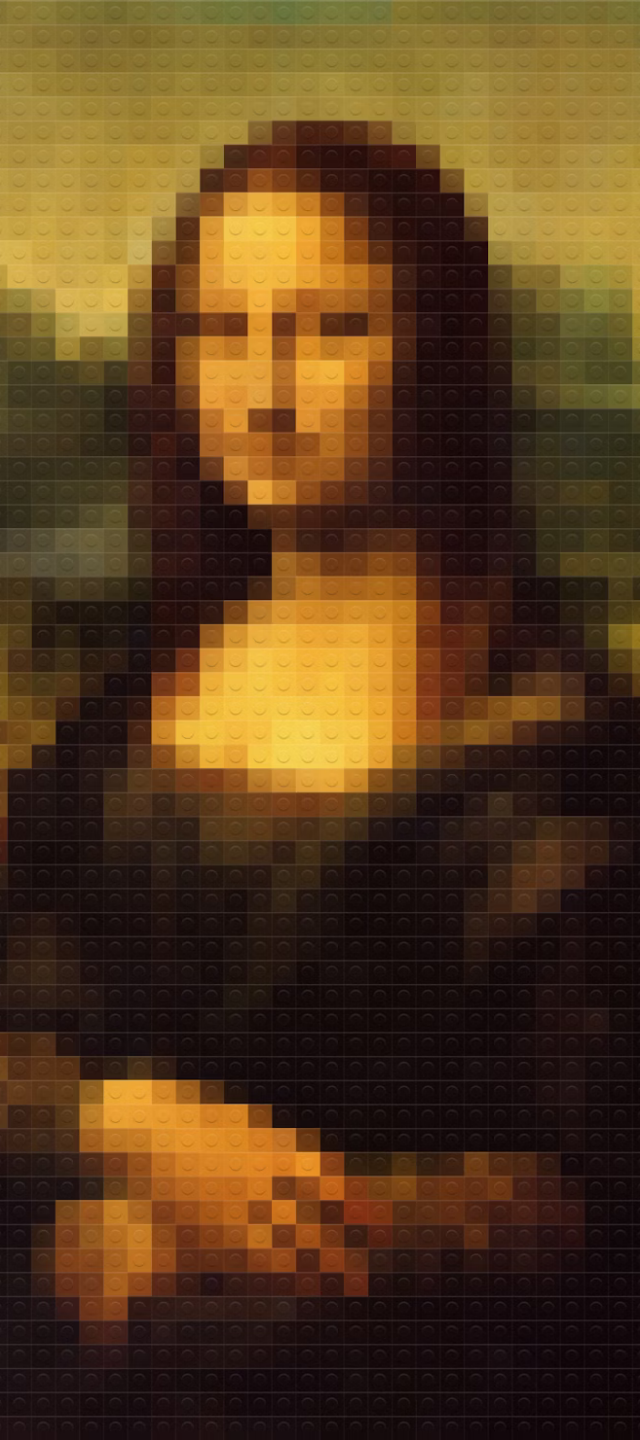




# Transformações de Intensidade Ponto-a-Ponto

---

## Visão Computacional

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

# Objetivos de Aprendizagem

---

- **Compreender** os fundamentos matemáticos das transformações de intensidade ponto-a-ponto.
- **Identificar** as principais funções radiométricas (linear, gama, logarítmica) e suas aplicações em contextos reais.
- **Implementar** algoritmos de transformação em código (Python).
- **Avaliar** o impacto das transformações sobre a distribuição de intensidades usando histogramas e visualizações.

