Lecture 03 – Fundamentals of digital imaging II

Prof. João Fernando Mari
joaofmari.github.io
joaof.mari@ufv.br

Agenda

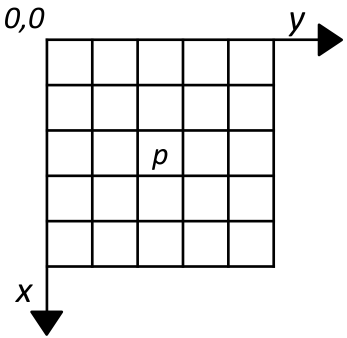
UFV
Universidade Federal de Viçosa

- Basic relationship between pixels
  - Neighborhood of a pixel
  - Adjacency
  - Digital path (or curve)
  - Regiões conectadas e componentes conectados
  - Fundo e objetos de uma imagem
  - Boundary, borders, contour, or frontier
- Logical and arithmetic operations between images
  - Arithmetic operations
  - Logical operations
- Distance measures

BASIC RELATIONSHIP BETWEEN PIXELS

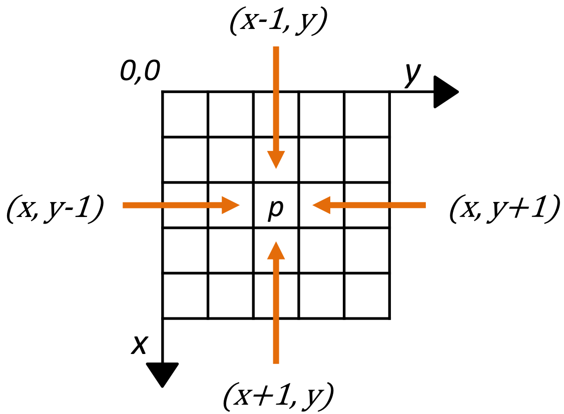
Neighborhood of a pixel

4-neighbors of p, $N_4(p)$:



Neighborhood of a pixel

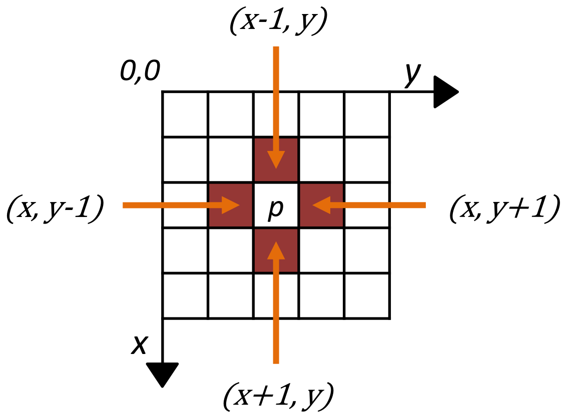
4-neighbors of p, $N_4(p)$:



Neighborhood of a pixel

4-neighbors of p, $N_4(p)$:

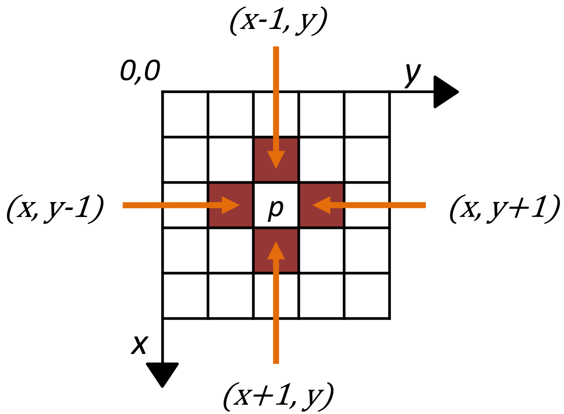
$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$

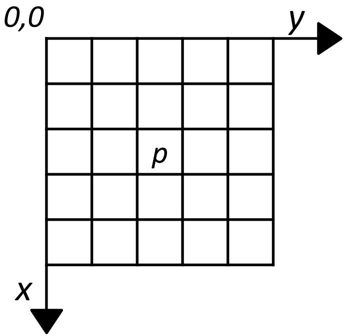


Neighborhood of a pixel

4-neighbors of p, $N_4(p)$:

$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$





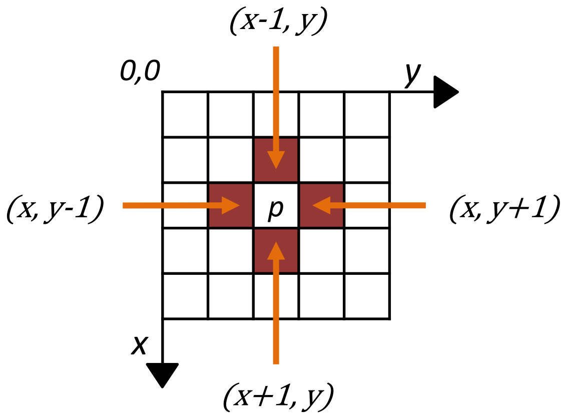
Neighborhood of a pixel

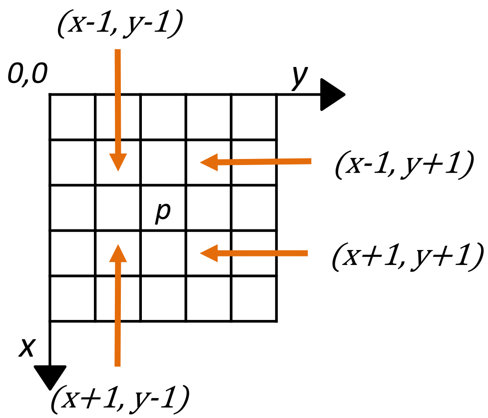
4-neighbors of p, $N_4(p)$:

$(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$

Diagonal-neighbors of p, $N_D(p)$:

$(x-1, y-1)$, $(x-1, y+1)$, $(x+1, y-1)$, $(x+1, y+1)$





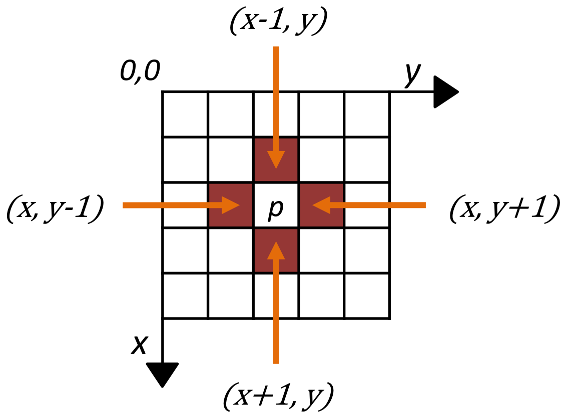
Neighborhood of a pixel

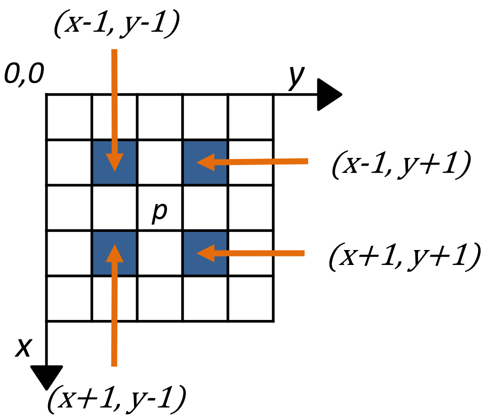
4-neighbors of p, $N_4(p)$:

$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$

Diagonal-neighbors of p, $N_D(p)$:

$(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)$





Neighborhood of a pixel

4-neighbors of p, $N_4(p)$:

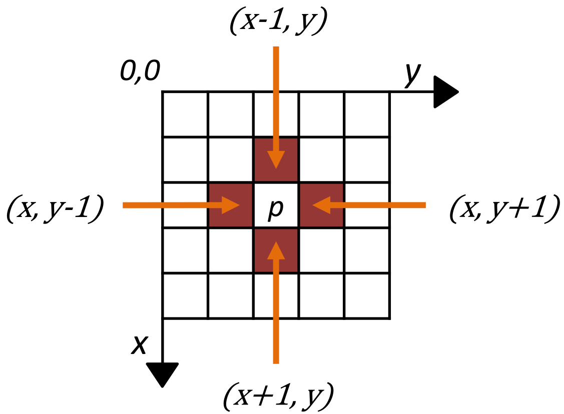
$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$

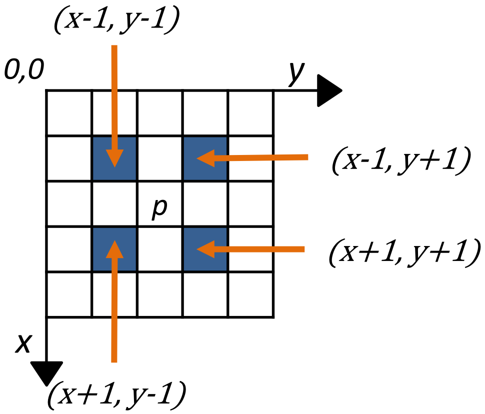
Diagonal-neighbors of p, $N_D(p)$:

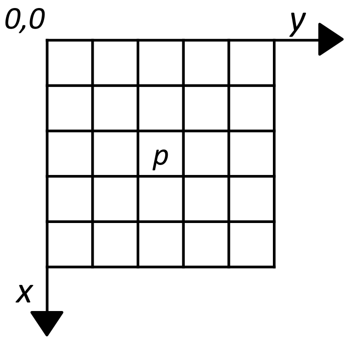
$(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)$

8-neighbors of p, $N_8(p)$:

$N_4(p) \cup N_D(p)$



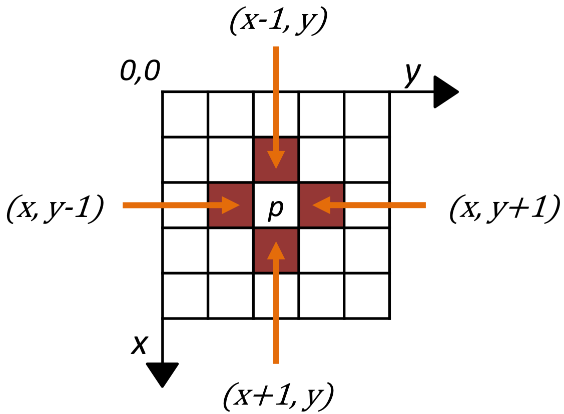




Neighborhood of a pixel

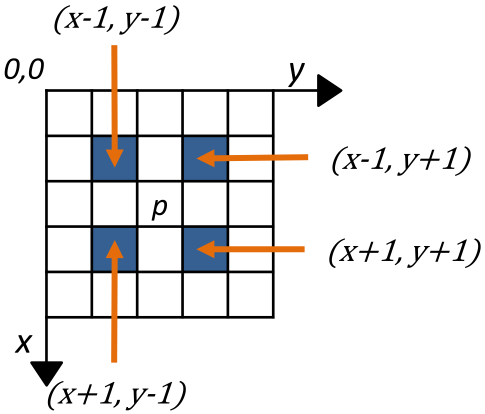
4-neighbors of p, $N_4(p)$:

$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$



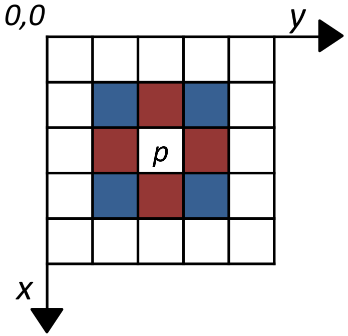
Diagonal-neighbors of p, $N_D(p)$:

$(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)$



8-neighbors of p, $N_8(p)$:

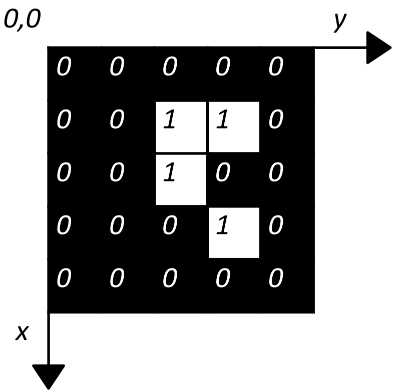
$N_4(p) \cup N_D(p)$



Adjacency

4-adjacency:

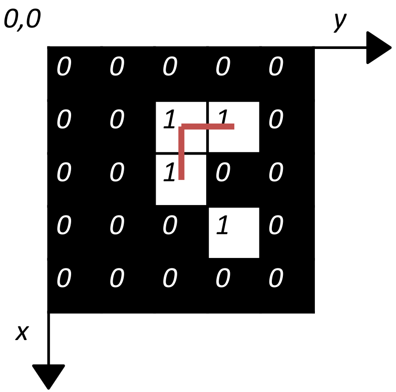
- Two pixels *p* and *q* are 4-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_4(p)$



(*) *V = {1}* for binary images

4-adjacency:

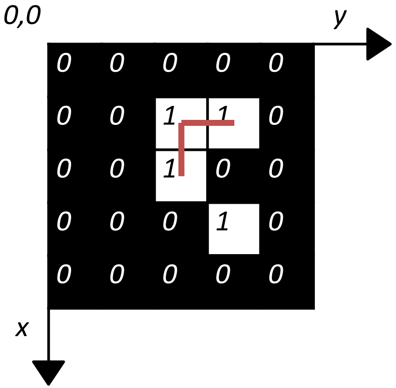
- Two pixels *p* and *q* are 4-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_4(p)$



(*) *V = {1}* for binary images

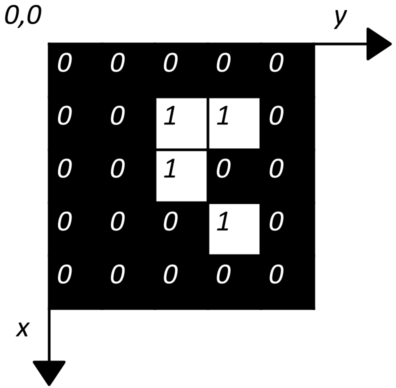
4-adjacency:

- Two pixels *p* and *q* are 4-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_4(p)$



8-adjaceny:

- Two pixels *p* and *q* are 8-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_8(p)$.

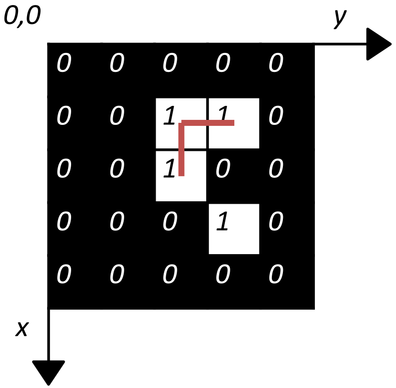


(*) *V = {1}* for binary images

Adjacency

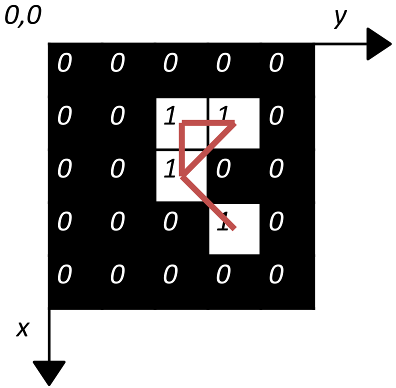
4-adjacency:

- Two pixels *p* and *q* are 4-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_4(p)$



8-adjaceny:

- Two pixels *p* and *q* are 8-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_8(p)$.



(*) *V = {1}* for binary images

4-adjacency:

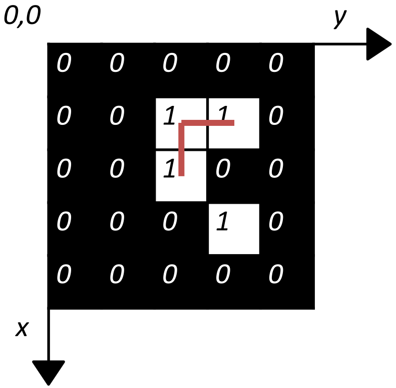
- Two pixels *p* and *q* are 4-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_4(p)$

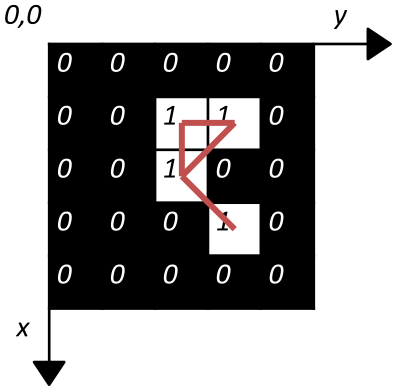
8-adjaceny:

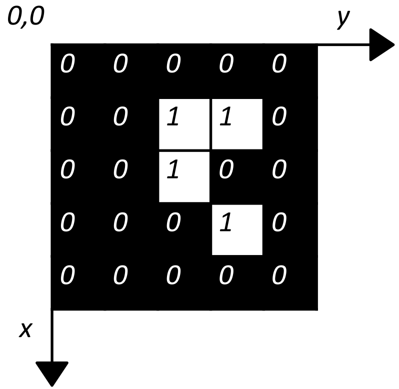
- Two pixels *p* and *q* are 8-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_8(p)$.

M-adjacency (mixed adjacency):

- Two pixels *p* and *q* are m-adjacent if:
  - *q* está em $N_4(p)$ **OU**
  - *q* estiver em $N_D(p)$ e a intersecção entre $N_4(p)$ e $N_4(q)$ não contém nenhum pixel cujos valores pertencem a *V*.







(*) *V = {1}* for binary images

Adjacency

4-adjacency:

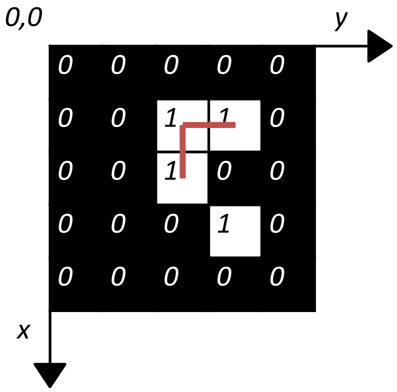
- Two pixels *p* and *q* are 4-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_4(p)$

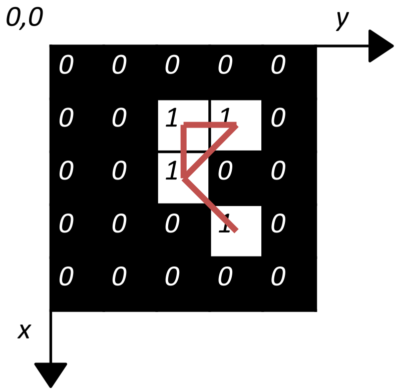
8-adjaceny:

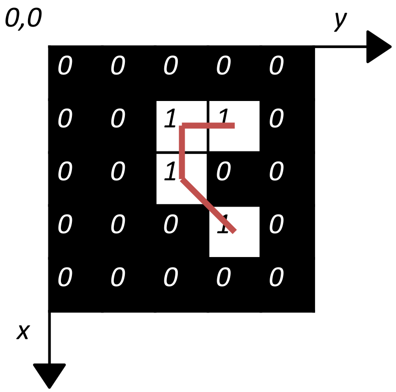
- Two pixels *p* and *q* are 8-adjacent if:
  - The values of *p* and *q* are in the set *V, and*
  - The pixel *q* is in the set $N_8(p)$.

Adjacência-m (adjacência mista):

- Dois pixels *p* e *q* são adjacentes-m se:
  - *q* está em $N_4(p)$ **OU**
  - *q* estiver em $N_D(p)$ e a intersecção entre $N_4(p)$ e $N_4(q)$ não contém nenhum pixel cujos valores pertencem a *V*.







