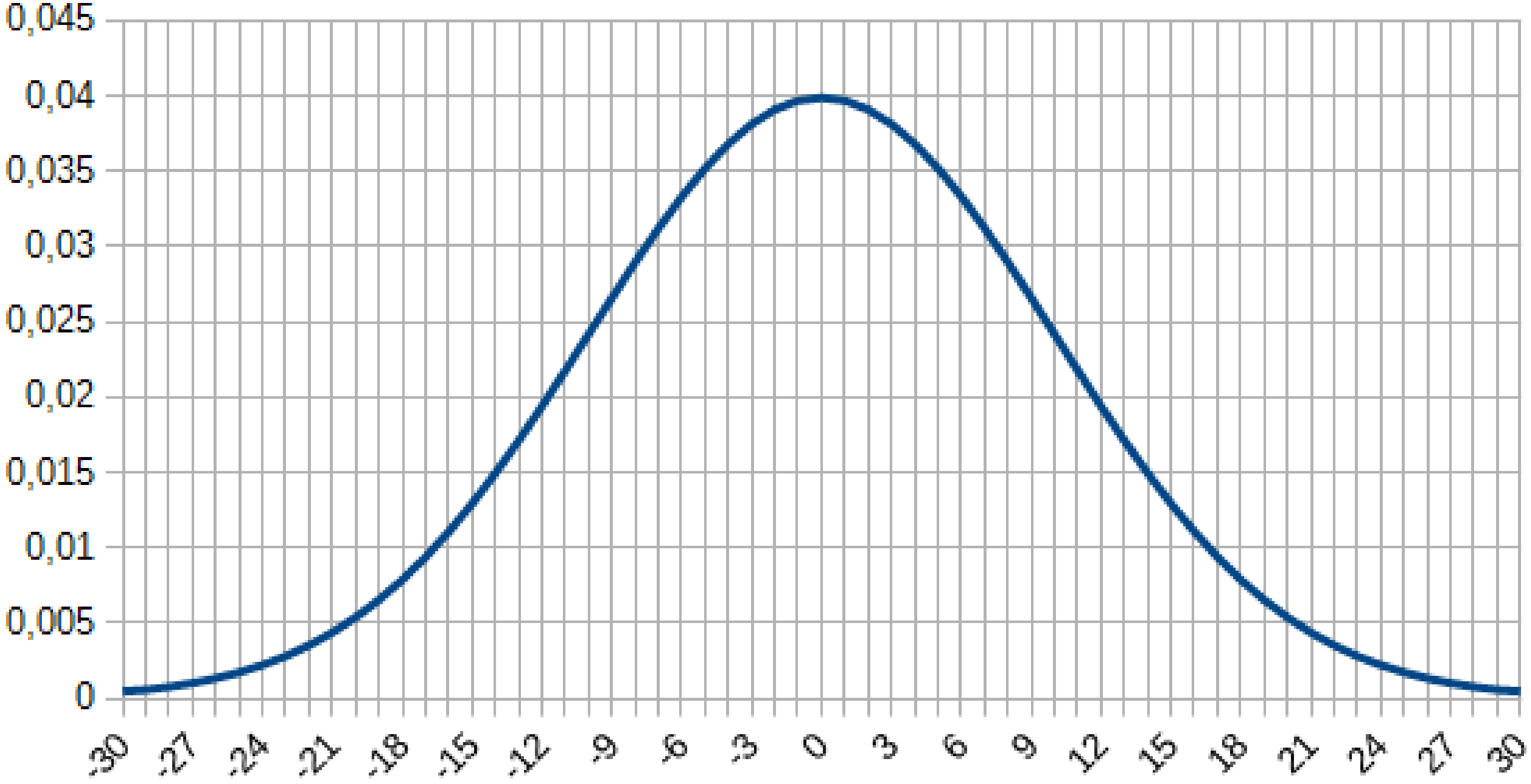
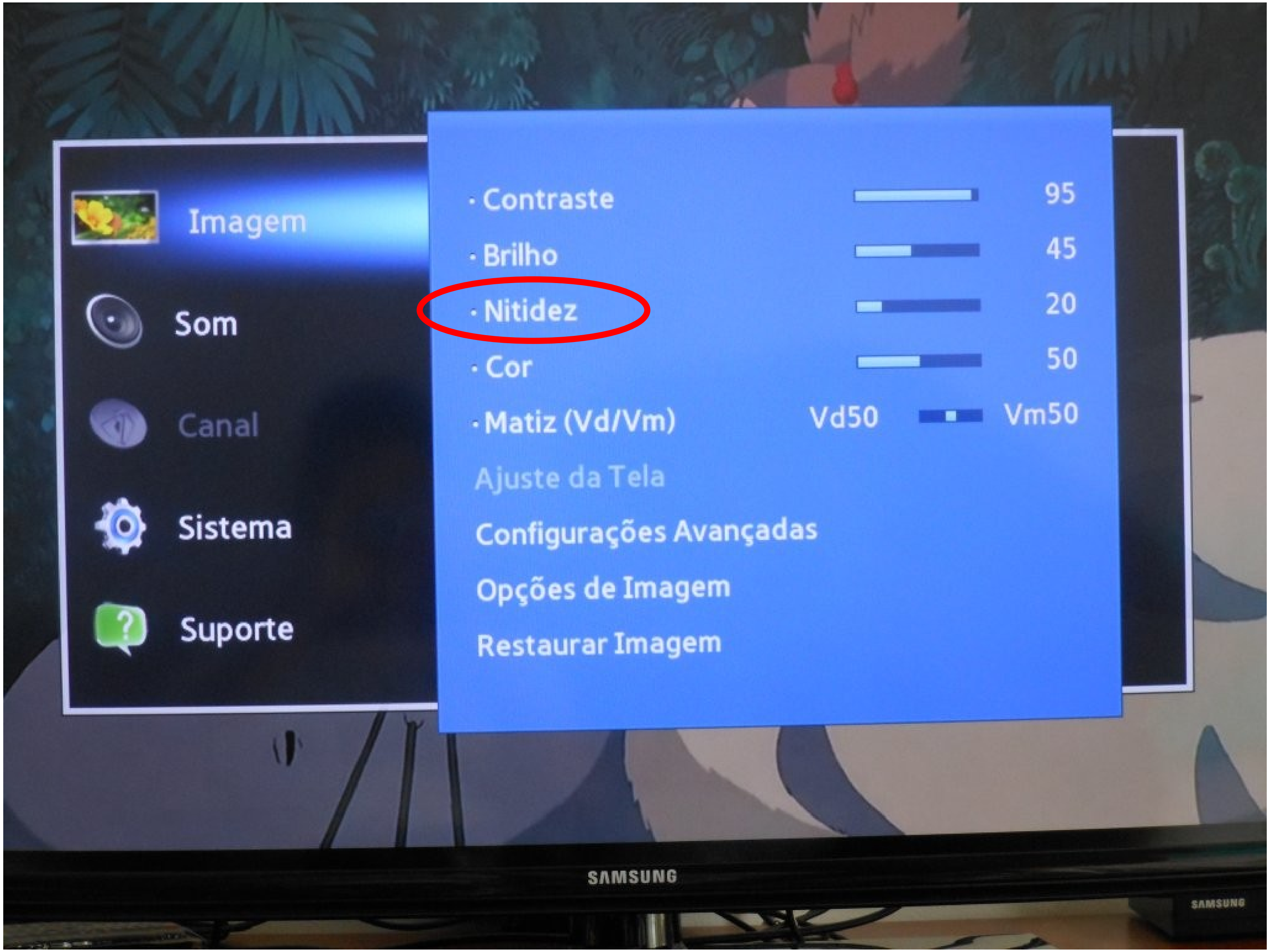
# Processamento Digital de Imagens

