos que a imagem final possua a mesma profundidade que a original, precisamos arredondar o resultado para o inteiro mais próximo:

$$s_0 = 1.33 \rightarrow 1$$

$$s_4 = 6.23 \rightarrow 6$$

$$s_1 = 3.08 \rightarrow 3$$

$$s_5 = 6.65 \rightarrow 7$$

$$s_2 = 4.55 \rightarrow 5$$

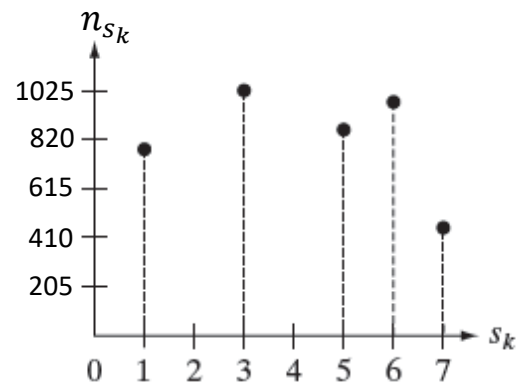
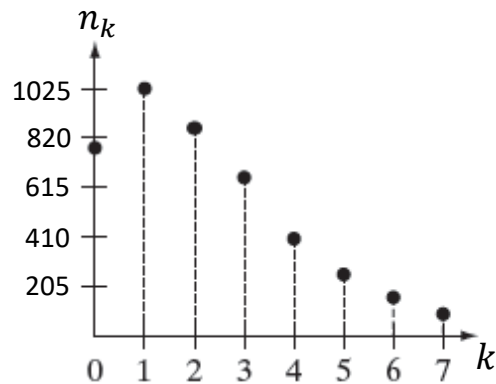
$$s_6 = 6.86 \rightarrow 7$$

$$s_3 = 5.67 \rightarrow 6$$

$$s_7 = 7.00 \rightarrow 7$$

# Equalização de histograma - Exemplo

Resultado:

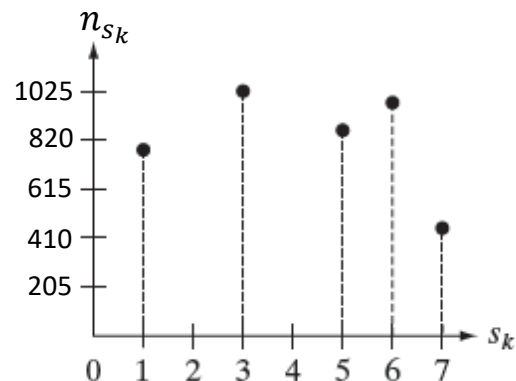
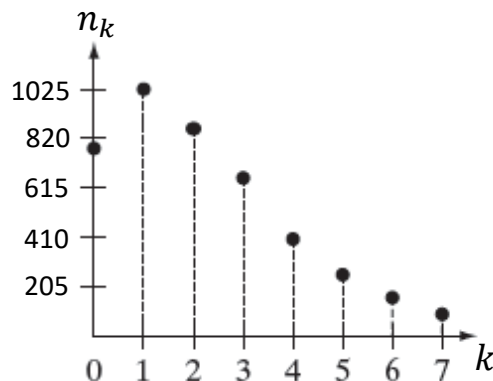




# Equalização de histograma

$$s_k = T(k) = \frac{(L-1)}{MN} \sum_{j=0}^k n_j$$

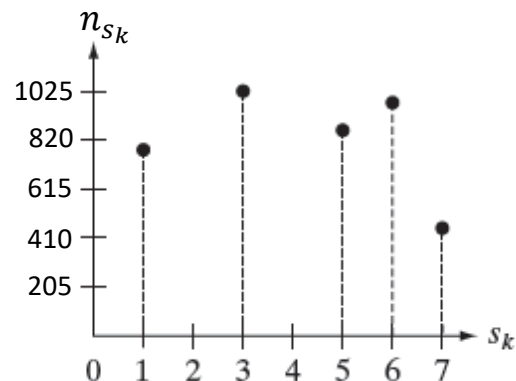
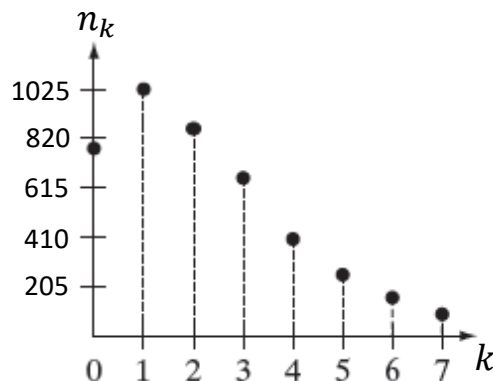
- Alguns problemas:
  - Dificilmente obtemos um histograma perfeitamente uniforme.