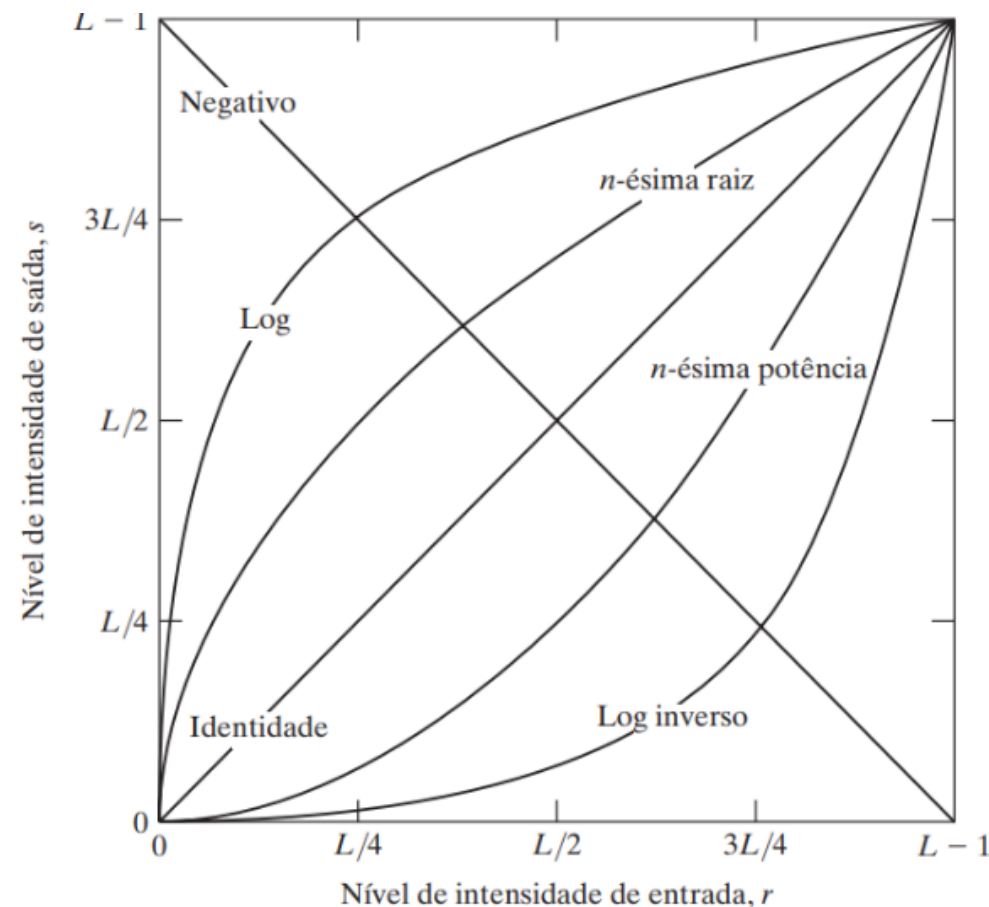


# Transformações no Domínio Espacial

- **Objetivo**: modificar a distribuição dos níveis de cinza ou cor de uma imagem para realçar características, ajustar contraste ou preparar dados para etapas posteriores.
- **Espacial**: Modificação **direta** nos pixels no plano da imagem.
- **Princípio**: cada pixel  $f(x, y)$  é convertido em um novo valor  $g(x, y) = T(f(x, y))$  através de uma função de transformação  $T(\cdot)$ .
- **Transformações Radiométricas**: Operam apenas sobre os valores de intensidade dos pixels.
  - **Não alteram a localização espacial dos pixels**.
- **Filtragem Espacial**: Envolve a combinação de um pixel com seus vizinhos por meio de operadores convolucionais ou morfológicos.
  - **Alteram o conteúdo espacial**.

# Transformações Radiométricas

- **Funções radiométricas** mais básicas aplicadas na transformação de intensidade
- Utilizadas para o **realce** de imagens.
- **Tipos de Transformações**
  - **Lineares** (ex.: contraste linear, subtração, soma)
  - **Não lineares** (ex.: logarítmica, gamma, equalização de histograma)
  - **Baseadas em estatísticas locais** (ex.: normalização por janela, stretching adaptativo)
- **Realce**: geralmente para minimizar na imagem efeitos de ruídos, perda de contraste, borramento e distorções.



Funções de transformação de intensidade. Fonte: [COVAP-UTFPR](#).

# Negativo de uma Imagem

---

O negativo **inverte a escala de brilho**: pixels claros tornam-se escuros e vice-versa.

- **Formulação Matemática:**  $g(x, y) = L_{\max} - f(x, y)$ ,
  - $f(x, y)$  é a intensidade original
  - $g(x, y)$  o negativo
  - $L_{\max}$  o nível máximo possível (e.g., 255 para 8-bits).
- **Histórico:** Na fotografia analógica, o papel fotográfico reproduzia um negativo que era então inverso do filme; a inversão era necessária para produzir uma imagem positiva em exibição.
- **Aplicações:** Detecção de **falhas** em processos industriais (ex.: inspeção de circuitos onde componentes brilhantes podem mascarar defeitos). Facilita a **identificação de detalhes** em regiões de alta luminosidade que, na forma original, são saturadas.



