

# Medidas de Distância

- Definimos a distância entre pixels a função  $D$  (ou métrica  $D$ ) se:

(i)  $D(p, q) \geq 0$  ( $D(p, q) = 0$  se e somente se  $p = q$ )

(ii)  $D(p, q) = D(q, p)$ , e

(iii)  $D(p, z) \leq D(p, q) + D(q, z)$

# Medidas de Distância

- A **distância Euclidiana** –  $D_e$  entre  $p$  e  $q$  é definida como

$$D_e(p, q) = [(x - s)^2 + (y - t)^2]^{1/2}$$

- Na distância Euclidiana, os pixels com uma distância menor ou igual a um valor  $d$ , formam um disco de **raio  $d$**  centrado em  $p$ .

Considere  $D_e \leq 2$  de um ponto  $(x, y)$  formam o seguinte conjunto:

$$\begin{array}{ccccc} & & 2 & & \\ & \sqrt{2} & 1 & \sqrt{2} & \\ 2 & 1 & 0 & 1 & 2 \\ & \sqrt{2} & 1 & \sqrt{2} & \\ & & 2 & & \end{array}$$

# Medidas de Distância

- Apesar da distância Euclidiana ser mais próxima do caso contínuo, requer mais esforço computacional.
- A distância city-block –  $D_4$  é definida por:  $D_4(p, q) = |x - s| + |y - t|$

Considere  $D_4 \leq 2$  de um ponto  $(x, y)$  formam o seguinte conjunto:

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

- Note que os pixels a uma distância  $D_4$  de p, menor ou igual a algum valor d formam um losango centrado em p.

# Medidas de Distância

A distância chessboard – D8 é definida por:  $D_8(p, q) = \max\{|x - s|, |y - t|\}$

- Note que os pixels a uma distância D8 de p, menor ou igual a algum valor **d** formam um quadrado centrado em p.

Considere  $D_4 \leq 2$  de um ponto  $(x, y)$  formam o seguinte conjunto:

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

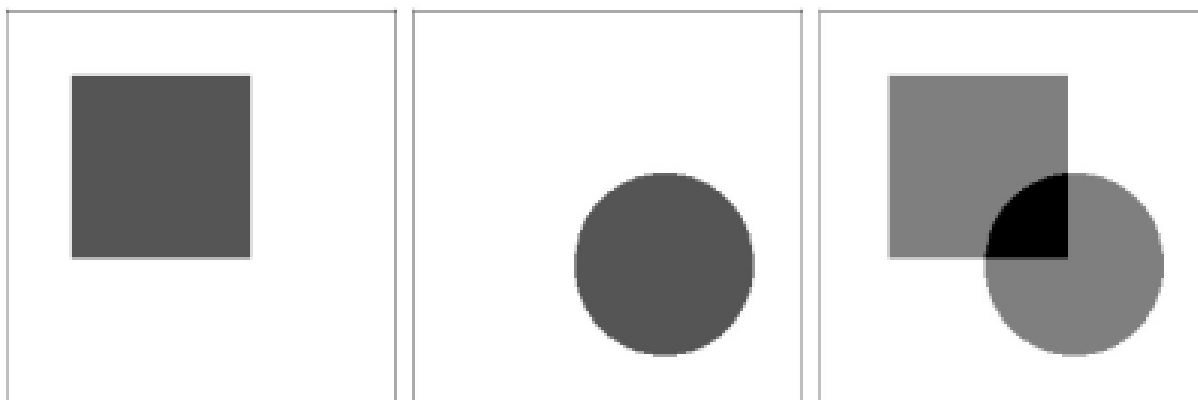
- A distância D4 entre dois pixels p e q é igual ao comprimento do caminho mais curto considerando vizinhança – 4, do mesmo modo D8 corresponde ao caminho considerando vizinhança – 8

# Operações Lógicas e Aritméticas

- Após uma imagem ter sido adquirida e digitalizada, ela pode ser vista como uma matriz de inteiros.
- Utilizando operações lógicas e/ou aritmética é possível manipular essas matrizes.
- As operações podem ser efetuadas pixel a pixel ou orientadas a vizinhança.