(*) *V = {1}* for binary images

Digital path (or curve)

- A **path** from pixel ***p*** with coordinates (x, y) to pixel ***q*** with coordinates (s, t) is
  - A sequence of distinct pixels with coordinates:
    - $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$
  - where:
    - $(x_0, y_0) = (x, y)$,
    - $(x_n, y_n) = (s, t)$, *and*
    - the pixels (x_i, y_i) *and* (x_{i-1}, y_{i-1}) are adjacent for $1 \leq i \leq n$

- If $(x_0, y_0) = (x_n, y_n)$, the path is closed

- Depending on the type of adjacency, the paths can be:
  - 4-path
  - 8-path
  - m-path

Digital path (or curve)

Considerando vizinhança-4:

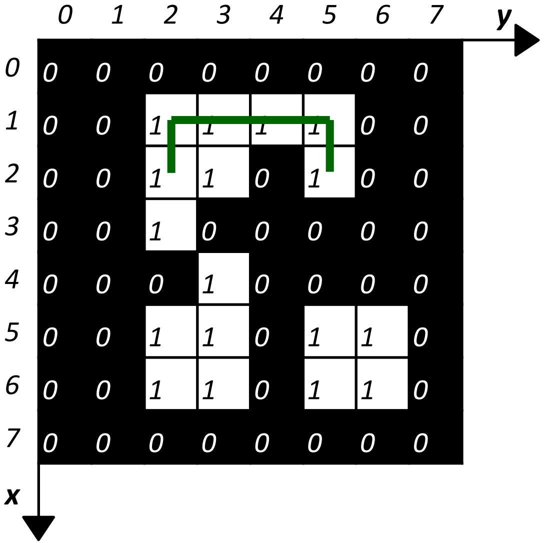
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	0	0
2	0	0	1	1	0	1	0	0
3	0	0	1	0	0	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	1	1	0	1	1	0
6	0	0	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Digital path (or curve)

Considerando vizinhança-4:

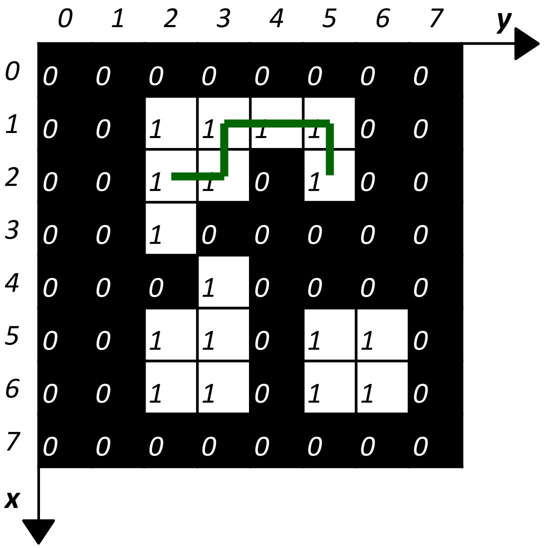
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*



Digital path (or curve)

Considerando vizinhança-4:

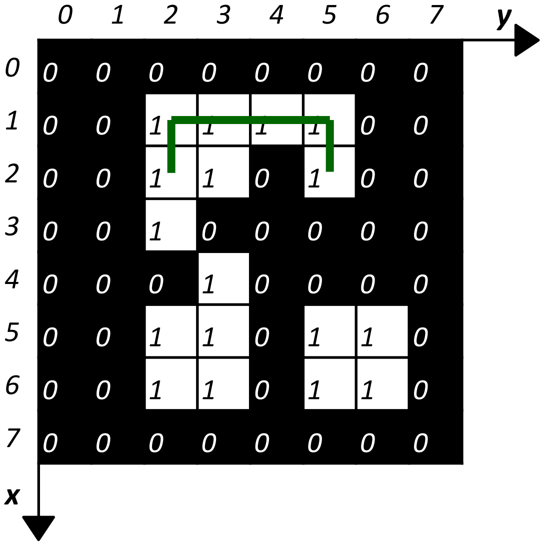
- Outro caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (2,3), (2,2).*



Digital path (or curve)

Considerando vizinhança-4:

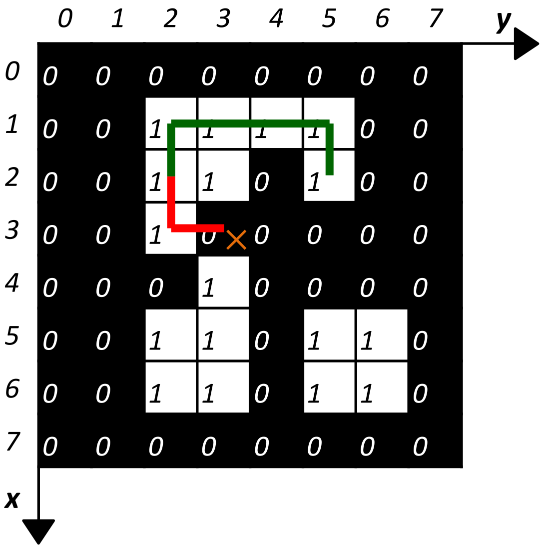
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:



Digital path (or curve)

Considerando vizinhança-4:

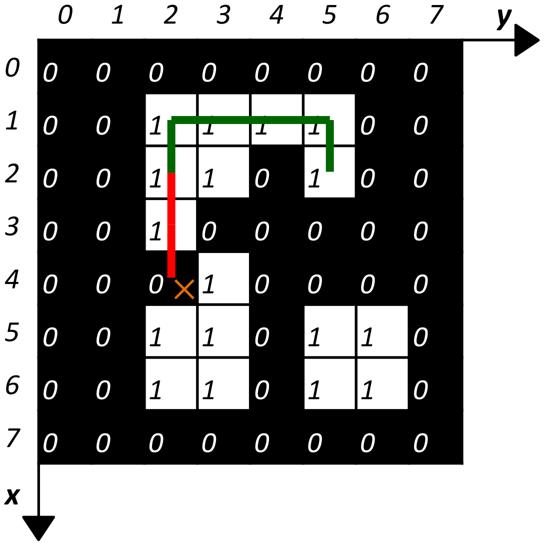
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:



Digital path (or curve)

Considerando vizinhança-4:

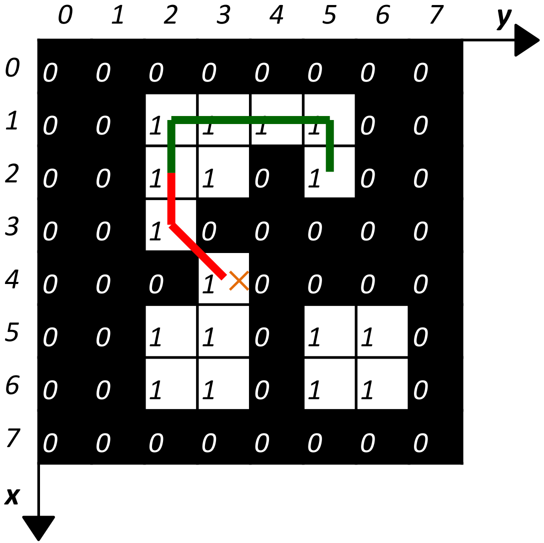
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:



Digital path (or curve)

Considerando vizinhança-4:

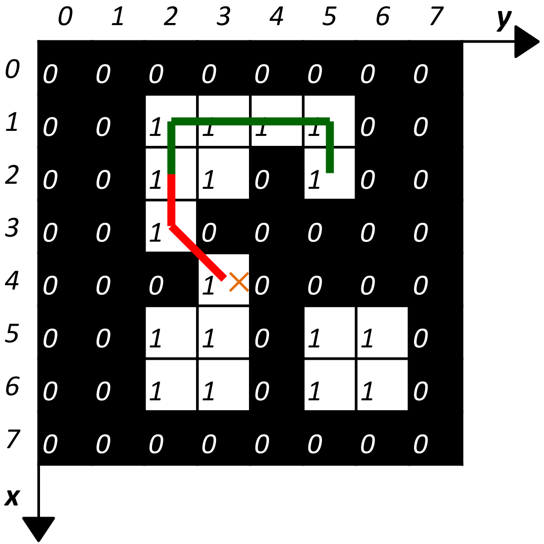
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!



Digital path (or curve)

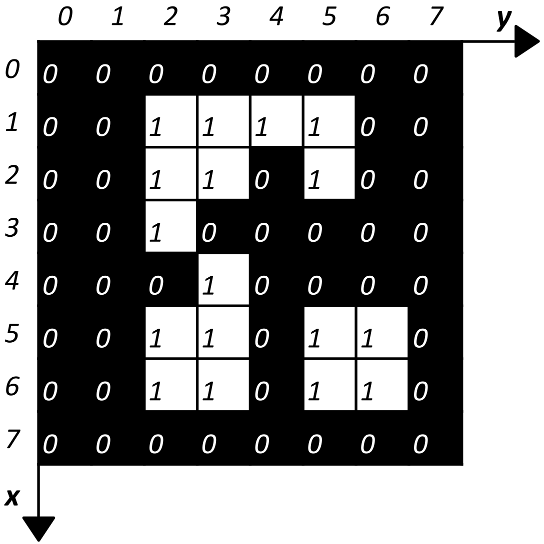
Considerando vizinhança-4:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!



Considerando vizinhança-8:

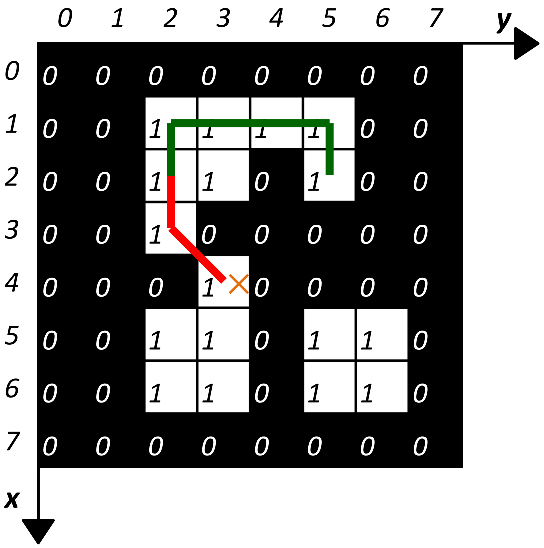
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:



Digital path (or curve)

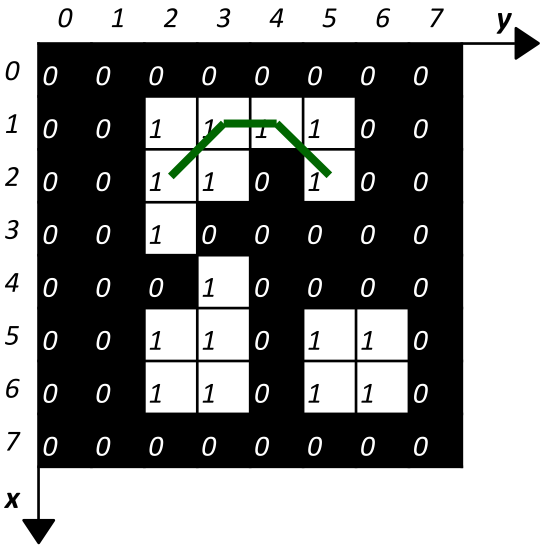
Considerando vizinhança-4:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!



