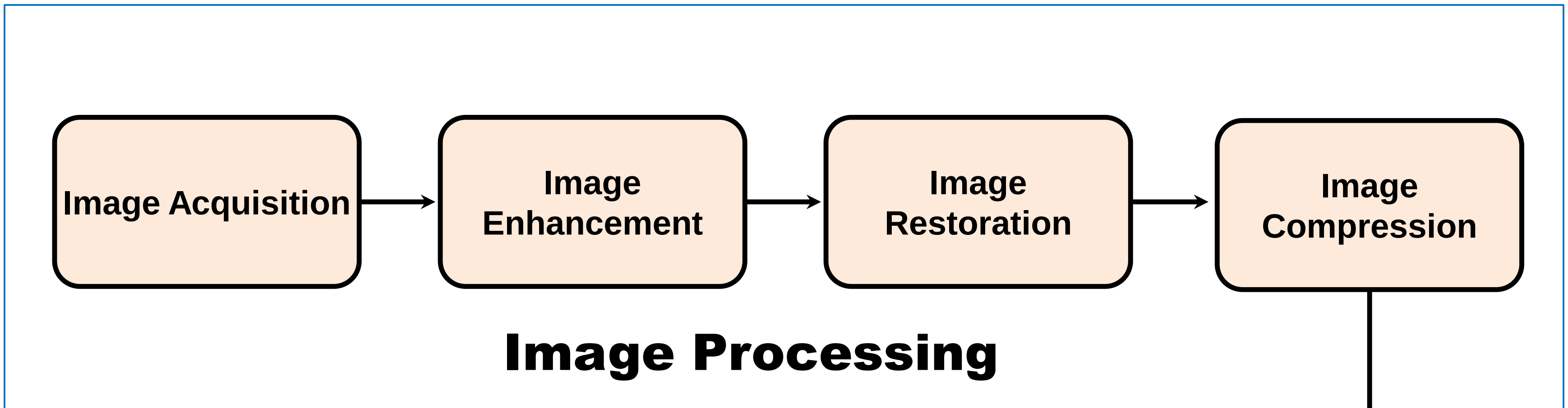
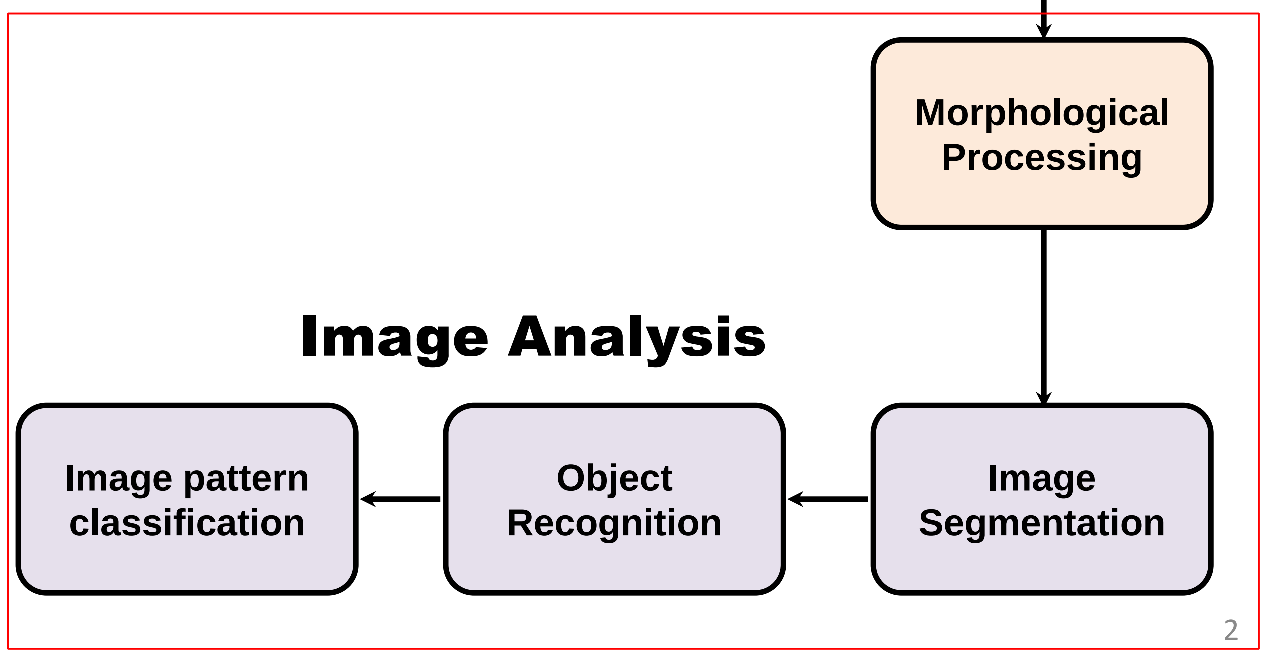
CS370 Computer Imaging

Image Representation and Operations Part-1

Key Stages in DIP





Computer Vision (making sense)