# Negativo de uma Imagem: Exemplo Numérico e Visual

- Em 8-bits:  $f(100) = 255 - 100 = 155$ ,  $f(250) = 255 - 250 = 5$
- Pixels muito claros tornam-se escuros; pixels escuros tornam-se quase brancos.



Isadora Renascentista: original em escala de cinza (à esquerda) e negativo (à direita).

# Transformação Logarítmica

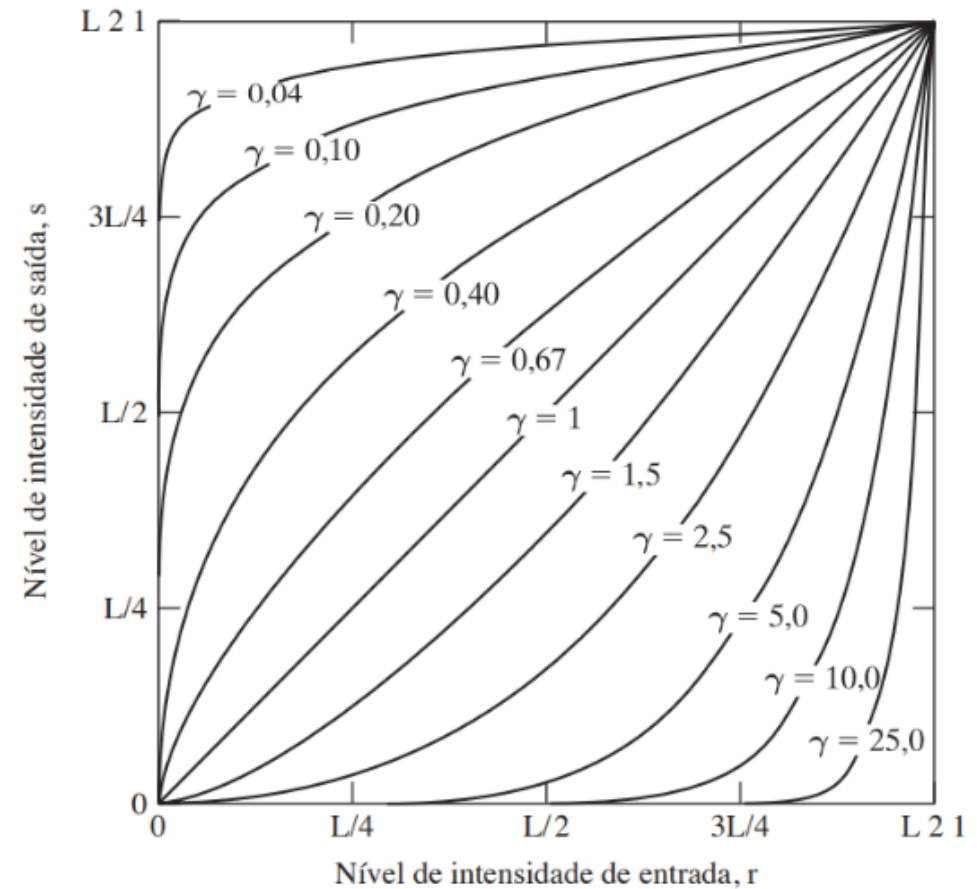
---

A função logarítmica comprime valores altos e expande baixos, **aumentando contraste** em trechos escuros.

- **Formulação Matemática:**  $g(x, y) = c \log(1 + f(x, y))$ 
  - $c$  escala a saída para o intervalo desejado.
- Se  $f(x, y) = 200$  (8-bits), com  $c = 1$ :  $g \approx \log(201) \approx 5.3$ .
  - Escalando por  $255/5.3 \approx 48$ , obtemos  $\sim 254$ , preservando brilho máximo.
- **Aplicação:** reduzir diferenças de brilho entre sombras e highlights.
- **Compressão dinâmica:** valores altos de  $f$  são comprimidos, reduzindo saturação em highlights.
- **Amplificação de sombras:** valores baixos de  $f$  são expandidos, revelando detalhes em regiões escuras.