Prof. Bogdan Tomoyuki Nassu




UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

# Hoje




UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

# Sobre o controle de “nitidez”

- Varia de fabricante para fabricante, mas no geral:
  - Nitidez no centro = imagem original.
  - Nitidez à esquerda = imagem ligeiramente borrada.
  - Nitidez à direita = imagem com detalhes realçados.

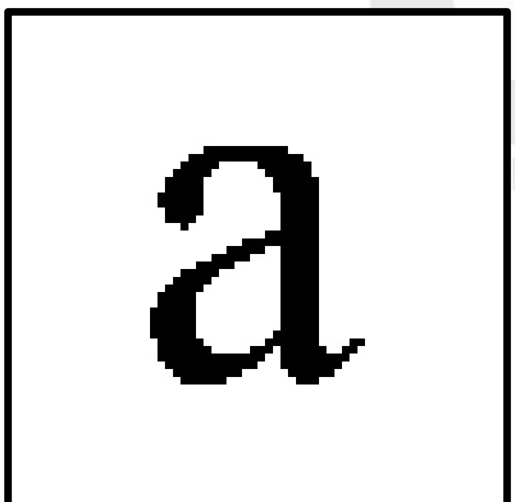
# Imagem borrada?

- Já vimos uma técnica que deixa a imagem borrada...
  - Qual?

# Imagem borrada?

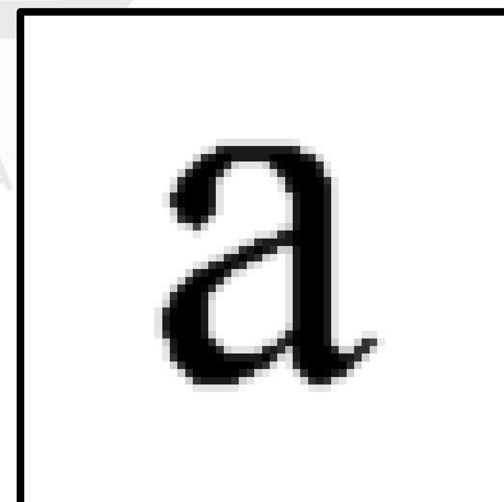
- Já vimos uma técnica que deixa a imagem borrada...
  - R: o filtro da média.
  - O controle de nitidez de um televisor necessita de resultados mais sutis do que aqueles produzidos pelo filtro da média...

Imagem com 64x64 pixels.



Após filtro da média 3x3.

Efeito mais sutil.



# Formalizando algumas coisas...

- Vimos vários filtros que são aplicados usando janelas deslizantes.
  - O valor do pixel central de cada janela é dado por alguma operação realizada com as intensidades dos pixels dentro da janela...
    - Média.
    - Mediana.
    - Mínimo.
    - Máximo.
- Este procedimento é chamado de *filtragem no domínio espacial*.

# Filtragem linear x não linear

- Os filtros espaciais são divididos em duas categorias:
  - Dadas as imagens *A*, *B*, e *C*, tal que $C(x,y) = A(x,y) + B(x,y)$.
  - Seja *I'* o resultado da filtragem da imagem *I*.
  - Um filtro é linear se $C'(x,y) = A'(x,y) + B'(x,y)$.
  - Do contrário, o filtro é não-linear.
- É muito (muito) mais fácil de entender se outra forma:

*Um filtro é linear se a saída para uma janela for igual à soma ponderada dos valores dentro da janela.*

- A partir desta descrição, descrevemos qualquer filtro linear como uma matriz de coeficientes, também chamada de *kernel*.
  - Nota 1: o pixel da saída normalmente fica no centro da janela, mas isso não é essencial!
  - Nota 2: para aplicar um filtro linear a uma imagem colorida, simplesmente filtramos cada canal independentemente.

# Exemplo: filtro da média

- O filtro da média é linear?

# Exemplo: filtro da média

- O filtro da média é linear?
  - R: sim!
  - O resultado para uma janela é igual à média simples dos valores dentro da janela!
- Como é o kernel para o filtro da média 3x3?

# Exemplo: filtro da média

- O filtro da média é linear?
  - R: sim!
  - O resultado para uma janela é igual à média simples dos valores dentro da janela!
- Como é o *kernel* para o filtro da média 3x3?

| | | |
|---|---|---|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

ou 1/9 ·

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 1 | 1 |

# Exemplo: filtro da média

- E quanto aos outros filtros que vimos?
  - Mediana?
  - Máximo local?
  - Mínimo local?