Considerando vizinhança-8:

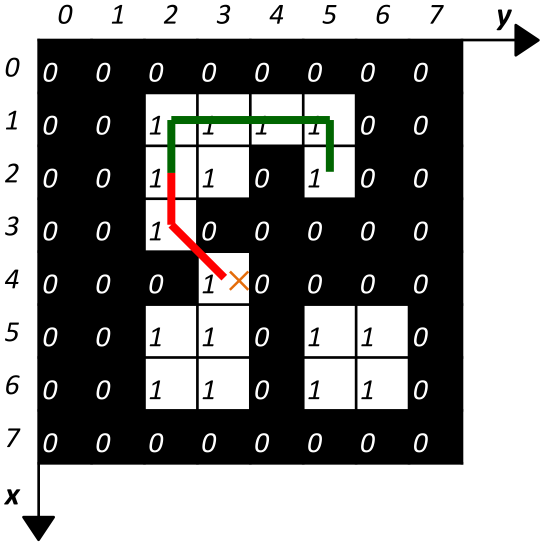
- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,4), (1,3), (2,2).*



Digital path (or curve)

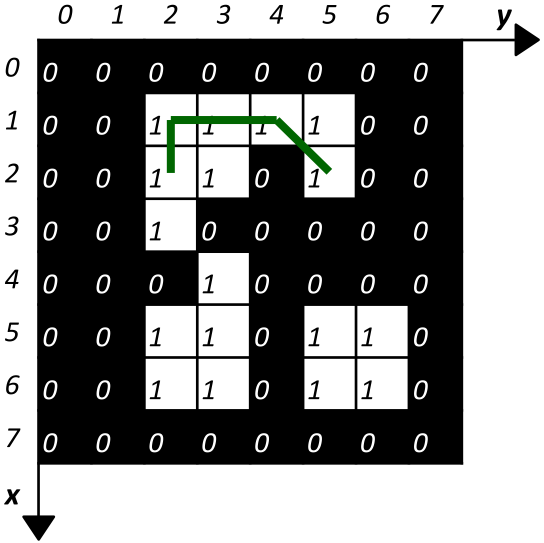
Considerando vizinhança-4:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!



Considerando vizinhança-8:

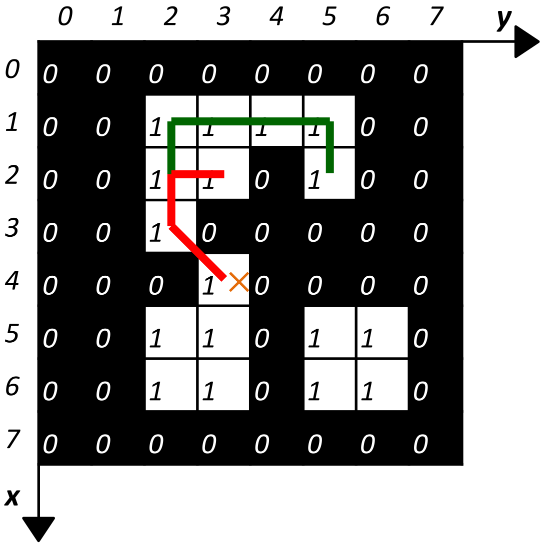
- Outro caminho entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,4), (1,3), (1,2), (2,2).*



Digital path (or curve)

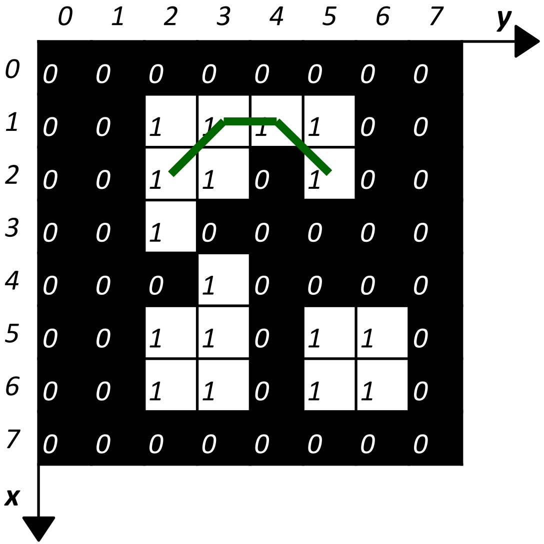
Considerando vizinhança-4:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!



Considerando vizinhança-8:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,4), (1,3), (2,2).*
- Um dos caminhos entre *p* em *(2,3)* e *q* em *(6,2)*:



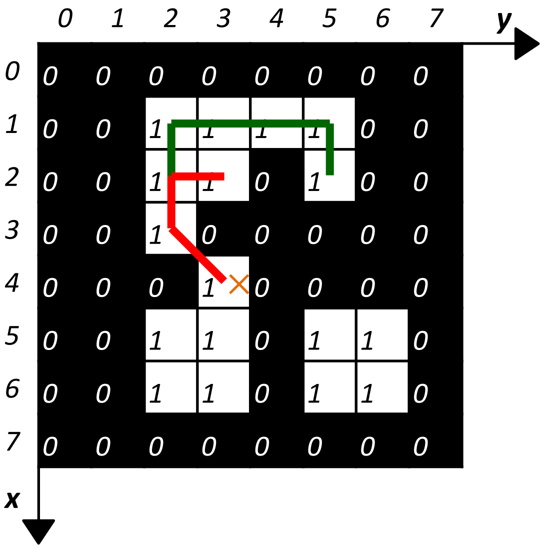
Digital path (or curve)

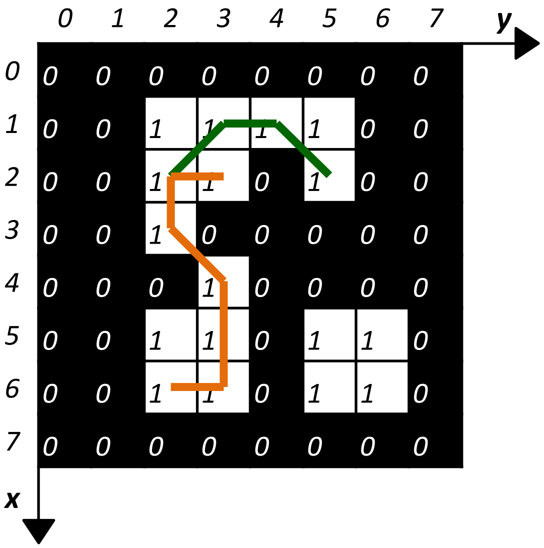
Considerando vizinhança-4:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!

Considerando vizinhança-8:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,4), (1,3), (2,2).*
- Um dos caminhos entre *p* em *(2,3)* e *q* em *(6,2)*:
  - *(2,3), (2,2), (3,2), (4,3), (5,3), (6,3), (6,2).*





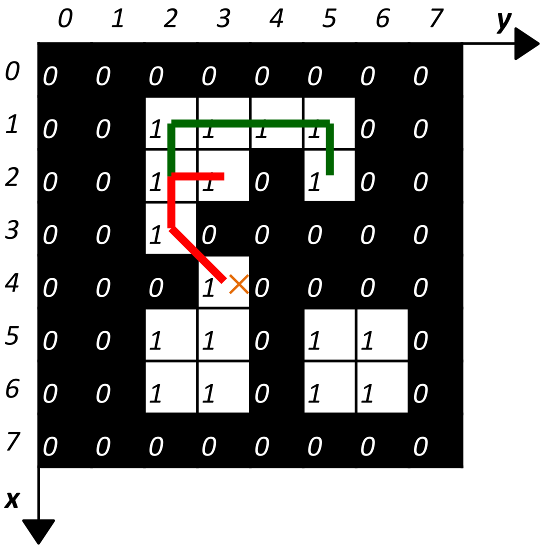
Digital path (or curve)

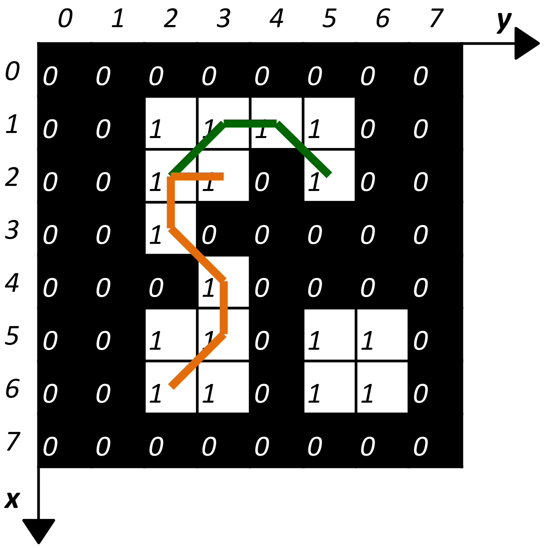
Considerando vizinhança-4:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,5), (1,4), (1,3), (1,2), (2,2).*
- Um caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - Não existe um caminho!