# Transformação de Potência (Gama)

- Ajusta a relação linear entre intensidade de entrada e saída.
- **Formulação Matemática:**  
 $g(x, y) = c f(x, y)^\gamma$ 
  - $\gamma < 1$ : realça trechos escuros
  - $\gamma > 1$ : realça trechos claros.
- Para garantir que o valor máximo não ultrapasse  $L_{\max}$ :  $c = \frac{L_{\max}}{(L_{\max})^\gamma}$
- Exemplo: para **8-bits**,  $L_{\max} = 255$  e  $\gamma = 0.4$ , temos  $c = 255/255^{0.4} \approx 1.5$ .
- Monitores CRT e LCD têm resposta não linear; a **correção de gamma** ( $\approx 2.2$ ) torna a imagem perceptualmente linear.



Funções de transformação de intensidade. Fonte: [COVAP-UTFPR](#).

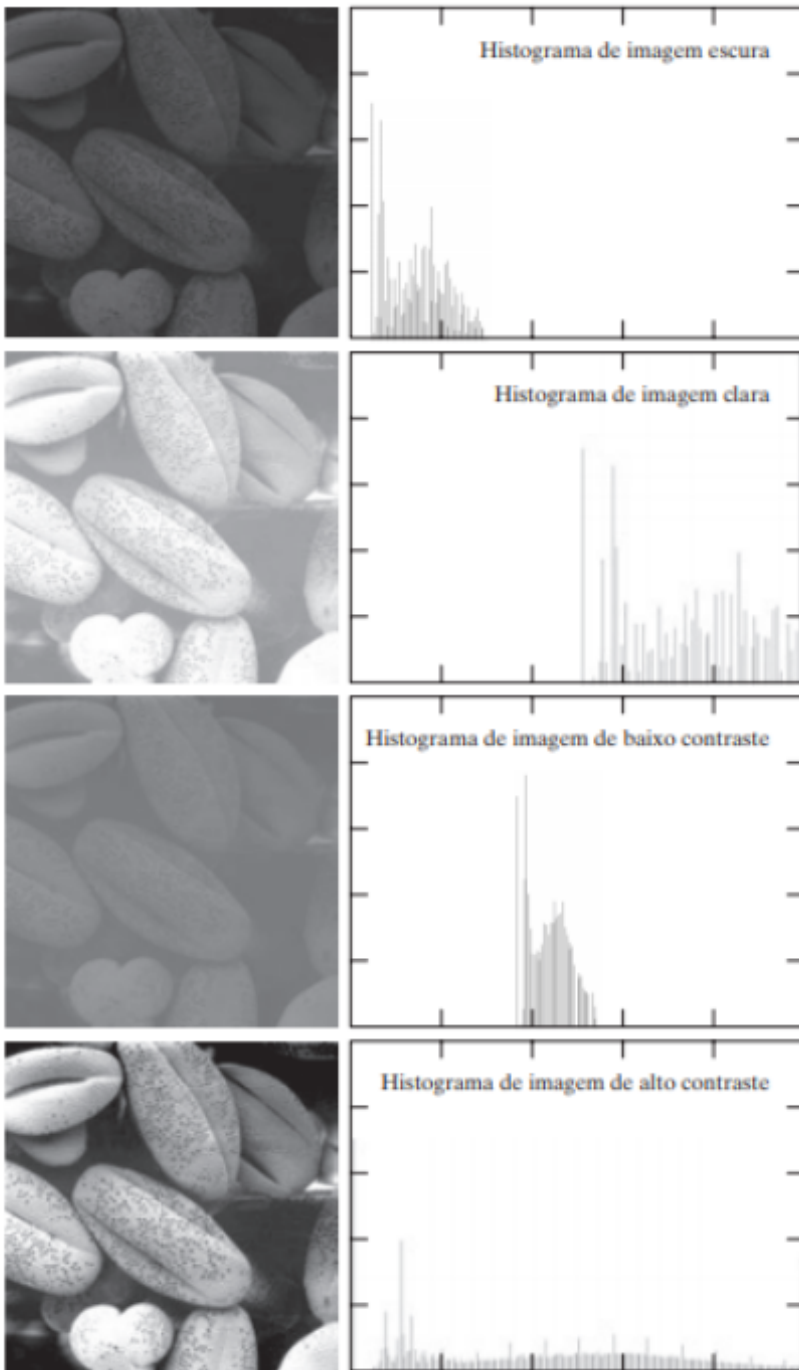


# Exemplo de Transformação de Potência (Gama)

Perceived (linear) brightness = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0  
Physical (linear) brightness = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0



Comparação de iluminação sem/com correção gama. Fonte das Imagens: [Learn OpenGL](https://learnopengl.com/).