# Convolução

•Definição da Wikipedia:

*Convolução é um operador linear que, a partir de duas funções dadas, resulta numa terceira que mede a área subentendida pela superposição das mesmas em função do deslocamento existente entre elas.*

# Convolução

- Definição da Wikipedia:

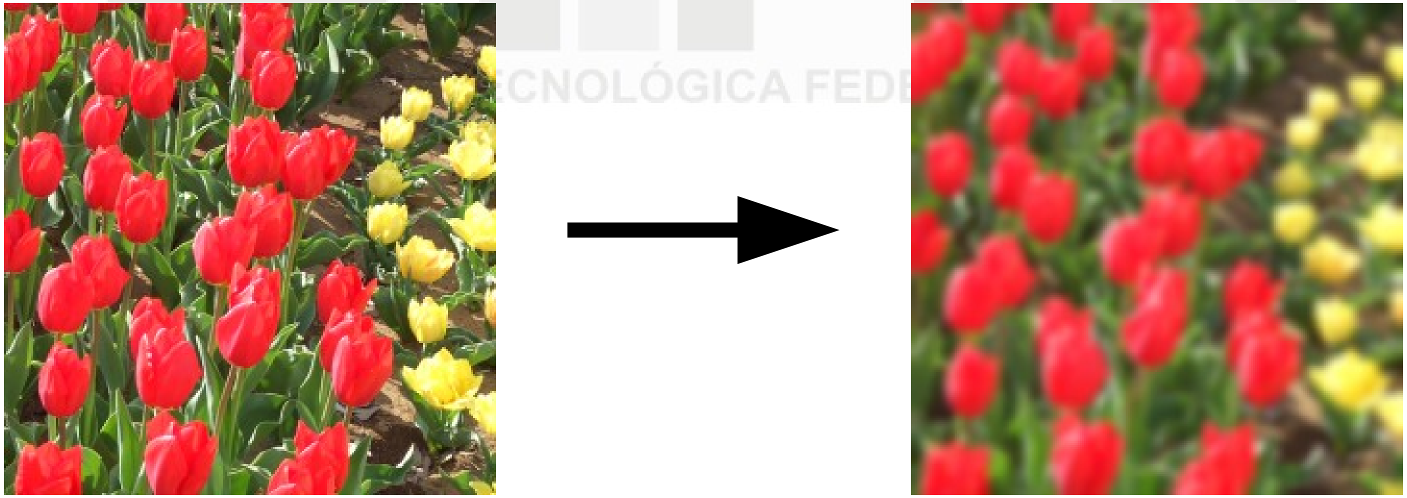
*Convolução é um operador linear que, a partir de duas funções dadas, resulta numa terceira que mede a área subentendida pela superposição das mesmas em função do deslocamento existente entre elas.*

- Definição informal:
  - <texto omitido>
- Definição informal e prática (para nós):
  - Convolução é a aplicação de um filtro linear espacial.
  - (Correlação cruzada).
- Definição já-subvertendo-o-termo:
  - Convolução é a aplicação de qualquer filtro espacial.
    - Formalmente, aplicar um filtro da mediana **não é** uma convolução, mas você pode encontrar textos informais dizendo algo como *convolution with a median filter*.

# Filtro Gaussiano

•Filtro Gaussiano: um dos filtros mais importantes e mais usados em processamento de imagens.

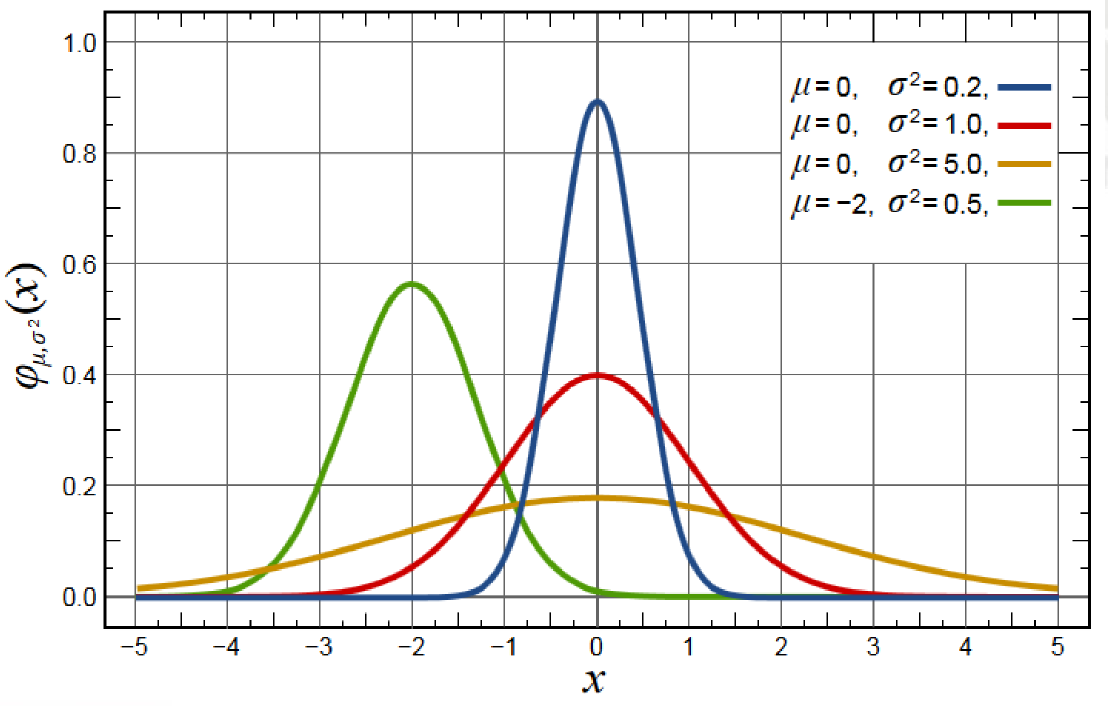
- Talvez seja **o** filtro mais importante.
- Também é chamado de suavização Gaussiana, por borrar a imagem.
- É um filtro linear e espacial.
- Os valores no *kernel* de um filtro Gaussiano são uma aproximação dos valores em uma distribuição Gaussiana.



# Distribuição Gaussiana?

# Para quem esqueceu...

- A distribuição Gaussiana / normal é uma distribuição de probabilidades extremamente comum no mundo real.
  - Se tomarmos um número grande de amostras de várias coisas (ex: altura de uma população), veremos que os valores se distribuem ao redor de uma média, com a distribuição podendo ser descrita de forma aproximada por uma média (μ) e um desvio padrão (σ).