Considerando vizinhança-8:

- Um dos caminhos entre *p* em *(2,5)* e *q* em *(2,2)*:
  - *(2,5), (1,4), (1,3), (2,2).*
- Outro caminho entre *p* em *(2,3)* e *q* em *(6,2)*:
  - *(2,3), (2,2), (3,2), (4,3), (5,3), (6,2).*

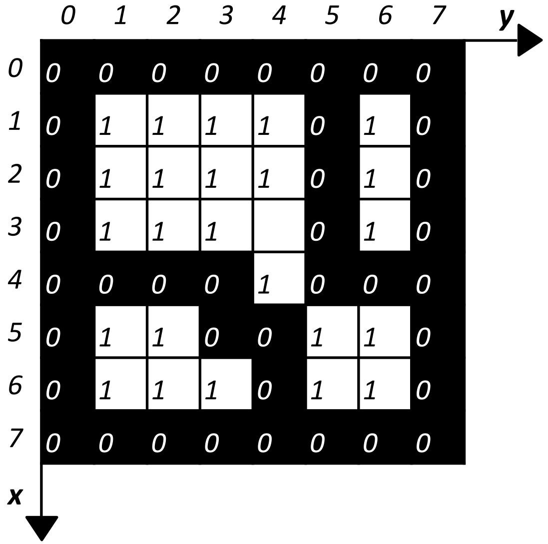




Connected regions and connected components

- **Região conectada:**
  - Qualquer região R que existe pelo menos um caminho entre quaisquer pares de pixels (p, q)

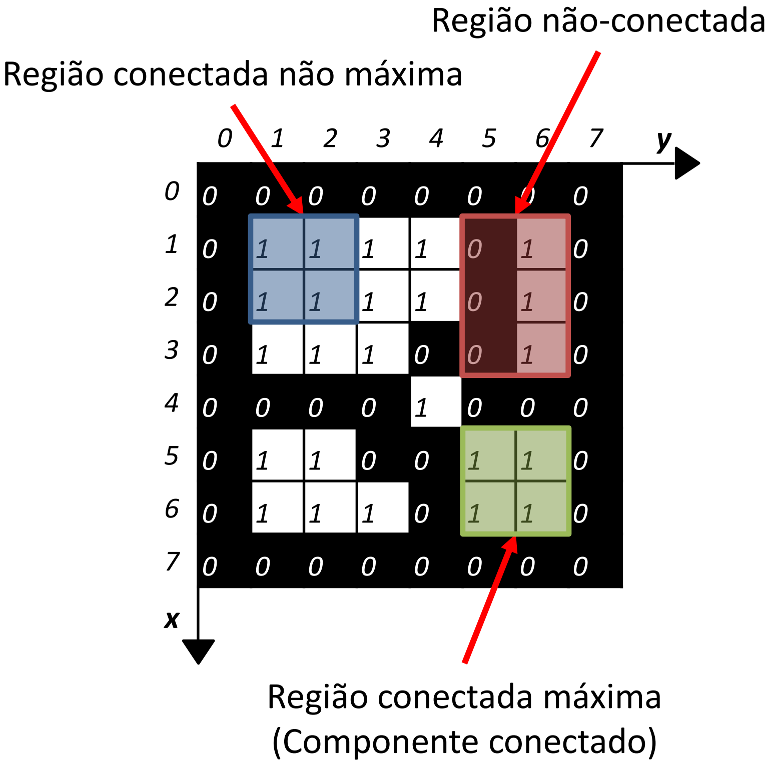
- **Componente conectado:**
  - Região conectada máxima
  - Não é um subconjunto próprio de nenhuma região conectada maior



Connected regions and connected components

- **Connected region:**
  - Qualquer região R que existe pelo menos um caminho entre quaisquer pares de pixels *(p, q)*

- **Connected component:**
  - Região conectada máxima
  - Não é um subconjunto próprio de nenhuma região conectada maior



Connected components

Binary image

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

Connected components

Binary image

N$_4$(p)

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

N$_8$(p)

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Connected components

Binary image

$N_4(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

$N_8(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Connected components

Binary image

N$_4$(p)

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

N$_8$(p)

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Connected components

Binary image

N$_4$(p)

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

N$_8$(p)

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Connected components

	Binary image	Label image

$N_4(p)$

Binary image

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Label image

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	5	0
2	0	1	1	1	1	0	5	0
3	0	1	1	1	0	0	5	0
4	0	0	0	0	3	0	0	0
5	0	2	2	0	0	4	4	0
6	0	2	2	2	0	4	4	0
7	0	0	0	0	0	0	0	0

$N_8(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Connected components

Binary image

Label image

$N_4(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	5	0	
2	0	1	1	1	1	0	5	0	
3	0	1	1	1	0	0	5	0	
4	0	0	0	0	3	0	0	0	
5	0	2	2	0	0	4	4	0	
6	0	2	2	2	0	4	4	0	
7	0	0	0	0	0	0	0	0	
x									

$N_8(p)$

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	1	0	
2	0	1	1	1	1	0	1	0	
3	0	1	1	1	0	0	1	0	
4	0	0	0	0	1	0	0	0	
5	0	1	1	0	0	1	1	0	
6	0	1	1	1	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

	0	1	2	3	4	5	6	7	y
0	0	0	0	0	0	0	0	0	
1	0	1	1	1	1	0	3	0	
2	0	1	1	1	1	0	3	0	
3	0	1	1	1	0	0	3	0	
4	0	0	0	0	1	0	0	0	
5	0	2	2	0	0	1	1	0	
6	0	2	2	2	0	1	1	0	
7	0	0	0	0	0	0	0	0	
x									

Objects and background in a image

- Image foreground (objects)
  - Conjunto de todos os componentes conectados na imagem
- Image background
  - O complemento do conjunto dos componentes conectados

$N_4(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Objects and background in a image

- Image foreground (objects)
  - Conjunto de todos os componentes conectados na imagem
- Image background
  - O complemento do conjunto dos componentes conectados

$N_4(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0
2	0	1	1	1	1	0	1	0
3	0	1	1	1	0	0	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	1	1	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

Borda, contorno ou fronteira

- Borda de um componente conectado C:
  - Conjunto de pontos em C que são **adjacentes** aos pontos do complemento de C.
  - Dependente da conectividade.
  - Borda interna.

$N_4(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0
2	0	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	0	0	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

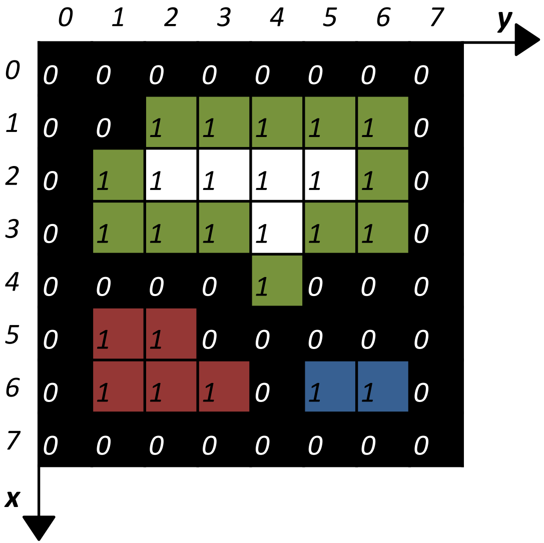
$N_8(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0
2	0	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	0	0	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

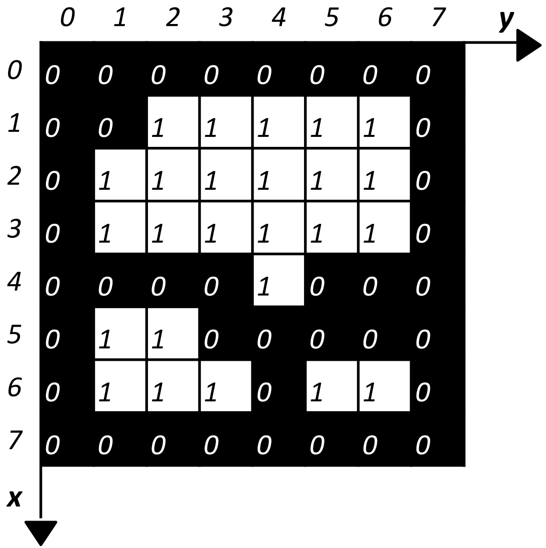
Boundary, border, contour, or frontier

- Borda de um componente conectado C:
  - Conjunto de pontos em C que são **adjacentes** aos pontos do complemento de C.
  - Dependente da conectividade.
  - Borda interna.

$N_4(p)$

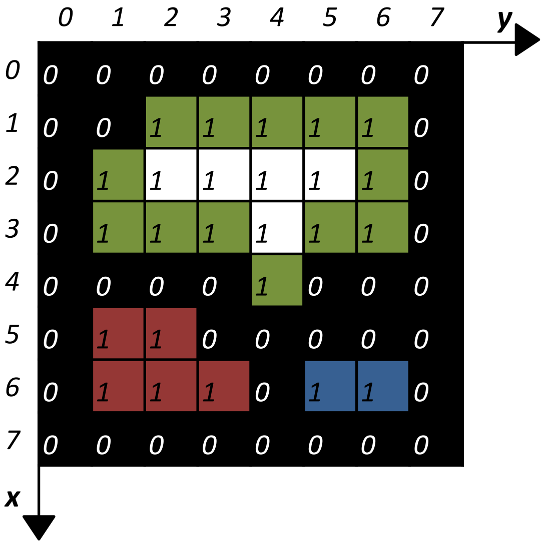


$N_8(p)$

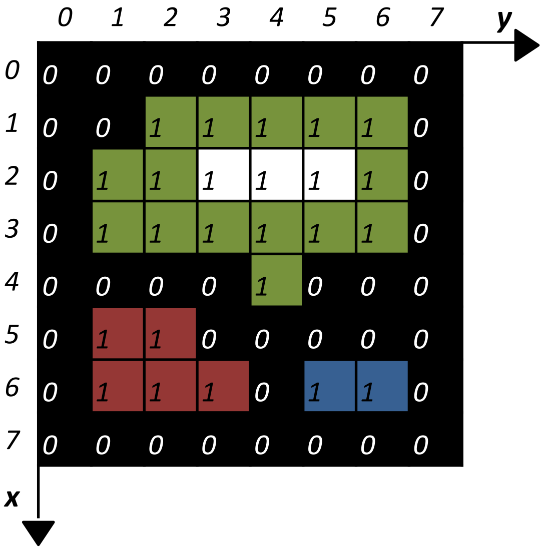


- Borda de um componente conectado C:
  - Conjunto de pontos em C que são **adjacentes** aos pontos do complemento de C.
  - Dependente da conectividade.
  - Borda interna.

$N_4(p)$



$N_8(p)$



Boundary, border, contour, or frontier

- Borda **externas** de um componente conectado C:
  - Conjunto de pontos no complemento de C, C^c, que são **adjacentes** aos pontos em C.
  - Bordas sempre formam um conjunto fechado.
    - Algoritmos seguidores de contorno.