Recap

- Goal of Image Processing
- Human Visual System
- Image Acquisition
- Digital Image Representation
- Image Sampling And Quantisation
- Image Resolution

Lecture Objective

- Pixel Neighborhood
- Adjacency
- Digital Path
- Connectivity
- Region and Boundary
- Proximity Relationship
- Defining Linear Operations

False Contouring

- <u>Insufficient</u> number of intensity levels in <u>smooth areas</u> of a digital image

k=8

K=4

False Contouring

256

128

64

32

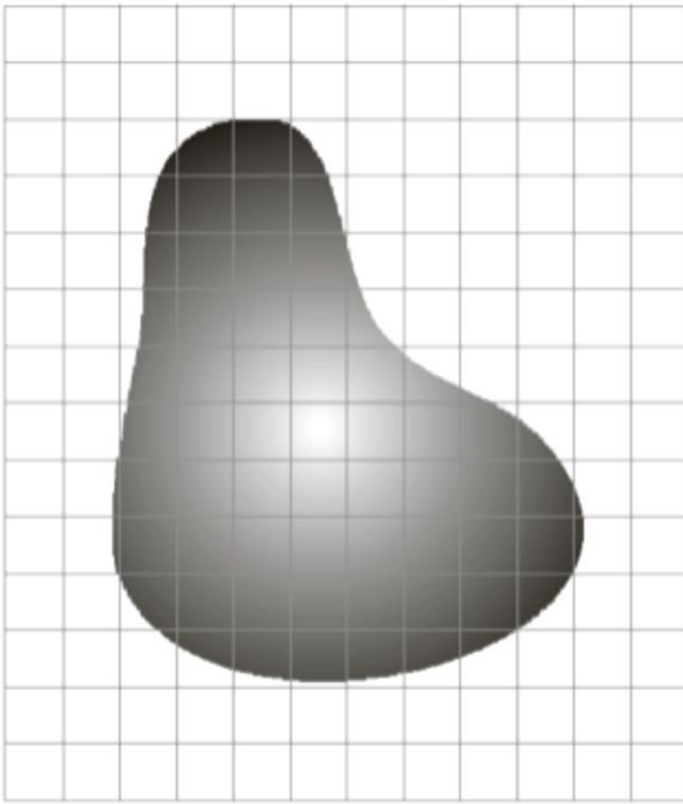
16

8

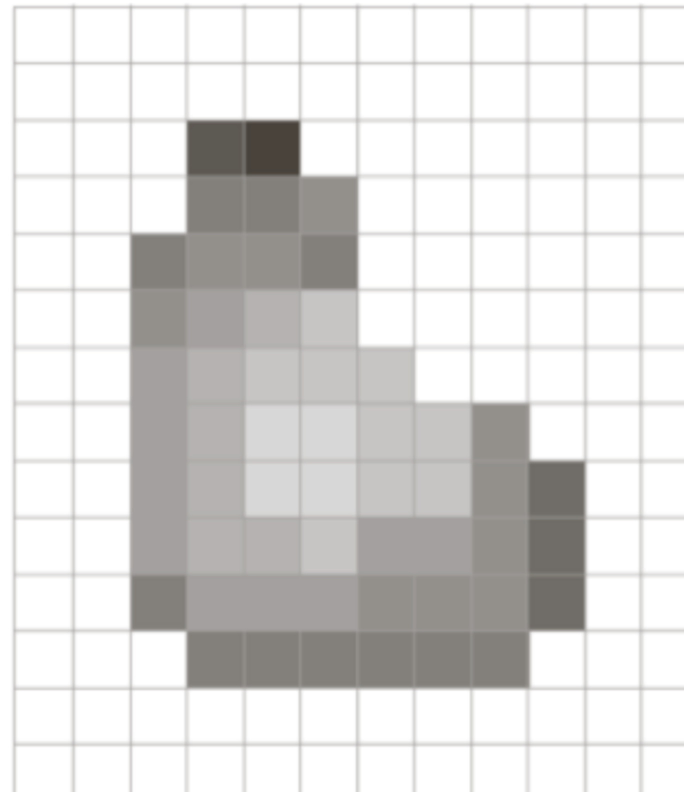
4

2

From Analog to Digital



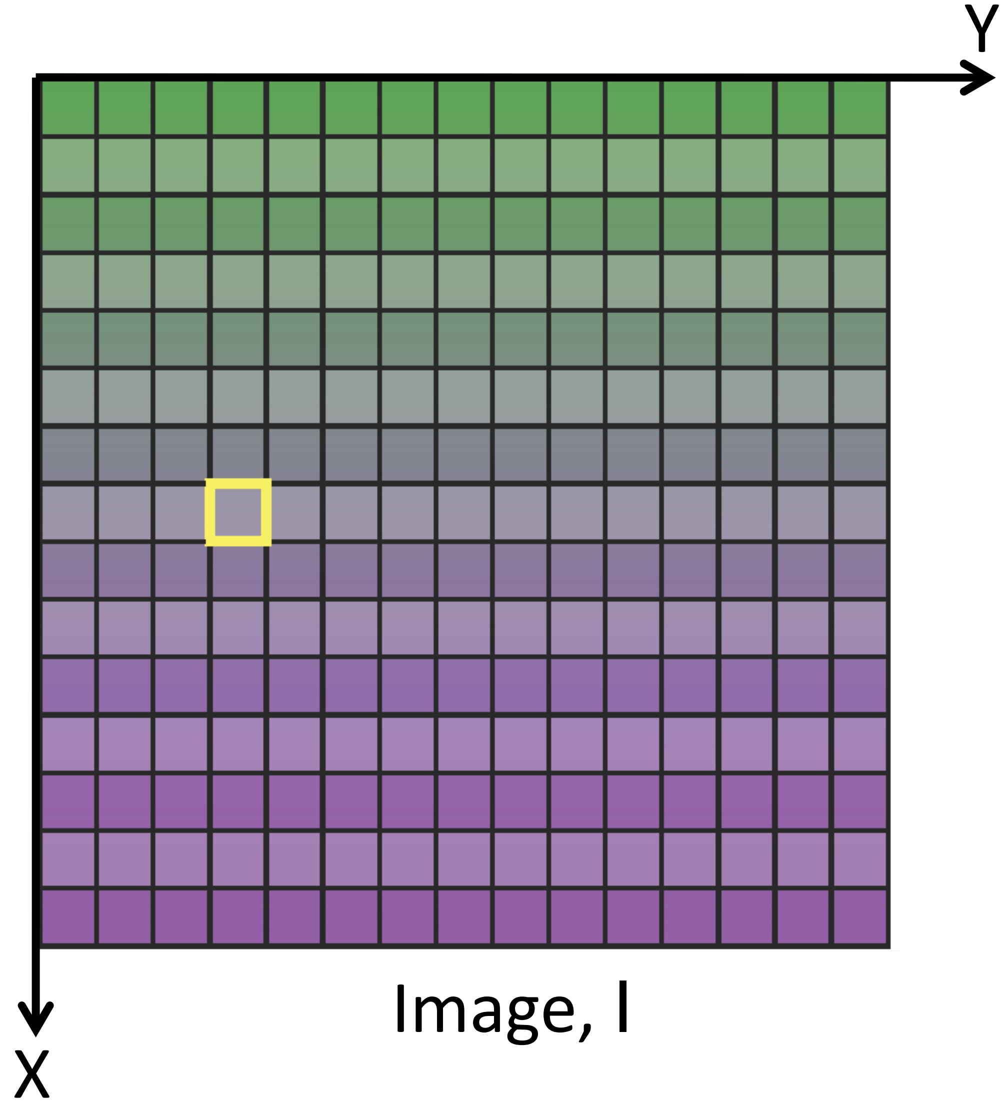
Continuous input on the sensor



Result of Image Sampling and Quantization

2-D Image Representation

- Array of 2D elements
  - Picture Elements – *pixel*
  - Randomly addressable locations
  - No adjacency information is stored explicitly

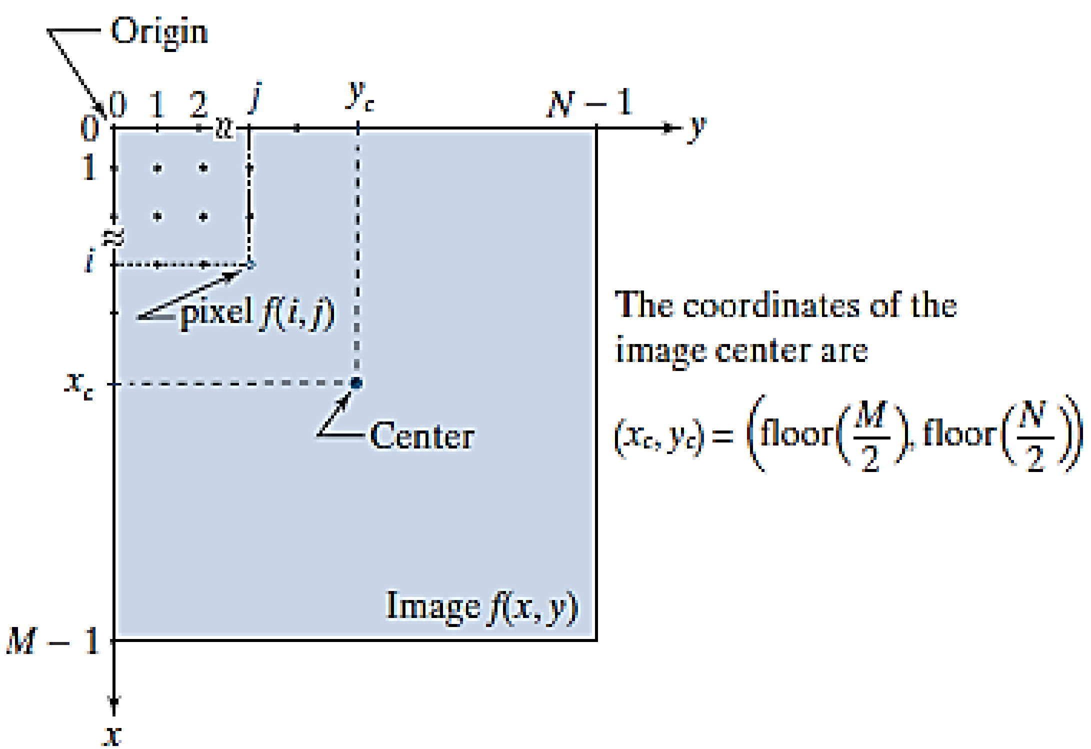


Image, I

2-D Image Representation

Spatial domain representation:

- We define the *origin* of an image at the ***top left corner***. This is a convention based on the fact that many image displays (e.g., TV monitors) sweep an image starting at the top left and moving to the right, one row at a time.

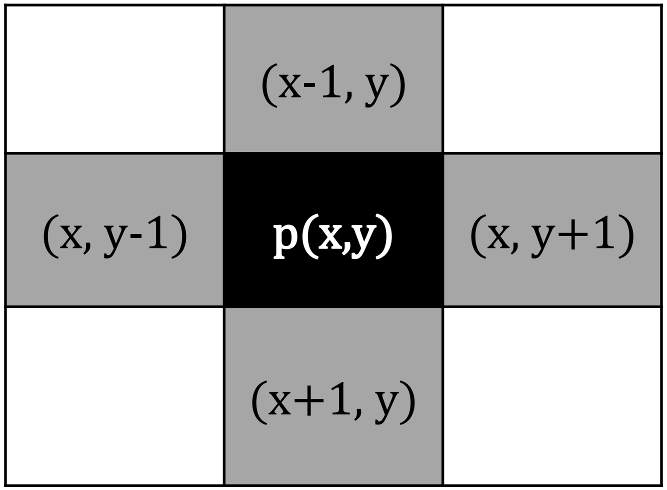


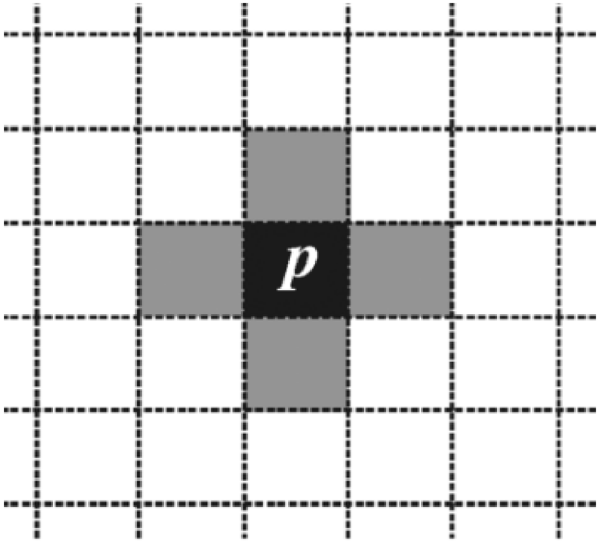
Neighbors of a Pixel

Neighbors of a Pixel

- The ***4-neighbors*** coordinates of a pixel are given by:

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





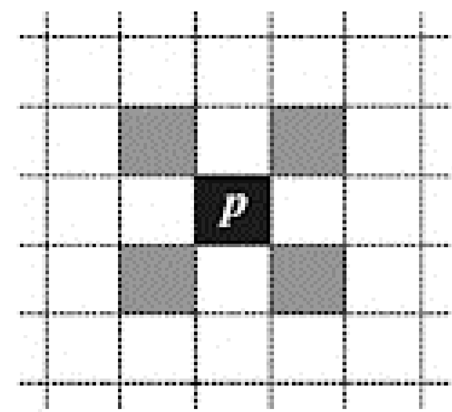
- This set of pixels is denoted by $N_4(p)$. Each pixel is <u>one unit distance</u> from (x, y).

Neighbors of a Pixel

- The ***four diagonal neighbors*** coordinates of a pixel are given by:

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

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

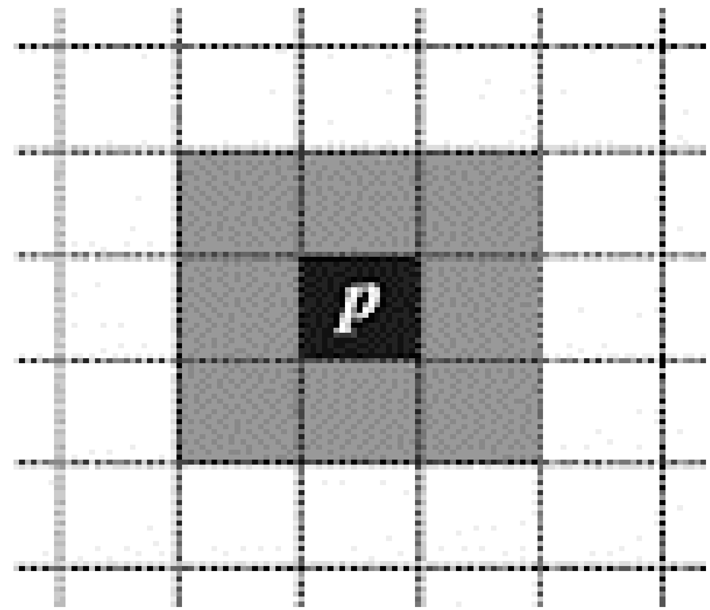


- This set of pixels is denoted by $N_D(p)$. Each pixel is <u>one unit distance</u> from (x, y).