# Distribuição Gaussiana 1D

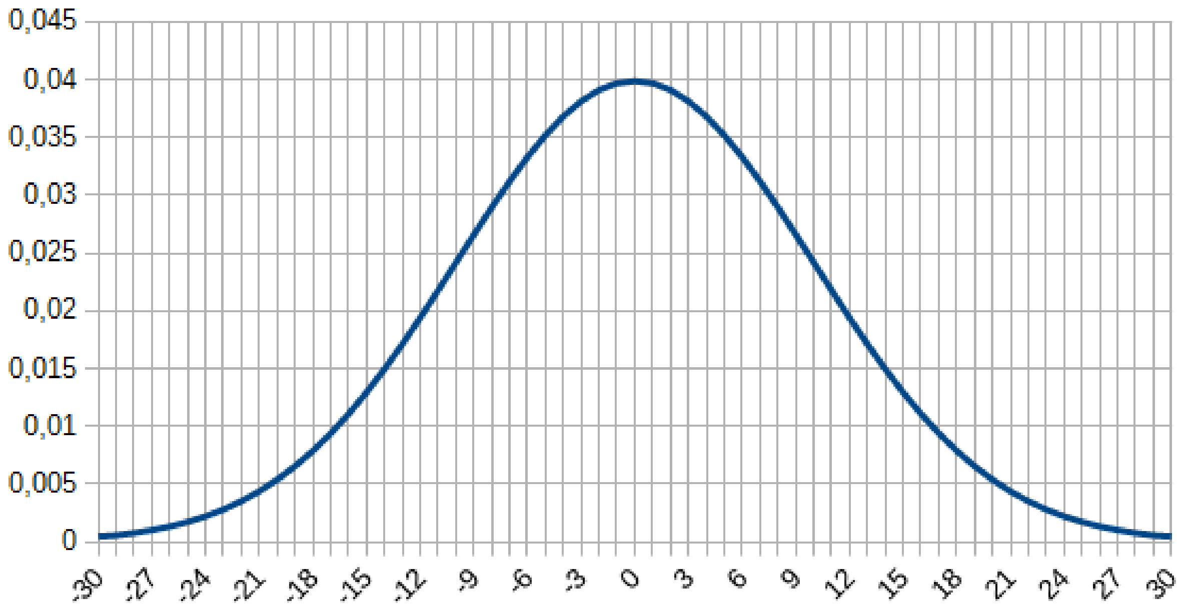
- A distribuição Gaussiana é muito usada em processamento de imagens e visão computacional.
  - Exemplo: o ruído em um sensor é muitas vezes modelado como seguindo uma distribuição Gaussiana com média 0.
  - Pressupor uma distribuição Gaussiana é muitas vezes uma simplificação grosseira, mas que funciona “suficientemente bem”.
- A distribuição Gaussiana é descrita por:
  - Um valor médio (μ).
  - Um desvio padrão (σ).

$$f(x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}}$$

Gaussiana 1D com média μ = 0

# Distribuição Gaussiana 1D

- Em uma distribuição Gaussiana...
  - A área sob a curva é igual a 1 (= 100%).
  - A área no intervalo $(\mu-\sigma, \mu+\sigma)$ é 0.6827 (= 68.27%).
  - A área no intervalo $(\mu-2\sigma, \mu+2\sigma)$ é 0.9545 (= 95.45%).
  - A área no intervalo $(\mu-3\sigma, \mu+3\sigma)$ é 0.9973 (= 99.73%).




Gaussiana 1D com $\mu = 0$ e $\sigma = 10$.

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

# Distribuição Gaussiana 2D

•Supondo que a distribuição está alinhada aos eixos *x* e *y*, podemos descrever a distribuição Gaussiana 2D com duas médias e dois desvios padrões.

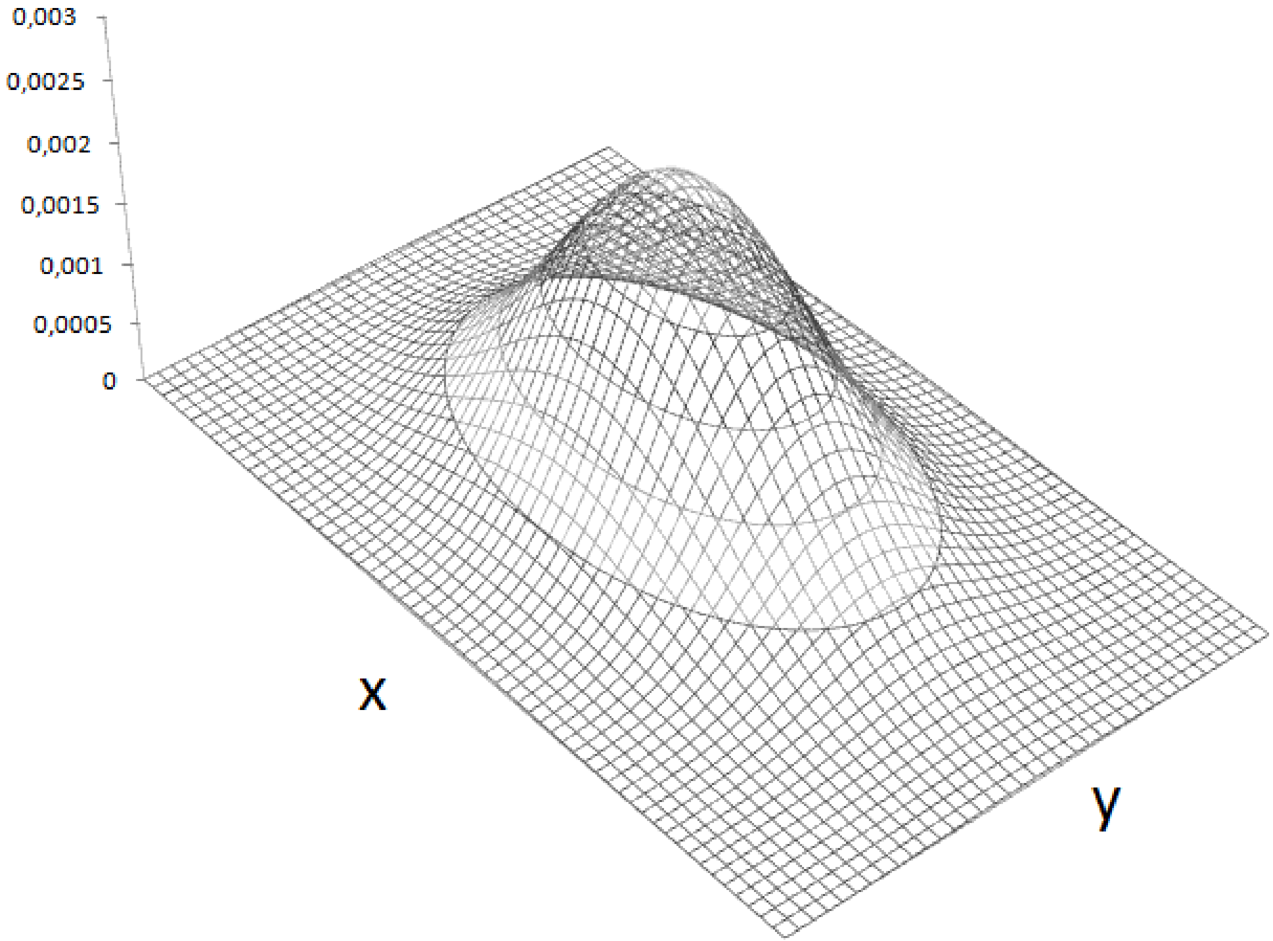
$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Gaussiana 2D com $\mu_x = \mu_y = 0$, e $\sigma_x = \sigma_y$.