$N_4(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0
2	0	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	0	0	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

$N_8(p)$

x \ y	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	0
2	0	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	0
4	0	0	0	0	1	0	0	0
5	0	1	1	0	0	0	0	0
6	0	1	1	1	0	1	1	0
7	0	0	0	0	0	0	0	0

LOGICAL AND ARITHMETIC OPERATIONS

Arithmetic operations

- Operações aritméticas são realizadas entre pixels correspondentes
  - SUM
    - $g(x, y) = f_1(x,y) + f_2(x,y)$
  - SUBTRACTION
    - $g(x, y) = f_1(x,y) - f_2(x,y)$
  - MULTIPLICATION
    - $g(x, y) = f_1(x,y) \times f_2(x,y)$
  - DIVISION
    - $g(x, y) = f_1(x,y) / f_2(x,y)$

Python data types (scikit-image)

dtype	from	until	Description
uint8	0	255	Inteiro de 8 bits sem sinal
uint16	0	65,535	Inteiro de 16 bits sem sinal
uint32	0	4,294,967,295	Inteiro de 32 bits sem sinal
float	-1.0	+1.0	Ponto flutuante de 64 bits
int8	-128	127	Inteiro de 8 bits com sinal
int16	-32,768	+32,767	Inteiro de 16 bits com sinal
int32	-2^{31}	$2^{31} - 1$	Inteiro de 32 bits com sinal

Funtion	Description
img_as_float	Converte para float
img_as_ubyte	Converte para uint8
img_as_uint	Converte para uint16
img_as_int	Converte para int16

https://scikit-image.org/docs/dev/user_guide/data_types.html

Arithmetic operations

SUM

k = 3 (number of bits)
$L = 2^k = 2^3 = 8$
Range: [0, L-1] or [0, 7]

$f_1(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	3	4	4
2	0	3	3	4	4
3	2	2	1	1	1
4	2	2	2	1	1

+

$f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

=

$g(x,y) = f_1(x,y) + f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	6	**8**	**9**	4
2	3	6	**8**	**9**	4
3	3	4	2	3	2
4	4	3	4	2	3

Truncamento:

$$g'(x,y) = \min(g(x,y), L-1)$$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	6	**7**	**7**	4
2	3	6	**7**	**7**	4
3	3	4	2	3	2
4	4	3	4	2	3

Normalização:

$$g' = \frac{L-1}{g_{max} - g_{min}} \times (g - g_{min})$$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	2	5	**6**	**7**	3
2	2	5	**6**	**7**	3
3	2	3	2	2	2
4	3	2	3	2	2

g	p / 9 * 7	p'
0	0.00	0
1	0.77	1
2	1.55	2
3	2.33	2
4	3.11	3
5	3.88	4
6	4.66	5
7	5.44	5
8	6.22	6
9	7.00	7

Wrap-around:

$$g(x,y) > L-1 \; ? \; g(x,y) - L : g(x,y)$$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	6	**0**	**1**	4
2	3	6	**0**	**1**	4
3	3	4	2	3	2
4	4	3	4	2	3

Arithmetic operations

SUBTRACTION

k = 3 (number of bits)
$L = 2^k = 2^3 = 8$
Range: [*0, L-1*] or [*0, 7*]

$f_1(x, y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

−

$f_2(x, y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	3	4	4
2	0	3	3	4	4
3	2	2	1	1	1
4	2	2	2	1	1

=

$g(x, y) = f_1(x, y) - f_2(x, y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	0	2	1	-4
2	3	0	2	1	-4
3	-1	0	0	1	0
4	0	-1	0	0	1

Truncamento:

$$g'(x, y) = \max(g(x, y), 0)$$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	0	2	1	0
2	3	0	2	1	0
3	0	0	0	1	0
4	0	0	0	0	1

Normalização:

$$g' = \frac{L - 1}{g_{max} - g_{min}} \times (g - g_{min})$$

x \ y	0	1	2	3	4
0	4	4	4	4	4
1	7	4	6	5	0
2	7	4	6	5	0
3	3	4	4	5	4
4	4	3	4	4	5

Valor absoluto:

$$g'(x, y) = |g(x, y)|$$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	0	2	1	4
2	3	0	2	1	4
3	1	0	0	1	0
4	0	1	0	0	1

Wrap-around:

$$g(x, y) < 0 \; ? \; L + g(x, y) \; : \; g(x, y)$$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	0	2	1	4
2	3	0	2	1	4
3	7	0	0	1	0
4	0	7	0	0	1

Arithmetic operations

MULTIPLICATION

k = 3 (number of bits)
L = $2^k = 2^3 = 8$
Range: [0, L-1] or [0, 7]

$f_1(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

X

$f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	3	4	4
2	0	3	3	4	4
3	2	2	1	1	1
4	2	2	2	1	1

=

$g(x,y) = f_1(x,y) \times f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	9	15	25	0
2	0	9	15	25	0
3	2	4	1	4	1
4	4	2	4	2	2

(*) Corrigir como fizemos com a SOMA e SUBTRAÇÃO

MULTIPLICATION

Mascaramento

$f_1(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

X

$f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	0	0
2	0	1	1	1	0
3	0	1	1	1	0
4	0	0	0	0	0

=

$g(x,y) = f_1(x,y) \times f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	5	0	0
2	0	3	5	5	0
3	0	2	1	2	0
4	0	0	0	0	0

Arithmetic operations

DIVISION

k = 3 (number of bits)
$L = 2^k = 2^3 = 8$
Range: [*0, L-1*] or [*0, 7*]

$f_1(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

/

$f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	3	4	4
2	0	3	3	4	4
3	2	2	1	1	1
4	2	2	2	1	1

=

$g(x,y) = f_1(x,y)/f_2(x,y)$

x \ y	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	1.	1.66	1,25	0
2	∞	1.	1.66	1.25	0
3	0.5	1.	1.	1.	1.
4	1.	0.5	1.	1.	2.

Arithmetic operations

DIVISION

k = 3 (number of bits)
$L = 2^k = 2^3 = 8$
Range: [*0, L-1*] or [*0, 7*]

$f_1(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

/

$f_2(x,y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	3	4	4
2	0	3	3	4	4
3	2	2	1	1	1
4	2	2	2	1	1

=

$g(x,y) = f_1(x,y)/f_2(x,y)$

x \ y	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	1.	1.66	1,25	0
2	∞	1.	1.66	1.25	0
3	0.5	1.	1.	1.	1.
4	1.	0.5	1.	1.	2.

Division by zero

Converter para float
Substituir o 0 (zero) pelo menor valor positivo.
ε = np.spacing(1)

$f_2(x,y)'$

x \ y	0	1	2	3	4
0	ε	ε	ε	ε	ε
1	ε	3.	3.	4.	4.
2	ε	3.	3.	4.	4.
3	2.	2.	1.	1.	1.
4	2.	2.	2.	1.	1.

Chityala, R; Pudipeddi, P. Image Processing and Acquisition using Python. CRC Press, 2014.

Arithmetic operations

UFV Universidade Federal de Viçosa

DIVISION

k = 3 (number of bits)
$L = 2^k = 2^3 = 8$
Range: [*0*, *L*-*1*] or [*0*, *7*]

$f_1(x, y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	3	3	5	5	0
2	3	3	5	5	0
3	1	2	1	2	1
4	2	1	2	1	2

/

$f_2(x, y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	3	3	4	4
2	0	3	3	4	4
3	2	2	1	1	1
4	2	2	2	1	1

=

$g(x, y) = f_1(x, y)/f_2(x, y)$

x \ y	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	1.	1.66	1,25	0
2	∞	1.	1.66	1.25	0
3	0.5	1.	1.	1.	1.
4	1.	0.5	1.	1.	2.

Division by zero

Converter para float
Substituir o 0 (zero) por um valor positivo muito pequeno.
ε = np.spacing(1)

$f_2(x, y)'$

x \ y	0	1	2	3	4
0	ε	ε	ε	ε	ε
1	ε	3.	3.	4.	4.
2	ε	3.	3.	4.	4.
3	2.	2.	1.	1.	1.
4	2.	2.	2.	1.	1.

Converter o resultado para inteiro (arredondar ou truncar).
Tratar valores.

$g(x, y)' = f_1(x, y)/f_2(x, y)$

x \ y	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	1	0
2	0	1	2	1	0
3	1	1	1	1	1
4	1	1	1	1	2

Chityala, R; Pudipeddi, P. Image Processing and Acquisition using Python. CRC Press, 2014.

Logical operations

- Logical operations occur between binary images
  - Pixels == 0 → False
  - Pixels == 1 → True

A	B	NOT A	A AND B	A OR B	A NAND B	A NOR B	A XOR B
0	0	1	0	0	1	0	0
0	1	1	0	1	1	0	1
1	0	0	0	1	1	0	1
1	1	0	1	1	0	1	0

Logical operations

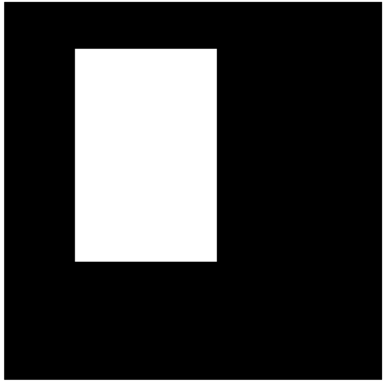


Imagem A

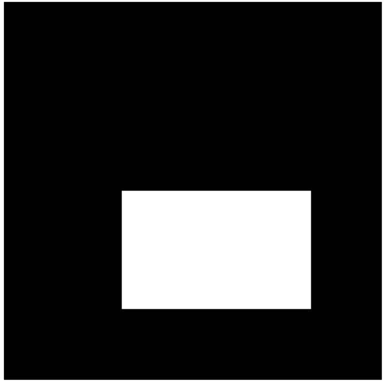
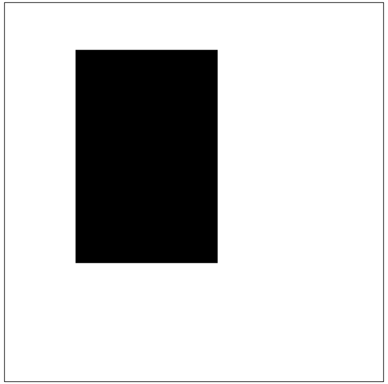
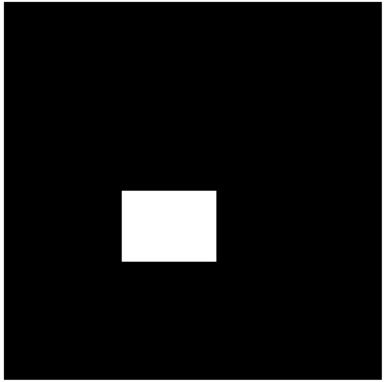