Neighbors of a Pixel

- $N_D(p)$, together with the 4-neighbors:$N_4(p)$, are called the ***eight neighbors*** of p, denoted by $N_8\ (p)$.

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



Adjacency and Connectivity

Adjacency and Connectivity

- Let $\mathbf{V}$ be a set of intensity values used to define adjacency and connectivity.

- In a binary image, $\mathbf{V} = \{\mathbf{1}\}$, if we are referring to adjacency of pixels with value 1. Similarly $\mathbf{V} = \{\mathbf{0}\}$, if we are referring to adjacency of pixels with value **0**.

- In a gray-scale image, the idea is the same, but V typically contains more elements.
  - If the possible intensity values are $0 - 255$, V set can be any subset of these 256 values. For example, $\mathrm{V} = \{180, 181, 182, \ldots, 200\}$.

3 Types of Adjacency

- 4-adjacency: Two pixels p and q with values from V are 4-adjacent if $q \in N_4(p)$.

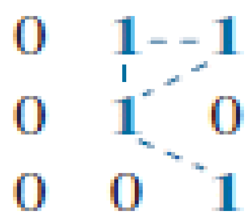
- 8-adjacency: Two pixels p and q with values from V are 8-adjacent if $q \in N_8(p)$.

- m-adjacency: Two pixels p and q with values from V are m-adjacent if:
  - $q \in N_4(p)$ or
  - $q \in N_D(p)$ and $N_4(p) \cap N_4(q)$ has no pixel whose values are from V (no intersection)
  - Use 4-adjacency where possible and 8-adjacency where not possible.

Types of Adjacency

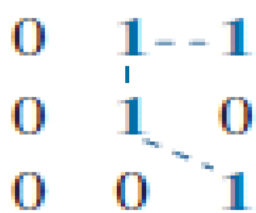
- **m-adjacency** (Mixed adjacency) is a modification of 8-adjacency, and is introduced to eliminate the ambiguities that may result from using 8-adjacency.

- For example, consider the pixel arrangement in following figure and let V = {1}. The three pixels at the top show multiple (ambiguous) 8-adjacency, as indicated by the dashed lines.



Path between top-right and bottom-right points

- This ambiguity is removed by using *m*-adjacency, as shown in the following figure.



Path between top-right and bottom-right points

Digital Path

A Digital Path

- A digital ***path*** (or curve) from pixel **p** with coordinate (x_0,y_0) to pixel **q** with coordinate (x_n,y_n) is a sequence of distinct pixels with coordinates (x_0,y_0), (x_1,y_1), …, (x_n, y_n) where pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are **adjacent** for $1 \le i \le n$.

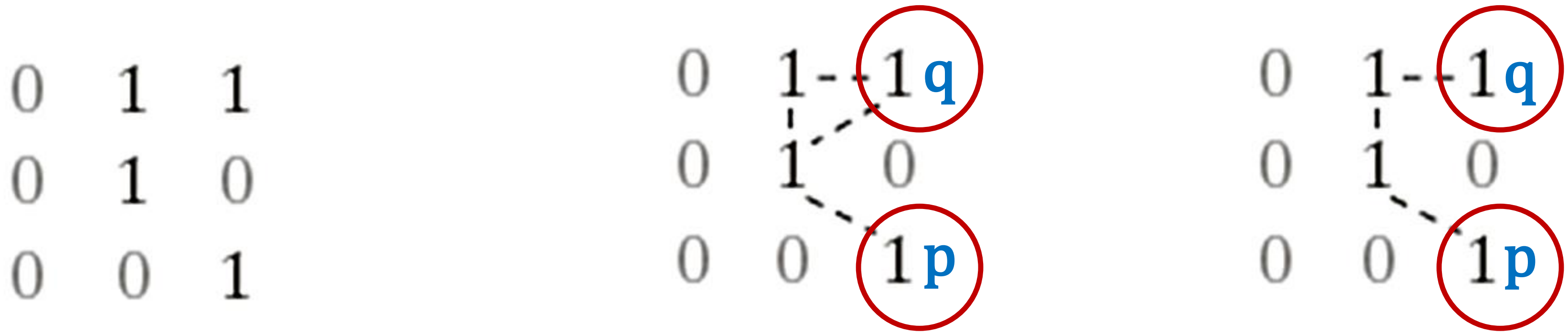
- In this case, ***n*** is the ***length*** of the path.

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

- We can specify **4-**, **8-** or **m-paths** depending on the type of adjacency specified.