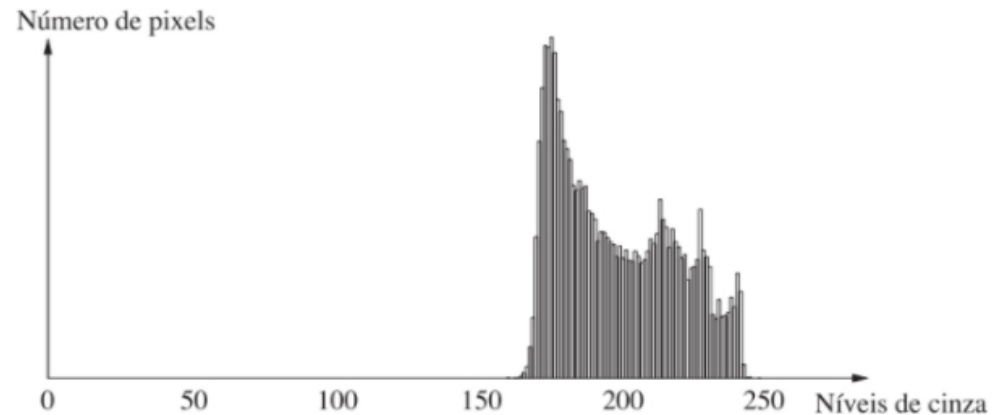
# Equalização de Histograma

- **Histograma:** distribuição de frequências  $p_f(r_k)$ .
- **Transf. cumulativa:**  $T(r_k) = (L_{\max}) \sum_{i=0}^k p_f(r_i)$
- **Saída:**  $g(x, y) = T(f(x, y))$ .
- **Algoritmo:**
  - Passo 1: Calcular histograma normalizado  $p_f$ .
  - Passo 2: Computar transformador cumulativo  $T(r_k)$ .
  - Passo 3: Aplicar  $g(x, y) = T(f(x, y))$ .
- **Resultado:** Imagem com contraste uniforme, adequado para imagens de baixa iluminação ou quando o histograma está concentrado em poucos níveis.
- Na imagem (à esquerda): Histogramas de uma imagem com grãos de pólen. De cima para baixo: escura, clara, baixo contraste e alto contraste. Fonte: [COVAP-UTFPR](#).

# Equalização de Histograma (cont.)



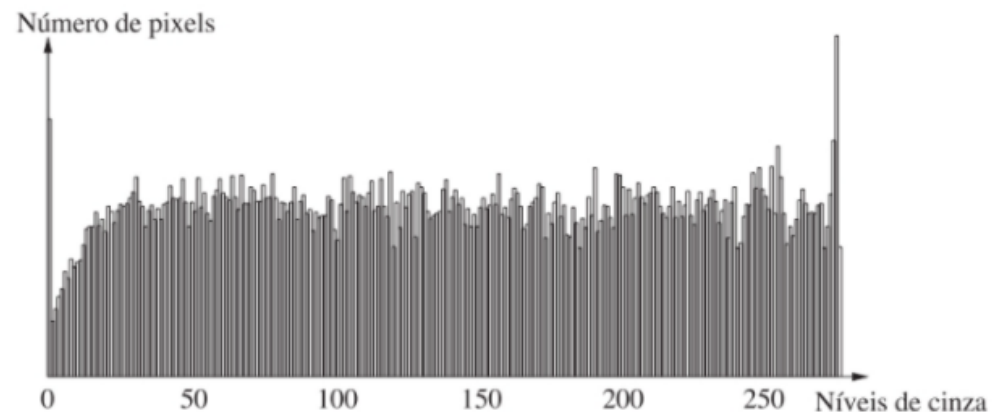
(a)