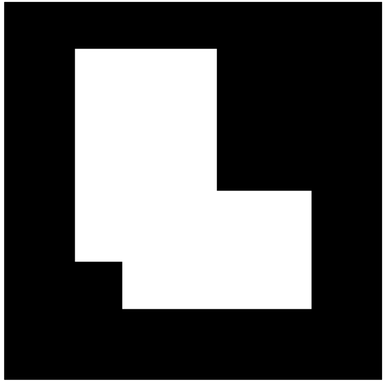
Imagem B



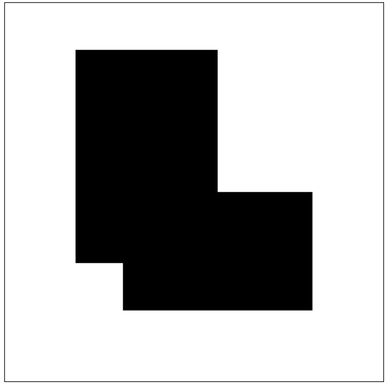
NOT A



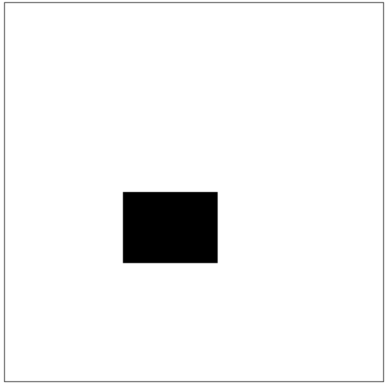
A AND B



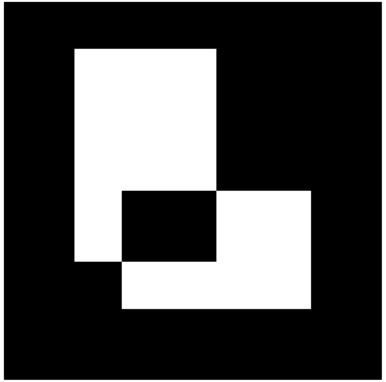
A OR B



A NOR B



A NAND B



A XOR B

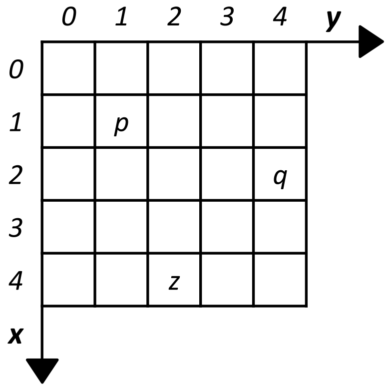
DISTANCE MEASURES

Distance measures

- Consider three pixels and their respective coordinates
  - *p* in *(x, y)*, *q* in *(s, t), and z* in *(v, w)*

- *D* is a function or distance measure
  - $D(p, q) \geq 0$
    - $D(p, q) = 0$ *if* $p = q$
  - $D(p, q) = D(q, p)$
  - $D(p, z) \leq D(p, q) + D(q, z)$

- Some distance measures:
  - Euclidian distance
  - City block distance
  - Chessboard distance



Distance measures

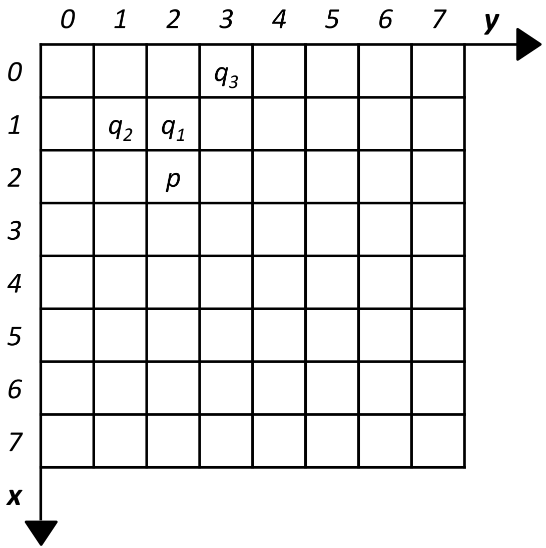
- The Euclidean distance between pixel *p* in *(x, y)* and pixel *q* in *(s, t)*:
  - $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

	0	1	2	3	4	**y**
0	√8	√5	2	√5	√8	
1	√5	√2	1	√2	√5	
2	2	1	0	1	2	
3	√5	√2	1	√2	√5	
4	√8	√5	2	√5	√8	
x						

Distance measures

- The Euclidean distance between pixel *p* in *(x, y)* and pixel *q* in *(s, t)*:
  - $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

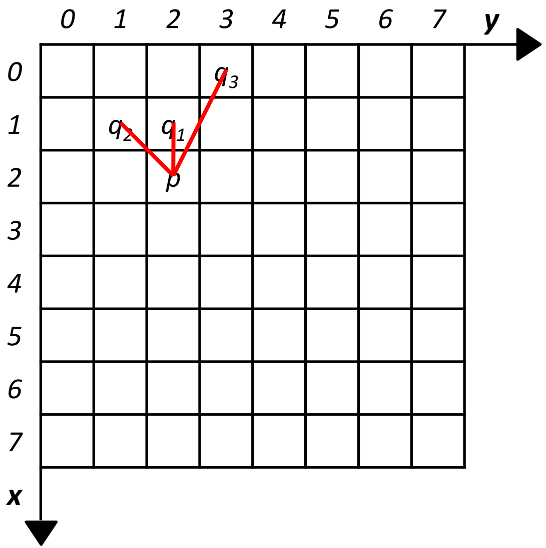
- For *p* with coordinates *(2, 2), and*
  - q_1 with coordinates *(1, 2)*:
    - $D_e(p,q) = \sqrt{(2-1)^2 + (2-2)^2}$
    - $D_e(p,q) = \sqrt{1^2 + 0^2}$
    - $D_e(p,q) = \sqrt{1} = 1$
  - q_2 with coordinates *(1,1)*:
    - $D_e(p,q) = \sqrt{(2-1)^2 + (2-1)^2}$
    - $D_e(p,q) = \sqrt{1^2 + 1^2} = \sqrt{2}$
  - q_3 with coordinates *(0,3)*:
    - $D_e(p,q) = \sqrt{(2-0)^2 + (2-3)^2}$
    - $D_e(p,q) = \sqrt{2^2 + (-1)^2} = \sqrt{5}$



Distance measures

- The Euclidean distance between pixel *p* in *(x, y)* and pixel *q* in *(s, t)*:
  - $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

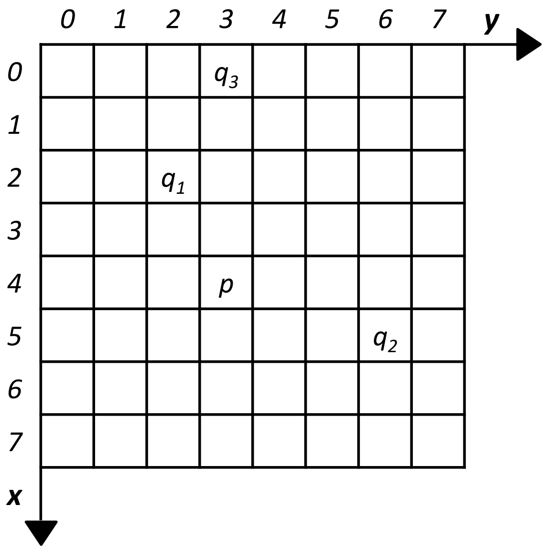
- For *p* with coordinates *(2, 2), and*
  - q_1 with coordinates *(1, 2)*:
    - $D_e(p,q) = \sqrt{(2-1)^2 + (2-2)^2}$
    - $D_e(p,q) = \sqrt{1^2 + 0^2}$
    - $D_e(p,q) = \sqrt{1} = 1$
  - q_2 with coordinates *(1,1)*:
    - $D_e(p,q) = \sqrt{(2-1)^2 + (2-1)^2}$
    - $D_e(p,q) = \sqrt{1^2 + 1^2} = \sqrt{2}$
  - q_3 with coordinates *(0,3)*:
    - $D_e(p,q) = \sqrt{(2-0)^2 + (2-3)^2}$
    - $D_e(p,q) = \sqrt{2^2 + (-1)^2} = \sqrt{5}$



Distance measures

- The Euclidean distance between pixel *p* in *(x, y)* and pixel *q* in *(s, t)*:
  - $D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$

- Para *p* com coordenadas *(4, 3)* e:
  - q_1 com coordenadas *(2, 2)*:
    - $D_e(p, q) = \sqrt{(4-2)^2 + (3-2)^2}$
    - $D_e(p, q) = \sqrt{2^2 + 1^2} = \sqrt{5}$
  - q_2 com coordenadas *(5, 6)*:
    - $D_e(p, q) = \sqrt{(4-5)^2 + (3-6)^2}$
    - $D_e(p, q) = \sqrt{(-1)^2 + (-3)^2} = \sqrt{1+9} = \sqrt{10}$
  - q_3 com coordenadas *(0, 3)*:
    - $D_e(p, q) = \sqrt{(4-0)^2 + (3-3)^2}$
    - $D_e(p, q) = \sqrt{4^2 + 0^2} = \sqrt{16} = 4$

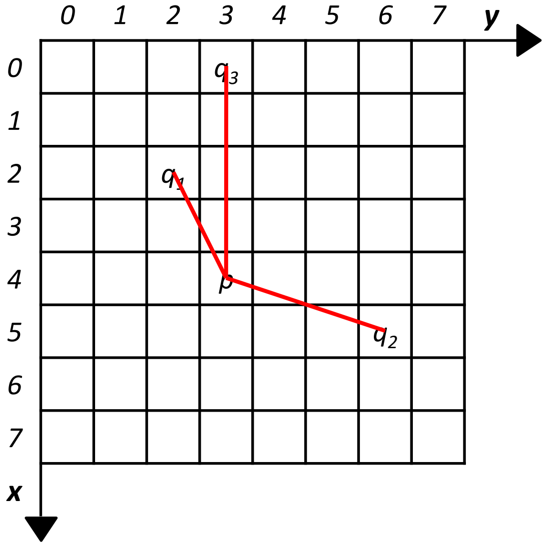


Medidas de distância

UFV Universidade Federal de Viçosa

- The Euclidean distance between pixel *p* in *(x, y)* and pixel *q* in *(s, t)*:
  - $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