A Digital Path

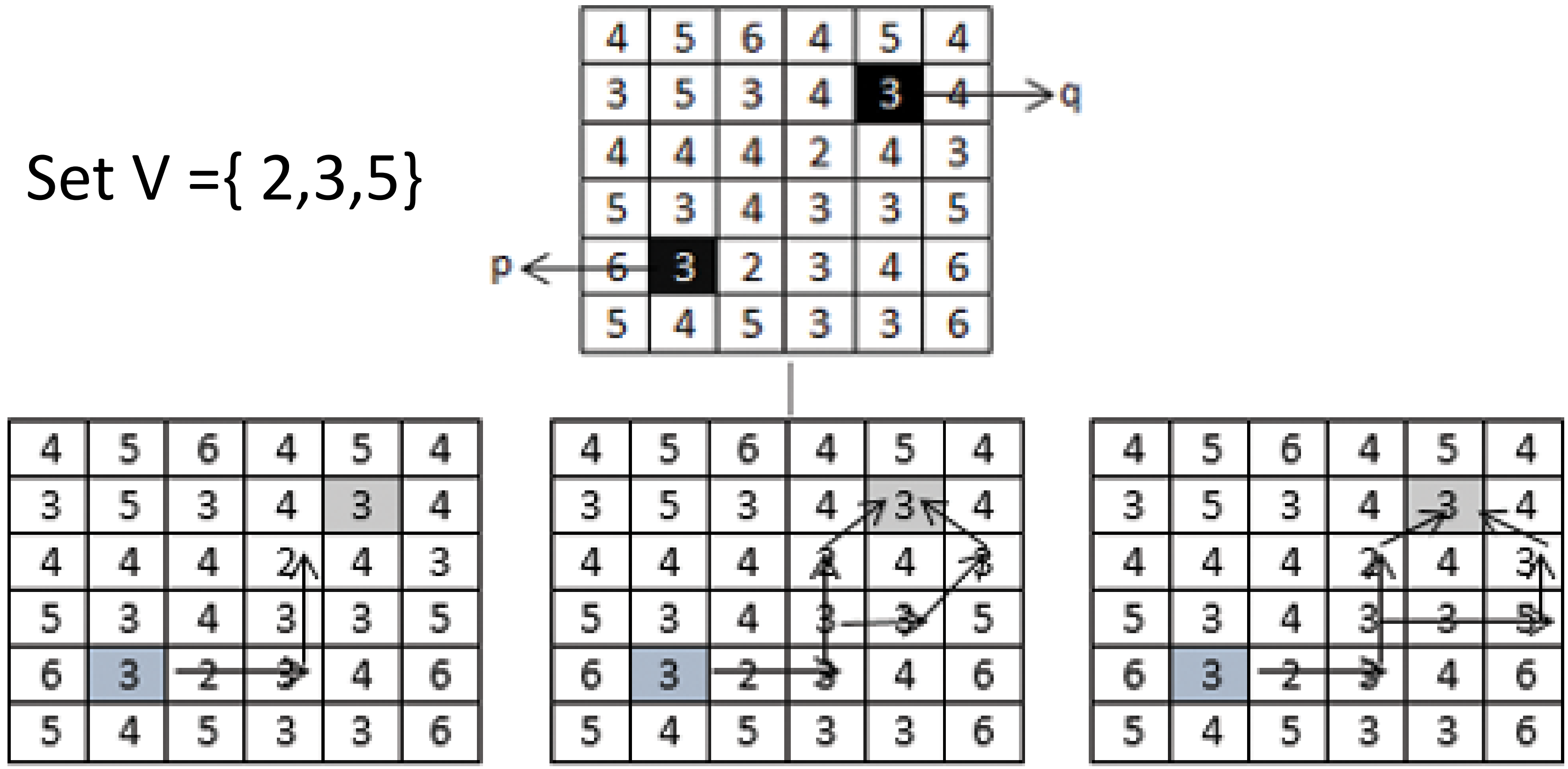
- For example:



- In the middle figure, the paths between the pixel **p** and **q** are **8-paths**. And the path between the same 2 pixels in the right figure is **m-path**.