# Adição

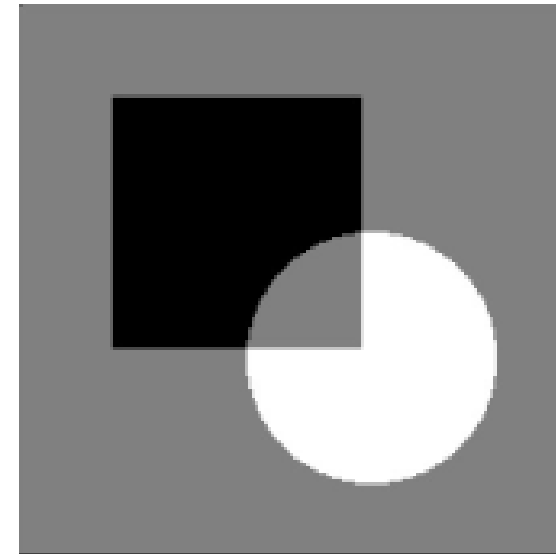
- **Efeito sobre a imagem:**  $Z$  é o resultado da soma dos valores de intensidade de  $X$  e  $Y$
- Se  $Y$  for um escalar positivo,  $Z$  será uma versão mais clara de  $X$ ; o acréscimo de intensidade será o próprio valor de  $Y$



- **Aplicações:** Normalização de brilho de imagens e Remoção de ruídos

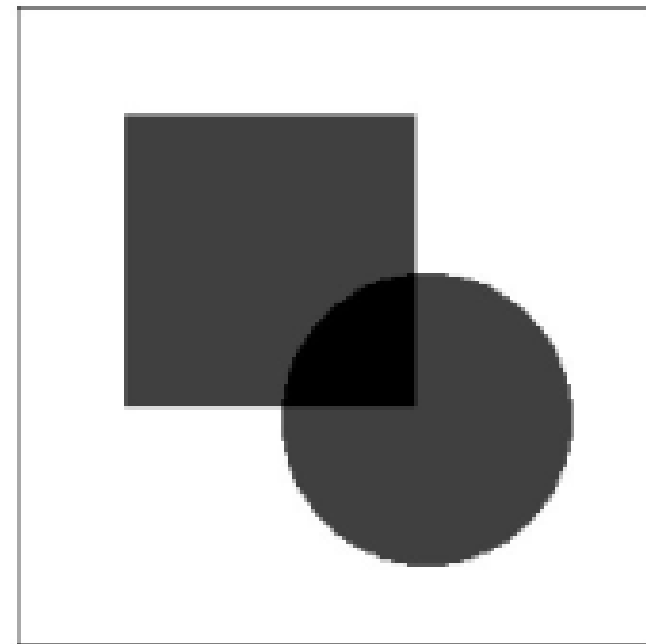
# Subtração

- **Efeito sobre a imagem:**  $Z$  é o resultado da diferença dos valores de intensidade de  $X$  e  $Y$
- Se  $Y$  for um escalar positivo,  $Z$  será uma versão mais escura de  $X$ ; o decréscimo de intensidade será o próprio valor de  $Y$
- **Aplicações:** Normalização de brilho de imagens e Remoção de ruídos



# Multiplicação

- **Efeito sobre a imagem:**  $Z$  é o produto dos valores de intensidade de  $X$  e  $Y$ .
- Se  $Y$  for um escalar positivo, os valores de intensidade de  $Z$  serão diretamente proporcionais a  $X$  por um fator  $Y$ .



- **Aplicações:** Calibração de brilho



# Divisão

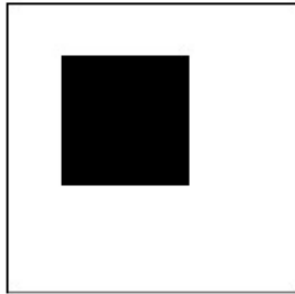
- **Efeito sobre a imagem:**  $Z$  é o razão dos valores de intensidade de  $X$  pelos valores correspondentes em  $Y$ .
- Se  $Y$  for um escalar positivo, os valores de intensidade de  $Z$  serão inversamente proporcionais a  $X$  por um fator  $Y$ .
- **Aplicações:** Normalização de brilho