- Para *p* com coordenadas *(4, 3)* e:
  - q_1 com coordenadas *(2, 2)*:
    - $D_e(p,q) = \sqrt{(4-2)^2 + (3-2)^2}$
    - $D_e(p,q) = \sqrt{2^2 + 1^2} = \sqrt{5}$
  - q_2 com coordenadas *(5, 6)*:
    - $D_e(p,q) = \sqrt{(4-5)^2 + (3-6)^2}$
    - $D_e(p,q) = \sqrt{(-1)^2 + (-3)^2} = \sqrt{1+9} = \sqrt{10}$
  - q_3 com coordenadas *(0, 3)*:
    - $D_e(p,q) = \sqrt{(4-0)^2 + (3-3)^2}$
    - $D_e(p,q) = \sqrt{4^2 + 0^2} = \sqrt{16} = 4$



Distance measures

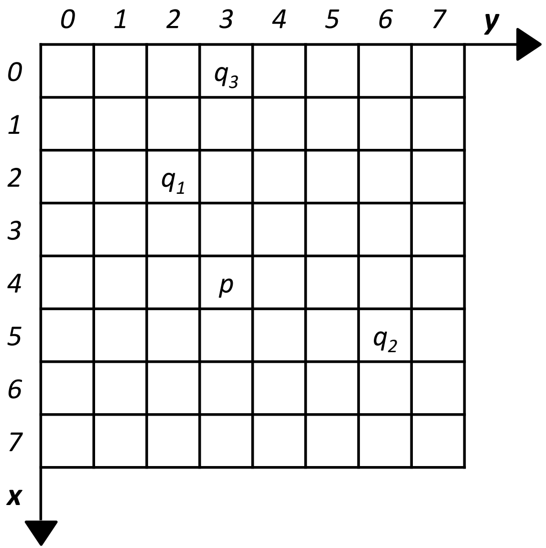
- *City block* distance between p in (x, y) and q in (s, t)
  - $D_4(p, q) = |x - s| + |y - t|$

$x \backslash y$	0	1	2	3	4
0	4	3	2	3	4
1	3	2	1	2	3
2	2	1	0	1	2
3	3	2	1	2	3
4	4	3	2	3	4

Distance measures

- *City block* distance between *p* in (*x, y*) and *q* in (*s, t*)
  - $D_4(p, q) = |x - s| + |y - t|$

- Para *p* com coordenadas *(4, 3)* e:
  - q_1 com coordenadas *(2, 2)*:
    - $D_4(p, q) = |4 - 2| + |3 - 2|$
    - $D_4(p, q) = 2 + 1 = 3$
  - q_2 com coordenadas *(5, 6)*:
    - $D_4(p, q) = |4 - 5| + |3 - 6|$
    - $D_4(p, q) = 1 + 3 = 4$
  - q_3 com coordenadas *(0, 3)*:
    - $D_4(p, q) = |4 - 0| + |3 - 3|$
    - $D_4(p, q) = 4 + 0 = 4$

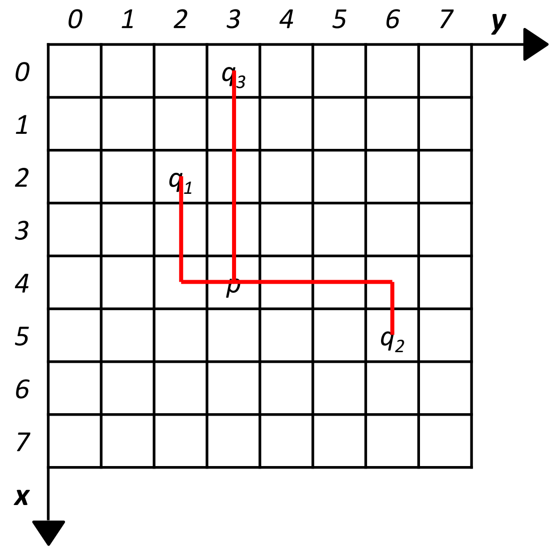


Distance measures

UFV
Universidade Federal de Viçosa

- *City block* distance between *p* in (*x, y*) and *q* in (*s, t*)
  - $D_4(p, q) = |x - s| + |y - t|$

- Para *p* com coordenadas *(4, 3)* e:
  - q_1 com coordenadas *(2, 2)*:
    - $D_4(p, q) = |4 - 2| + |3 - 2|$
    - $D_4(p, q) = 2 + 1 = 3$
  - q_2 com coordenadas *(5, 6)*:
    - $D_4(p, q) = |4 - 5| + |3 - 6|$
    - $D_4(p, q) = 1 + 3 = 4$
  - q_3 com coordenadas *(0, 3)*:
    - $D_4(p, q) = |4 - 0| + |3 - 3|$
    - $D_4(p, q) = 4 + 0 = 4$



Distance measures

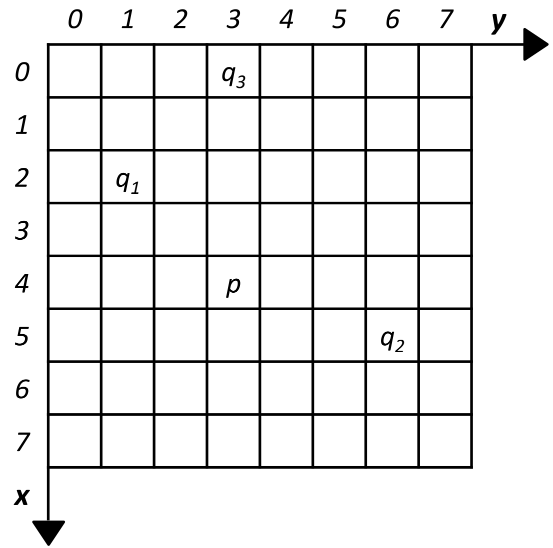
- *Chessboard* distance between *p* in (*x, y*) and *q* in (*s, t*)
  - $D_8(p, q) = max(|x - s|, |y - t|)$

	0	1	2	3	4	**y**
0	2	2	2	2	2	
1	2	1	1	1	2	
2	2	1	0	1	2	
3	2	1	1	1	2	
4	2	2	2	2	2	
x						

Distance measures

- *Chessboard* distance between *p* in (*x, y*) and *q* in (*s, t*)
  - $D_8(p, q) = max(|x - s|, |y - t|)$

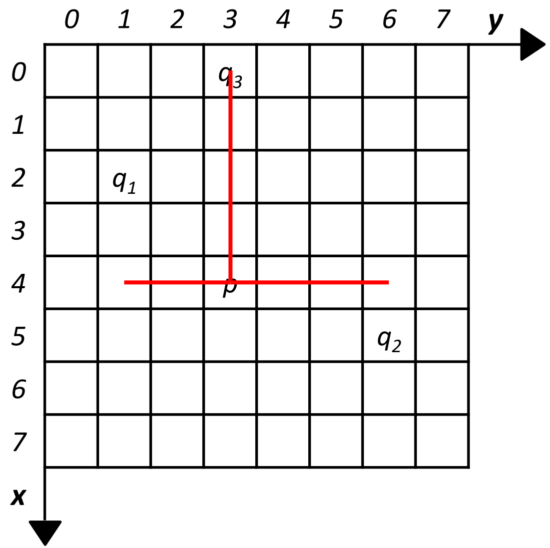
- Para *p* com coordenadas *(4, 3)* e:
  - q_1 com coordenadas *(2, 1)*:
    - $D_8(p, q) = \max(|4 - 2|, |3 - 1|)$
    - $D_8(p, q) = \max(2, 2) = 2$
  - q_2 com coordenadas *(5, 6)*:
    - $D_8(p, q) = \max(|4 - 5|, |3 - 6|)$
    - $D_8(p, q) = \max(1, 3) = 3$
  - q_3 com coordenadas *(0, 3)*:
    - $D_8(p, q) = \max(|4 - 0|, |3 - 3|)$
    - $D_8(p, q) = \max(4, 0) = 4$



Distance measures

- *Chessboard* distance between *p* in (*x, y*) and *q* in (*s, t*)
  - $D_8(p, q) = max(|x-s|, |y-t|)$

- Para *p* com coordenadas *(4, 3)* e:
  - q_1 com coordenadas *(2, 1)*:
    - $D_8(p,q) = \max(|4-2|,|3-1|)$
    - $D_8(p,q) = \max(2,2) = 2$
  - q_2 com coordenadas *(5, 6)*:
    - $D_8(p,q) = \max(|4-5|,|3-6|)$
    - $D_8(p,q) = \max(1,3) = 3$
  - q_3 com coordenadas *(0, 3)*:
    - $D_8(p,q) = \max(|4-0|,|3-3|)$
    - $D_8(p,q) = \max(4,0) = 4$



Bibliography

- GONZALEZ, R.C.; WOODS, R.E. **Digital Image Processing.** 3rd ed. Pearson, 2007.
  - Sections 2.5, and 2.6 (until 2.3.3)

- MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento digital de imagens**. Brasport, 1999.
  - (*in Brazilian Portuguese*)
  - Available on the author's website (for personal use only)
  - http://dainf.ct.utfpr.edu.br/~hvieir/pub.html
  - Section 2.2, and 2.3 (except 2.3.3)

- J. E. R. Queiroz, H. M. Gomes. **Introdução ao Processamento Digital de Imagens**. RITA. v. 13, 2006.
  - (*in Brazilian Portuguese*)
  - http://www.dsc.ufcg.edu.br/~hmg/disciplinas/graduacao/vc-2016.2/Rita-Tutorial-PDI.pdf
  - Section 3

Complementary bibliography

- scikit-image. **Image data types and what they mean.**
  - https://scikit-image.org/docs/dev/user_guide/data_types.html
- Chityala, R; Pudipeddi, P. **Image Processing and Acquisition using Python**. CRC Press, 2014.

```
@misc{mari_im_proc_2023,
	author = {João Fernando Mari},
	title = {Fundamentals of digital imaging II},
	year = {2023},
	publisher = {GitHub},
	journal = {Introduction to digital image processing - UFV},
	howpublished = {\url{https://github.com/joaofmari/SIN392_Introduction-to-digital-image-processing_2023}}
}
```

THE END