•Propriedade importante: uma distribuição Gaussiana 2D pode ser descrita pela multiplicação de duas distribuições Gaussianas 1D.

# Distribuição Gaussiana 2D

- Curva para $\sigma_x = 10$ e $\sigma_y = 8$, com $\mu = 0$ em ambos os casos.



# Filtragem Gaussiana

- Na filtragem Gaussiana, o *kernel* se aproxima de uma distribuição Gaussiana 2D alinhada aos eixos *x* e *y, com* $\mu_x = \mu_y = 0$.

Valores de *x* e *y* = distância até o centro do *kernel*.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| ... | -3,-2 | -2,-2 | -1,-2 | 0,-2 | 1,-2 | 2,-2 | 3,-2 | ... |
| ... | -3,-1 | -2,-1 | -1,-1 | 0,-1 | 1,-1 | 2,-1 | 3,-1 | ... |
| ... | -3,0 | -2,0 | -1,0 | 0,0 | 1,0 | 2,0 | 3,0 | ... |
| ... | -3,1 | -2,1 | -1,1 | 0,1 | 1,1 | 2,1 | 3,1 | ... |
| ... | -3,2 | -2,2 | -1,2 | 0,2 | 1,2 | 2,2 | 3,2 | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

# Filtragem Gaussiana

- O *kernel* é simétrico em cada eixo.
  - = os valores à esquerda do centro são iguais aos valores à direita do centro, mas "espelhados".

*Kernel* do filtro Gaussiano com $\sigma_x = \sigma_y = 1$.

Qual seria o efeito de ter valores maiores de $\sigma$?

| | | | | | | |
|---|---|---|---|---|---|---|
| ... | ... | ... | ... | ... | ... | ... |
| ... | 0,003 | 0,013 | 0,022 | 0,013 | 0,003 | ... |
| ... | 0,013 | 0,059 | 0,097 | 0,059 | 0,013 | ... |
| ... | 0,022 | 0,097 | 0,159 | 0,097 | 0,022 | ... |
| ... | 0,013 | 0,059 | 0,097 | 0,059 | 0,013 | ... |
| ... | 0,003 | 0,013 | 0,022 | 0,013 | 0,003 | ... |
| ... | ... | ... | ... | ... | ... | ... |

# Sobre os coeficientes...

- Como o *kernel* é discreto, os valores formam apenas uma aproximação de uma curva Gaussiana.
- Existem várias formas de se discretizar a curva!
  - Solução simples: usar diretamente as distâncias até o centro.
    - = amostrar em intervalos de 1 pixel.
    - Basta substituir os valores de σ, *x* e *y* na fórmula.
  - Solução mais precisa (e demorada): amostrar com passos menores que 1 pixel, acumulando os valores para cada posição do *kernel*.
    - Ex: variar *x* e *y* em 0.00001.

# Sobre o tamanho do *kernel*...

- Teoricamente, os coeficientes são *sempre* diferentes de 0, mesmo para grandes distâncias até o centro.
  - Sim, até a posição (-99999999999999999,-99999999999999999)!!!
- É impossível usar um *kernel* infinito.
  - Qual é o menor *kernel* que <u>garante</u> que temos todos os valores necessários?

# Sobre o tamanho do *kernel*...

- Mesmo que os valores do *kernel* sejam sempre diferentes de 0, posições muito distantes do centro têm valores muito baixos.
  - Podemos dizer até que são desprezíveis para o resultado final.
- Qual seria a altura / largura de um *kernel* que cobre 99.73% da área total sob a curva Gaussiana?

# Sobre o tamanho do *kernel*...

- Qual seria a altura / largura de um *kernel* que cobre 99.73% da área total sob a curva Gaussiana?

- Podemos cobrir o intervalo de $-3\sigma$ a $+3\sigma$.
  - (lembrando que estamos supondo $\mu = 0$).
  - Exemplo: se $\sigma = 2.1$, podemos cobrir o intervalo de -6.3 a +6.3.
    - Isso nos dá uma altura / largura igual a 12.6.
  - Para garantir que o *kernel* tem um centro único, arredondamos este valor para o próximo número ímpar.

- É normal usar outros valores, como $\pm 2\sigma$ ou $\pm 1.5\sigma$.
  - *Kernel* maior → mais próximo de uma curva Gaussiana real.
  - *Kernel* menor → menos cálculos.

# Sobre o tamanho do *kernel*...

- Descartar valores → a soma dos coeficientes será menor que 1.
  - (1 é a área total sob uma curva Gaussiana).
- Qual o efeito disso no resultado?

# Sobre o tamanho do *kernel*...

- Descartar valores → a soma dos coeficientes será menor que 1.
  - (1 é a área total sob uma curva Gaussiana).
- Qual o efeito disso no resultado?

Cortando em ±1.5σ

Cortando em ±2σ

Cortando em ±3σ

# Sobre o tamanho do *kernel*...

- Descartar valores → a soma dos coeficientes será menor que 1.
  - (1 é a área total sob uma curva Gaussiana).

- Qual o efeito disso no resultado?
  - R: a imagem ficará mais escura.
    - Solução: dividir todos os coeficientes pela soma dos mesmos.
      - = normalizar o *kernel*.
      - É como redistribuir os valores que foram descartados!

- Como os coeficientes são normalizados no final, podemos também simplificar a equação!

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

Todos os coeficientes serão divididos por isso!