A Digital Path

Set V ={ 2,3,5}



4-path | 8-path | m-path

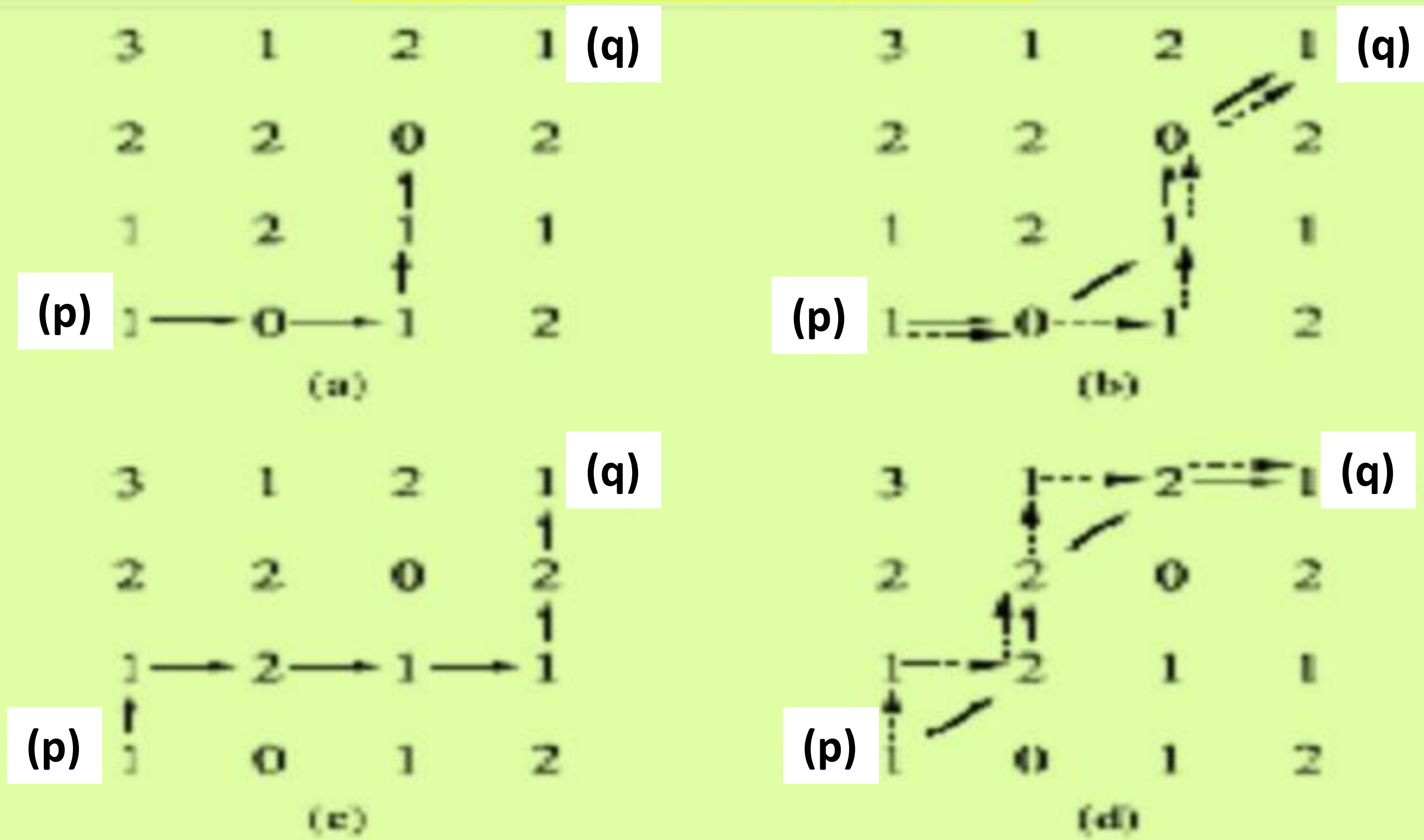
No path

Minimum path length = 5

Minimum path length = 5

A Digital Path

Find the minimum path distance



(a)4path for V = {0,1} (b)8 path(Solid Lines), m path(Dashed Lines)

(c) 4path for V = {1,2} (d) 8 path(Solid Lines), m path(Dashed Lines)

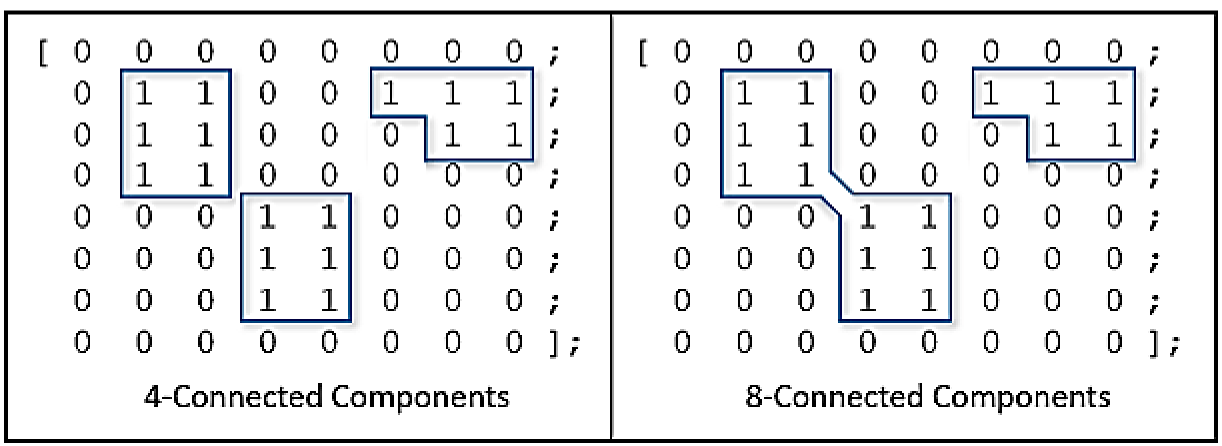
Connectivity

Connectivity

- Let **S** represent a subset of pixels in an image. Two pixels **p** and **q** are said to be <u>connected in S</u> if there exists a path between them consisting entirety of pixels in S.