(b)



(c)

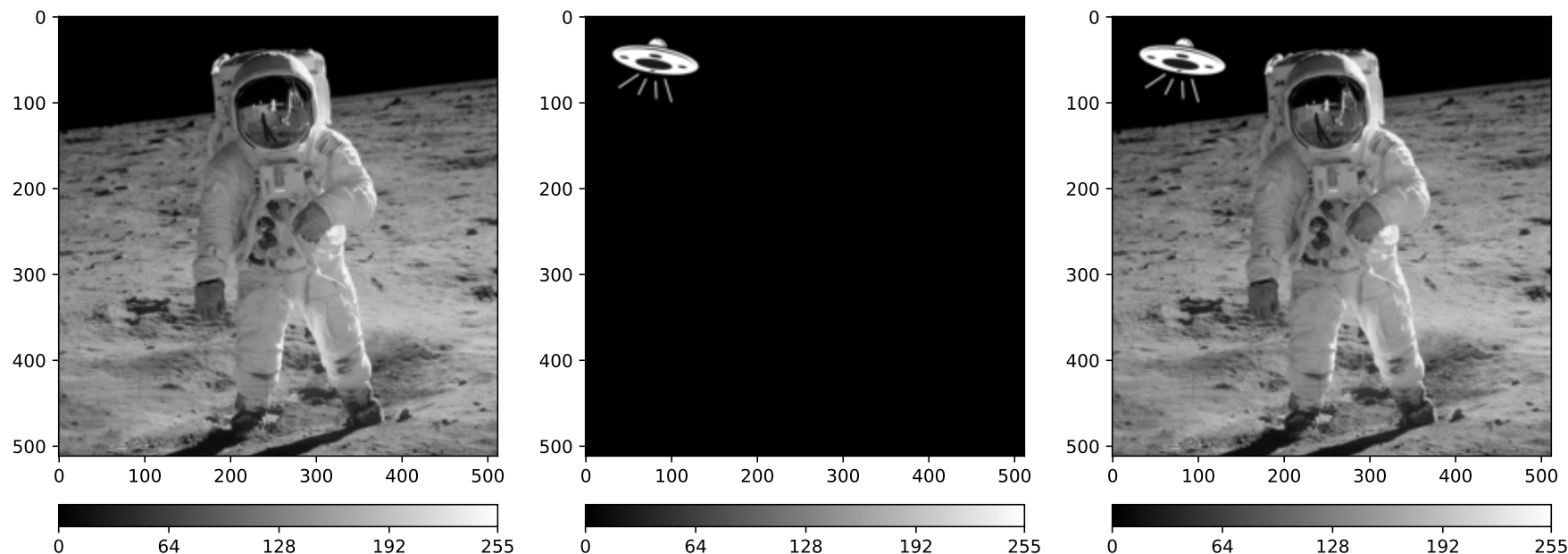


(d)

Resultado da equalização de histograma. Fonte: [COVAP-UTFPR](#).

# Extra: Operações Aritméticas em Imagens

- **Adição:**  $\forall x, y, \quad g(x, y) = f_1(x, y) + f_2(x, y)$ , onde  $f_1, f_2$  são imagens de mesma dimensionalidade.
- Valores resultantes podem exceder a faixa  $[0, L_{\max}]$ ; geralmente se aplica saturação:  
$$g(x, y) = \min(L_{\max}, f_1(x, y) + f_2(x, y))$$