Calculamos assim, e a normalização re-escala tudo!

| | | | | | | |
|---|---|---|---|---|---|---|
| 0,00022 | 0,00122 | 0,00345 | 0,00488 | 0,00345 | 0,00122 | 0,00022 |
| 0,00122 | 0,00691 | 0,01958 | 0,02771 | 0,01958 | 0,00691 | 0,00122 |
| 0,00345 | 0,01958 | 0,05549 | 0,07852 | 0,05549 | 0,01958 | 0,00345 |
| 0,00488 | 0,02771 | 0,07852 | 0,11112 | 0,07852 | 0,02771 | 0,00488 |
| 0,00345 | 0,01958 | 0,05549 | 0,07852 | 0,05549 | 0,01958 | 0,00345 |
| 0,00122 | 0,00691 | 0,01958 | 0,02771 | 0,01958 | 0,00691 | 0,00122 |
| 0,00022 | 0,00122 | 0,00345 | 0,00488 | 0,00345 | 0,00122 | 0,00022 |

*Kernel* do filtro Gaussiano com $\sigma_x = \sigma_y = 1.2$, cobrindo $\pm 3\sigma$.

| | | | | |
|---|---|---|---|---|
| 0,00733 | 0,02078 | 0,02941 | 0,02078 | 0,00733 |
| 0,02078 | 0,05889 | 0,08333 | 0,05889 | 0,02078 |
| 0,02941 | 0,08333 | 0,11793 | 0,08333 | 0,02941 |
| 0,02078 | 0,05889 | 0,08333 | 0,05889 | 0,02078 |
| 0,00733 | 0,02078 | 0,02941 | 0,02078 | 0,00733 |

*Kernel* do filtro Gaussiano com $\sigma_x = \sigma_y = 1.2$, cobrindo $\pm 2\sigma$.

# Alguns *kernels* muito comuns...

- Se você achar em algum lugar uma referência à suavização Gaussiana 3x3, 5x5 ou 7x7, possivelmente o kernel é um desses...
  - ... (mas existem outras versões comuns para os kernels 5x5 e 7x7).

1/16 ·

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 2 | 4 | 2 |
| 1 | 2 | 1 |

$\sigma_x = \sigma_y = 0.8$

1/256 ·

| | | | | |
|---|---|---|---|---|
| 1 | 4 | 6 | 4 | 1 |
| 4 | 16 | 24 | 16 | 4 |
| 6 | 24 | 36 | 24 | 6 |
| 4 | 12 | 24 | 16 | 4 |
| 1 | 4 | 6 | 4 | 1 |

$\sigma_x = \sigma_y = 1.1$

1/4096 ·

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 14 | 28 | 36 | 28 | 14 | 4 |
| 14 | 49 | 98 | 126 | 98 | 49 | 14 |
| 28 | 98 | 196 | 252 | 196 | 98 | 28 |
| 36 | 126 | 252 | 324 | 252 | 126 | 36 |
| 28 | 98 | 196 | 252 | 196 | 98 | 28 |
| 14 | 49 | 98 | 126 | 98 | 49 | 14 |
| 4 | 14 | 28 | 36 | 28 | 14 | 4 |

$\sigma_x = \sigma_y = 1.4$

# Implementando o filtro Gaussiano

- Repetindo: uma distribuição Gaussiana 2D pode ser descrita pela multiplicação de duas distribuições Gaussianas 1D.
  - O que isso nos diz sobre a implementação do filtro Gaussiano?

# Implementando o filtro Gaussiano

- Repetindo: uma distribuição Gaussiana 2D pode ser descrita pela multiplicação de duas distribuições Gaussianas 1D.
  - = o filtro Gaussiano é separável.
  - Podemos aplicar primeiro um filtro horizontal com $\sigma = \sigma_x$, e depois um filtro vertical com $\sigma = \sigma_y$ (ou vice-versa).

- Exemplo: o filtro 5x5 mostrado anteriormente pode ser obtido aplicando os dois *kernels* abaixo em sequência.

$$1/16 \cdot \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$1/16 \cdot \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix}$$