# Equalização de histograma

$$s_k = T(k) = \frac{(L-1)}{MN} \sum_{j=0}^k n_j$$

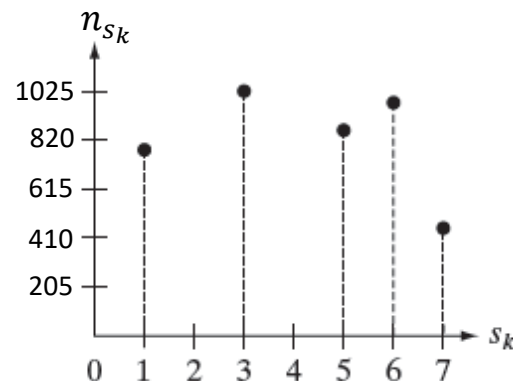
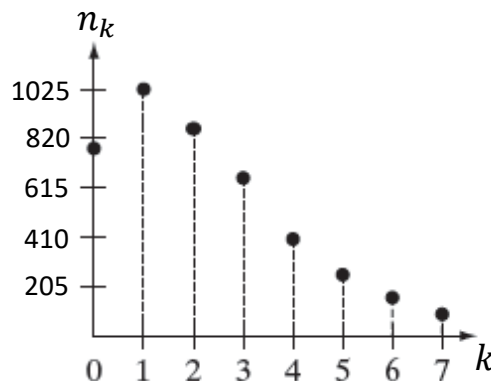
- Alguns problemas:
  - Dificilmente obtemos um histograma perfeitamente uniforme.
  - Usualmente, há perda de informação



# Equalização de histograma

$$s_k = T(k) = \frac{(L-1)}{MN} \sum_{j=0}^k n_j$$

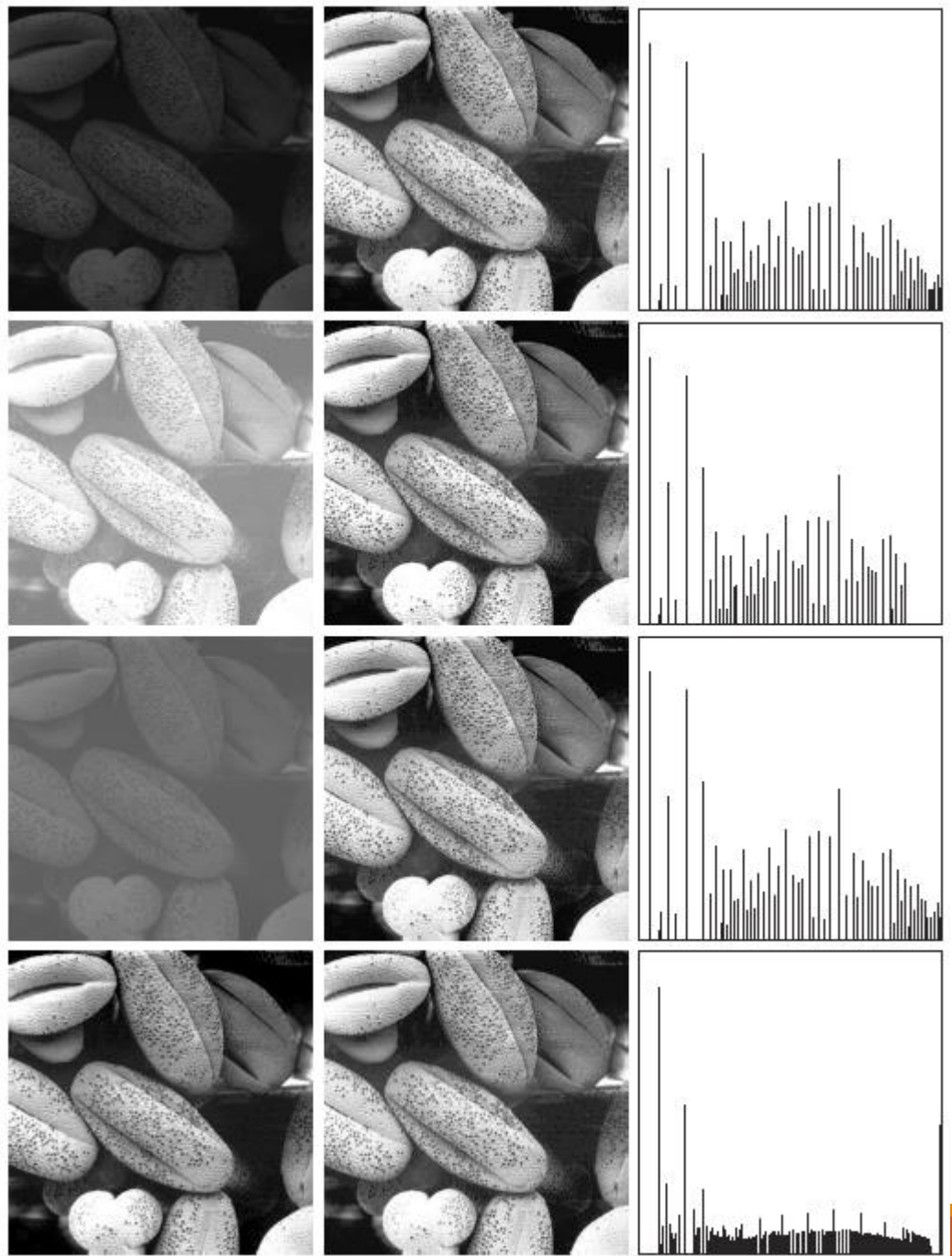
- Alguns problemas:
  - Dificilmente obtemos um histograma perfeitamente uniforme.
  - Usualmente, há perda de informação
  - Resultados ruins costumam ser obtidos para imagens com poucos níveis de intensidade



# Equalização de histograma

- “Vantagem” da equalização de histograma: não há parâmetros para ajustar.
- Se o objetivo for somente visualizar regiões da imagem com baixo contraste, equalização de histograma tende a funcionar bem.

# Exemplos de equalização de histograma



# Equalização de histograma

Notebook “**Equalização de histograma**”