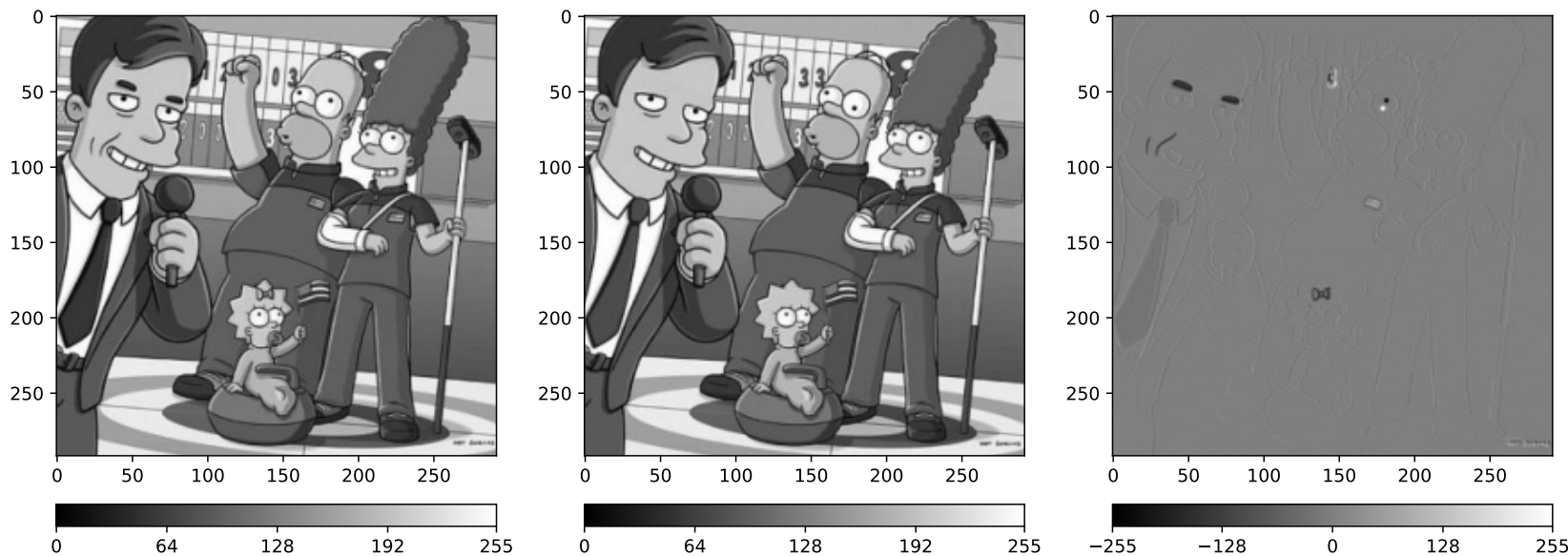


Adição de duas imagens. Fonte: [Basics of Image Processing – Vincent Mazet \(Université de Strasbourg\)](#).



# Extra: Operações Aritméticas em Imagens (cont.)

- **Subtração:**  $\forall x, y, \quad g(x, y) = |f_1(x, y) - f_2(x, y)|$ , onde  $f_1, f_2$  são imagens de mesma dimensionalidade.



Subtração de duas imagens. Fonte: [Basics of Image Processing — Vincent Mazet \(Université de Strasbourg\)](#).

- **Uso típico:** detecção de movimento (subtração frame-a-frame), contraste diferencial em imagens médicas, remoção de fundo estático.
- **Normalização:** antes da subtração, alinhar as imagens e equalizar brilho para evitar artefatos de fase.

# Resumo e Próximos Passos

- **Ideia geral:**  $g(x, y) = T(f(x, y))$ , onde  $f$  é a imagem original e  $T(\cdot)$  um função de transformação.
- Transformações ponto-a-ponto são ferramentas fundamentais e simples que, quando aplicadas corretamente, **umentam a utilidade** das imagens em diversas áreas.
- São utilizadas como etapa **pré-processamento** antes de filtragem espacial, detecção de bordas ou reconhecimento de padrões
- Transformações mal escolhidas **podem introduzir artefatos** visuais ou distorcer cores.
- **Próximos passos:** Implementar transformações e analisar resultados (prática).



**Atividade recomendada:** Leitura dos capítulo 3.



# Perguntas e Discussão

---

1. Como a escolha da função  $T(\cdot)$ , e.g., linear, gama, logarítmica, **afeta a percepção visual** de contraste e detalhes finos na imagem resultante?
2. Em quais cenários a equalização de histograma pode **degradar a qualidade** de uma imagem em vez de melhorá-la?
3. Quais são as implicações computacionais de aplicar transformações ponto-a-ponto em tempo real versus offline? Quais os desafios de realizar essas transformações de **forma concorrente/paralela**?
4. Quais são os principais desafios ao subtrair duas imagens capturadas em **condições diferentes** (e.g., com iluminação variada)?
5. Como a representação em bits (8-bits vs 16-bits) **afeta a precisão** das operações aritméticas e a **preservação de informação**?