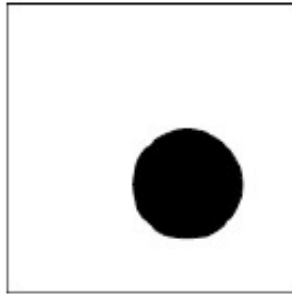
# Operações Lógicas pixel a pixel

- Todas as operações lógicas (ou booleanas) conhecidas podem ser aplicadas entre imagens, inclusive a operação de complemento (NOT), que é uma operação unária (requer apenas um operando)
- Operações lógicas podem ser efetuadas em imagens com qualquer número de níveis de cinza mas são melhor compreendidas quando vistas em imagens binárias.

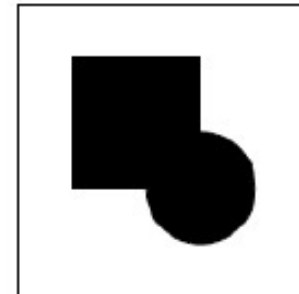
# Operações Lógicas pixel a pixel



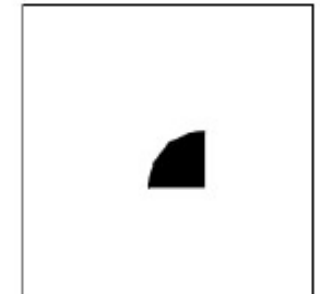
X



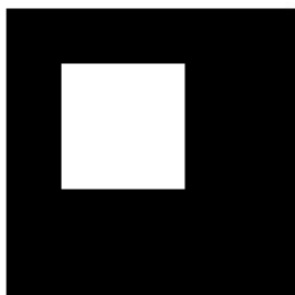
Y



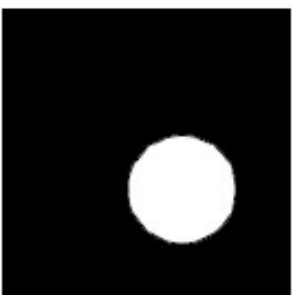
X and Y



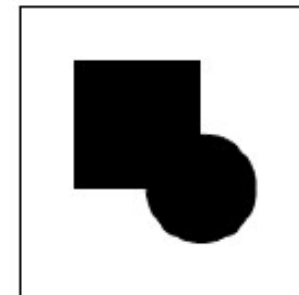
X or Y



not X



not Y



X xor Y



(not X) and Y

# EXERCÍCIO 4

- Implemente um algoritmo para adição de imagens
- Considere as imagens disponíveis em <https://links.uwaterloo.ca/Repository.html>
- Gere um ruído Gaussiano branco (média zero e desvio padrão 100) e adiciona à imagem ao lado



# Operações de convolução com máscaras

- O processo de convolução bidimensional é uma expansão do processo de convolução unidimensional
- Considere uma imagem que pode ser representada pela seguinte matriz, A,

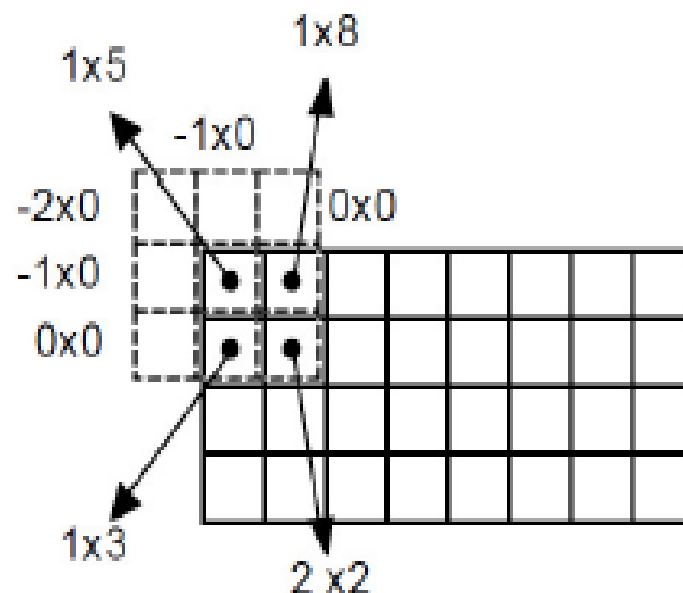
5	8	3	4	6	2	3	7
3	2	1	1	9	5	1	0
0	9	5	3	0	4	8	3
4	2	7	2	1	9	0	6
9	7	9	8	0	4	2	4
5	2	1	8	4	1	0	9
1	8	5	4	9	2	3	8
3	7	1	2	3	4	4	6