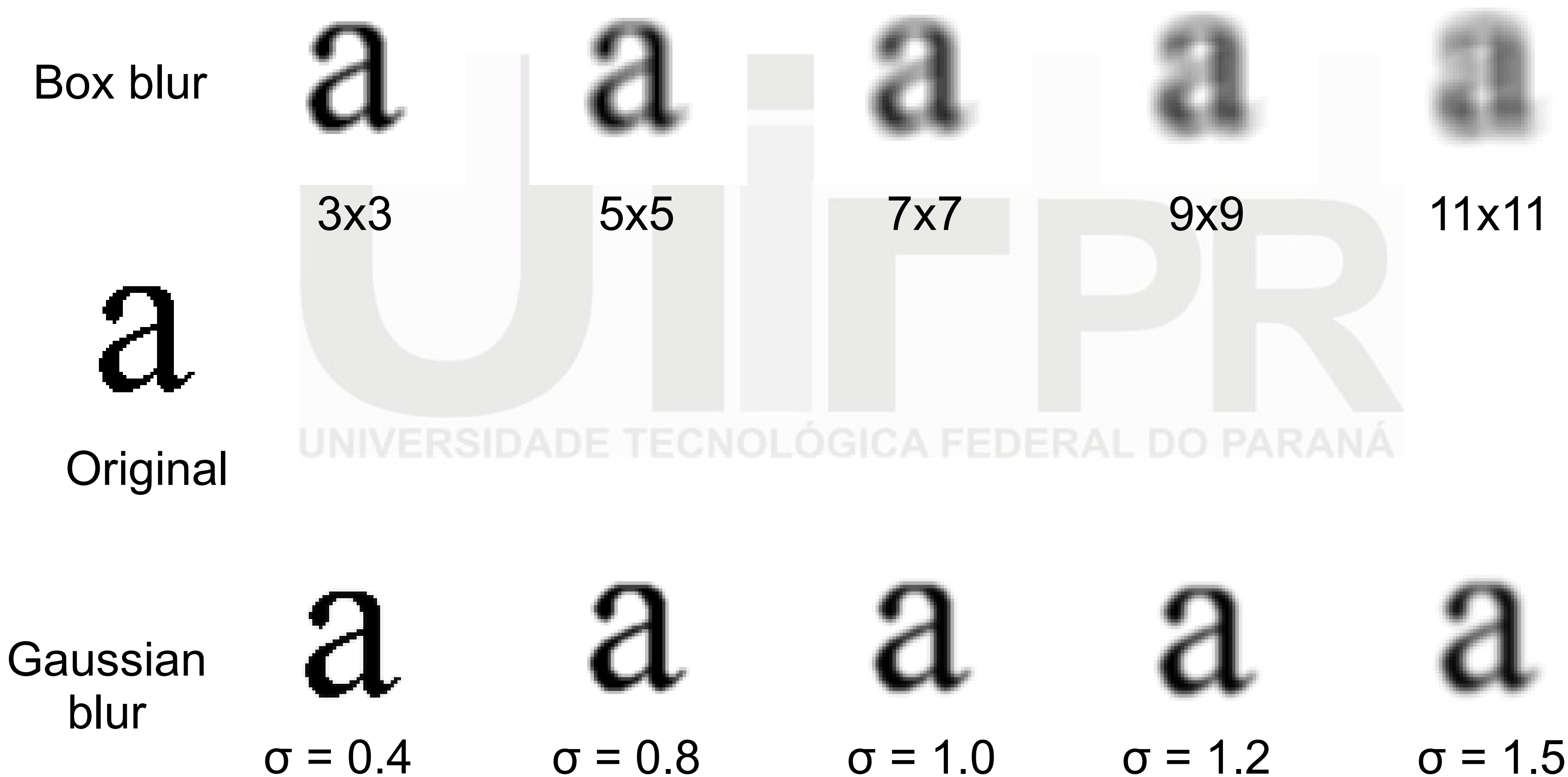
# *Box blur* x *Gaussian blur*

- Assim como o filtro da média, o filtro Gaussiano borra a imagem.
- Podemos usar o filtro Gaussiano em aplicações similares àquelas em que usamos o filtro da média, como:
  - Limiarização adaptativa.
  - Redução de ruído.
  - Efeito *depth-of-field.*
- Usando imagens integrais, o filtro da média pode ser aplicado muito rapidamente.
- Qual é então a vantagem da suavização Gaussiana?

# *Box blur* x *Gaussian blur*

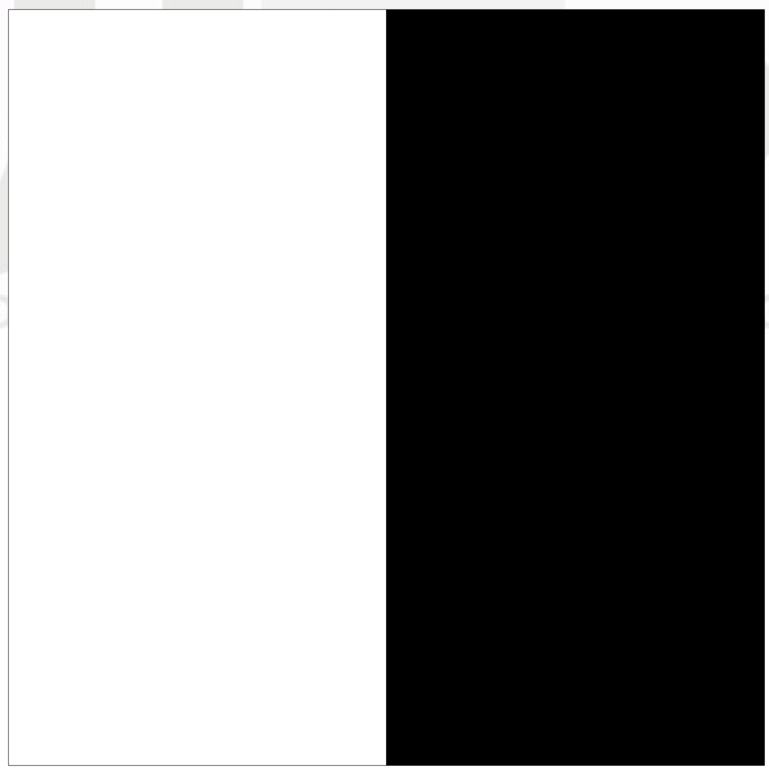
- Qual é então a vantagem da suavização Gaussiana?
  - Por enquanto, vamos destacar duas diferenças visíveis:
    - O filtro Gaussiano permite variações menores na quantidade de suavização.
    - O filtro Gaussiano não cria “fantasmas”.
  - Voltaremos a este assunto em aulas futuras!

# *Box blur x Gaussian blur*

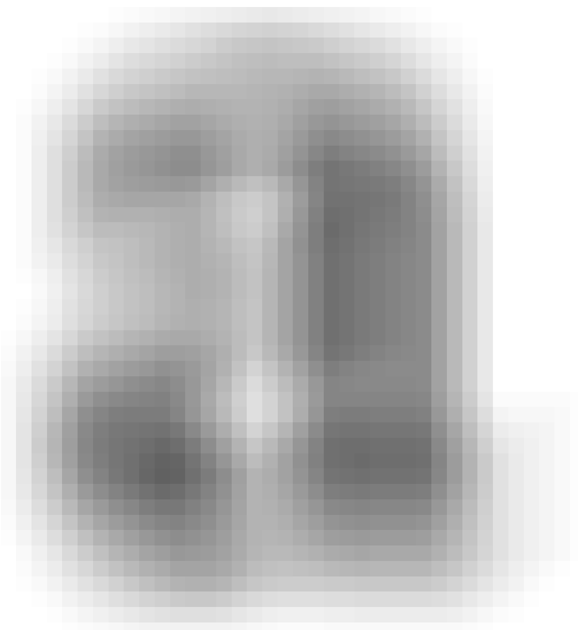


# *Box blur* x *Gaussian blur*

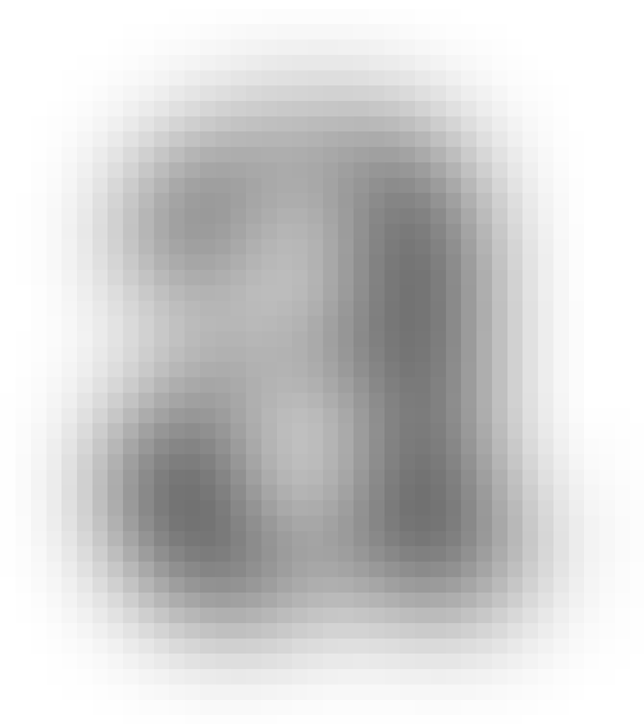
- Considere a linha central da imagem abaixo.
- Como a linha ficaria após a aplicação do filtro da média? E se aplicarmos o filtro Gaussiano?