# Operações de convolução com máscaras

- e uma matriz B (máscara)

$$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

A operação de convolução bidimensional será



$$(-2 \times 0) + (-1 \times 0) + (0 \times 0) + (-1 \times 0) + (1 \times 5) + (1 \times 8) + (0 \times 0) + (1 \times 3) + (2 \times 2) = 20$$

Diagram illustrating the result of the convolution operation. The output is a 3x3 grid with the value 20 in the top-left cell.

# Operações de convolução com máscaras

- O resultado da convolução será uma matriz de mesma dimensão de A

20	10	2	26	23	6	9	4
18	1	-8	2	7	3	3	-11
14	22	5	-1	9	-2	8	-1
29	21	9	-9	10	12	-9	-9
21	1	16	-1	-3	-4	2	5
15	-9	-3	7	-6	1	17	9
21	9	1	6	-2	-1	23	2
9	-5	-25	-10	-12	-15	-1	-12

# Operações de convolução com máscaras

- Para calcular os valores resultantes dos pixels próximos às bordas da imagem, podem ser adotadas diversas estratégias, dentre elas:
  1. preencher com zeros o contorno da imagem, de maneira condizente com o tamanho de máscara utilizado
  2. preencher o contorno da imagem com os mesmos valores da(s) primeira(s) e última(s) linha(s) e coluna(s)
- Prevenir a eventual introdução de erros nas regiões de bordas da imagem causados por qualquer um dos métodos anteriores, considerando na imagem resultante apenas os valores para os quais a máscara de convolução ficou inteiramente contida na imagem original.



# Aplicações: Detecção de pontos isolados

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Essa máscara é um exemplo de operador de convolução que, quando aplicado a uma imagem, destacará pixels brilhantes circundados por pixels mais escuros
- Este operador corresponde a um filtro passa-altas

# Aplicações: Detecção de linhas

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

**Linhas horizontais**

**Linhas verticais**

$$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

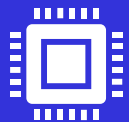
**Linhas Diagonais**

$$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

# Aplicações: Detecção de bordas

- Define-se **borda (edge)** como a fronteira entre duas regiões cujos níveis de cinza predominantes são razoavelmente diferentes:
  - **borda de luminosidade** é uma descontinuidade na luminosidade de uma imagem.
  - **borda de textura ou borda de cor** é uma descontinuidade na textura, é usada em imagens onde as informações de textura ou cor, respectivamente, são as mais importantes

# APLICAÇÕES: DETECÇÃO DE BORDAS



O tema detecção de bordas (edge detection) vem desafiando os pesquisadores da área de Processamento de Imagens há muitos anos e sobre ele continuam sendo experimentadas novas técnicas, cujos resultados são publicados ainda hoje nos mais conceituados periódicos científicos mundiais



Trata-se, portanto, de um tema em aberto, a detecção de bordas em cenas consideradas “difíceis”



Trataremos apenas as bordas de luminosidade, às quais denominaremos simplesmente bordas

# Aplicações: Detecção de bordas

- Apenas para efeito didático à operação de convolução com máscaras apresentaremos algumas máscaras que podem ser utilizadas para a tarefa de detecção de bordas
- Para a detecção e realce de bordas, aplicam-se habitualmente filtros espaciais lineares de dois tipos:
  - (a) baseados no gradiente da função de luminosidade,  $I(x, y)$ , da imagem, e
  - (b) baseados no laplaciano de  $I(x, y)$ .
- Tanto o gradiente quanto o laplaciano costumam ser aproximados por máscaras de convolução ou operadores  $3 \times 3$ .