# Box blur x Gaussian blur

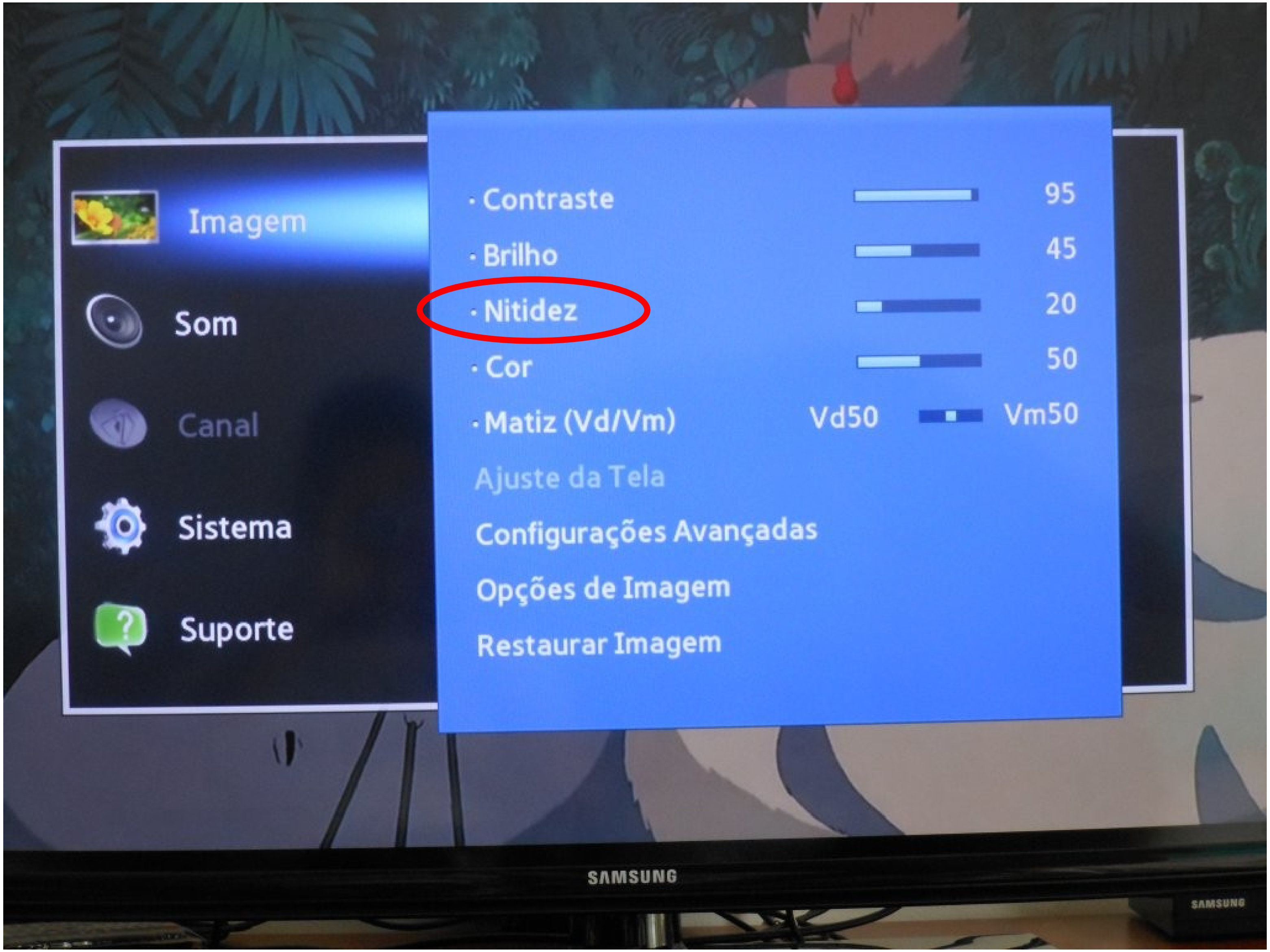


Box blur



Gaussian blur

# E à direita do centro?!




UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

# Unsharp masking

- Como realçar os detalhes de uma imagem?
  - Aviso: realçar os detalhes também realça ruídos!

# Unsharp masking

Uma imagem com 256x256 pixels.

# Unsharp masking

Após um filtro Gaussiano 3x3. É uma versão com menos detalhes da imagem.

# Unsharp masking

- A diferença entre as duas versões da imagem está nos detalhes.

# Unsharp masking

Resultado da subtração Original – Borrada.
A diferença foi reescalada para o intervalo
[-1,1], para facilitar a visualização.

# Unsharp masking

- *Unsharp masking*:
  - Cria uma versão borrada (= "*unsharp*") da imagem original.
  - Subtrai a versão borrada da imagem original.
    - O que resta é uma imagem contendo os detalhes removidos.
  - Verifica quais são os detalhes mais significativos.
    - = onde a diferença é maior que um limiar $T$.
  - Soma os detalhes significativos à imagem original.
    - As diferenças normalmente têm valores pequenos, então multiplicamos elas por um $\alpha$ antes da soma.

# Unsharp masking: parâmetros

- σ do filtro Gaussiano.
  - Quanto maior, maiores são os “detalhes” realçados.
  - Normalmente, é um valor pequeno.
- O limiar *T* que indica o que é um detalhe significativo.
  - Quanto maior, mais “afiados” os detalhes precisam ser.
  - Normalmente, é 0 ou quase isso.
- O multiplicador α aplicado à diferença antes da soma.
  - Quanto maior, mais realçados os detalhes são.
  - Em implementações simples, pode ser o único parâmetro variável.

# Imagem original

# Resultado

$\sigma = 0.8$ (3x3)
$T = 0.$
$\alpha = 2.0$

# Resultado

σ = 0.8 (3x3)
*T = 0.*
α = 5.0

# Resultado

σ = 1.4 (7x7)
*T = 0.*
α = 5.0

# Imagem original

# Resultado