# Aplicações: Detecção de bordas

Operador	Vertical	Horizontal
Roberts	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Sobel	$\frac{1}{4} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Prewitt	$\frac{1}{3} \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Frei-Chen	$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{2 + \sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

# Aplicações: Detecção de bordas



(a) imagem original



(b) realce de bordas  
utilizando os  
operadores de Prewitt  
horizontal e vertical



(c) realce de bordas  
utilizando os  
operadores de Sobel  
horizontal e vertical.

# Aplicações: Detecção de bordas usando Laplaciano

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 8 & 8 & 8 & -1 & -1 & -1 \\ -1 & -1 & -1 & 8 & 8 & 8 & -1 & -1 & -1 \\ -1 & -1 & -1 & 8 & 8 & 8 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

- Embora o laplaciano seja insensível à rotação, e portanto capaz de realçar ou detectar bordas em qualquer direção, seu uso é restrito devido a sua grande **suscetibilidade a ruído**.

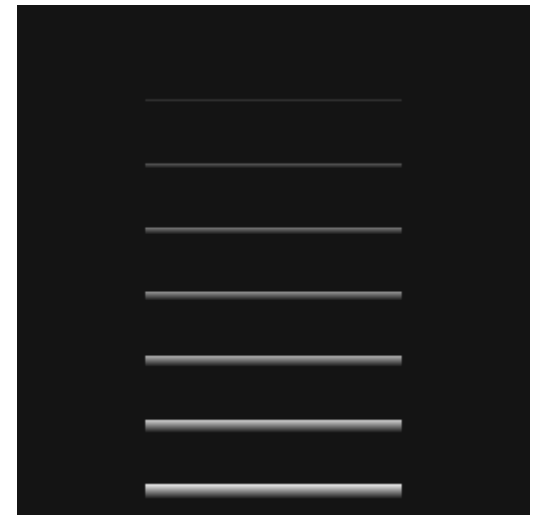
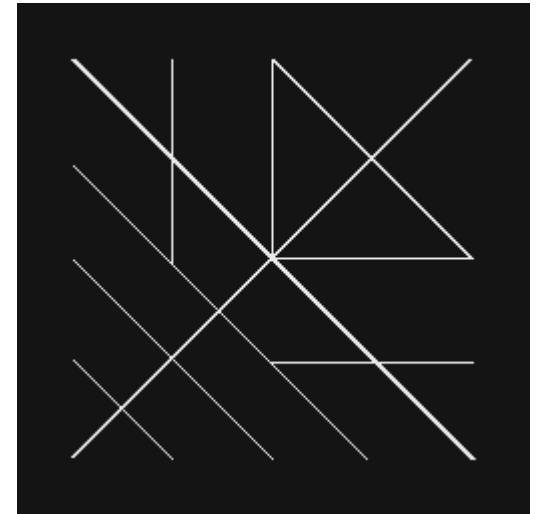
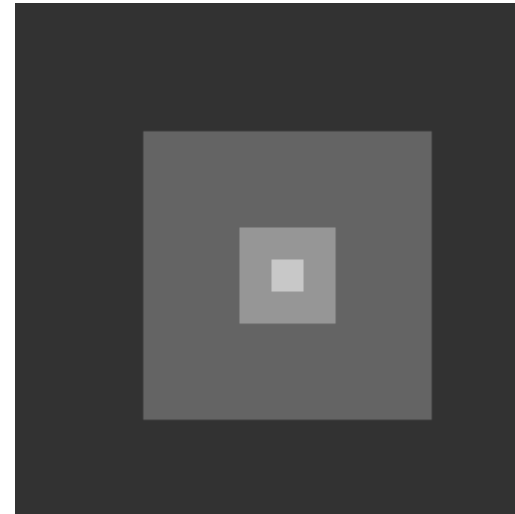


# Aplicações: Detecção de bordas usando Laplaciano



# EXERCÍCIO 5

- Implemente um algoritmo para detecção de linhas usando convolução com máscara e aplique nas imagens ao lado.
- Considere as imagens disponíveis em <https://links.uwaterloo.ca/Repository.html>



# EXERCÍCIO 6

- Implemente um algoritmo para detecção de bordas usando convolução com máscara e aplique na imagem ao lado
- Considere a imagem disponível em <https://links.uwaterloo.ca/Repository.html>