- For any pixel **p** in **S**, the set of pixels that are connected to it in S is called a <u>connected component</u> of S.

- If the set **S** only has <u>one connected component</u>, then it is called a <u>connected set</u>.



V={1} V={1}

Region and Boundary

Region

- Let **R** be a subset of pixels in an image, we call R a **region** of the image if R is a connected set.

OR

- A region in an image is a group of connected pixels with similar properties.

- Two regions are adjacent if their union forms a connected set, otherwise disjoint.

- We consider 4 and 8 adjacency when referring to regions.

$$\begin{array}{l} \left.\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{array}\right\} R_i \\ \left.\begin{array}{ccc} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{array}\right\} R_j \end{array}$$

- In this example, the two regions of 1's are adjacent only if 8-adjacency is used.

- A 4-path between the two regions does not exist, so their union is not a connected set.

Region

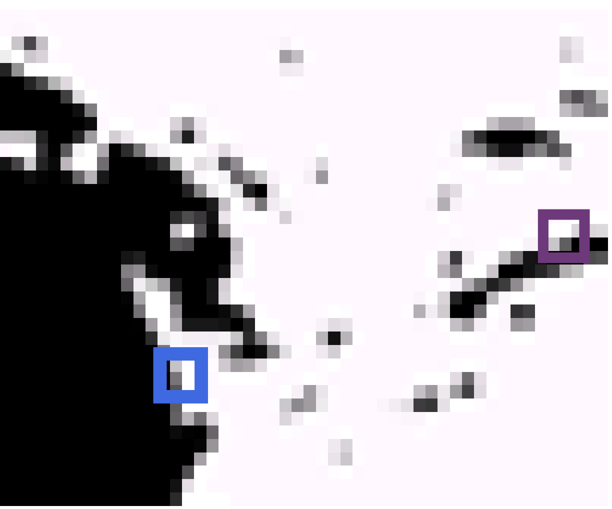
- Suppose an image contains K disjoint regions, R_k , k =1, 2,…, K, none of which touches the image border. Let $\mathbf{R_u}$ denote the union of all the K regions, and let $(\mathbf{R_u})^{\mathbf{c}}$ denote its complement:
  - ❖ We call all the points in R_u the ***foreground***, and
  - ❖ All the points in $(R_u)^c$ the ***background*** of the image.

Boundary

- The **boundary** (also called border or contour) of a region **R** is the set of pixels in the region that have one or more neighbors that are not in R.

OR

- The **boundary** of a region is the set of pixels in the region that have at least one background neighbor.



0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

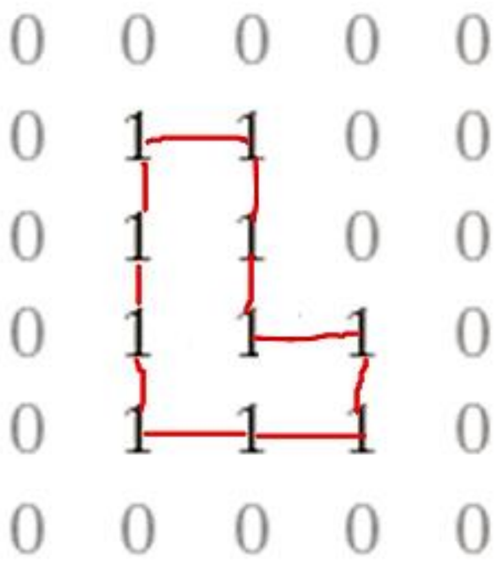
- In this example, the circled 1 is **not a member** of the border of the 1-valued region if 4-connectivity is used.
- As a rule, adjacency between points in a region and its background is defined using 8-connectivity to handle situations such as this.

Boundary

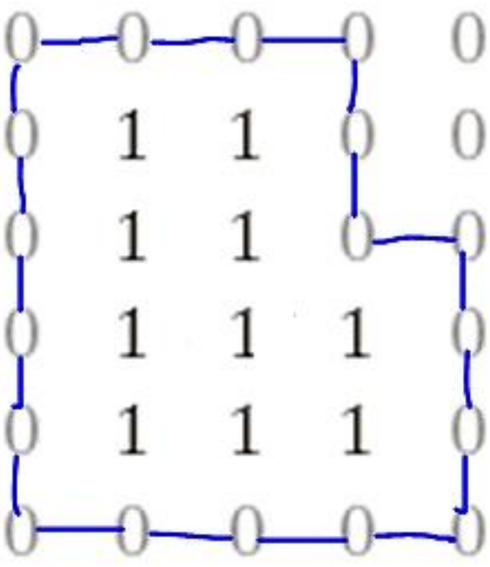
- Inner boundary: of the region
- Outer boundary: the border in the background

0	0	0	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

A region of V={1}, 8-connectivity and its background



Inner boundary

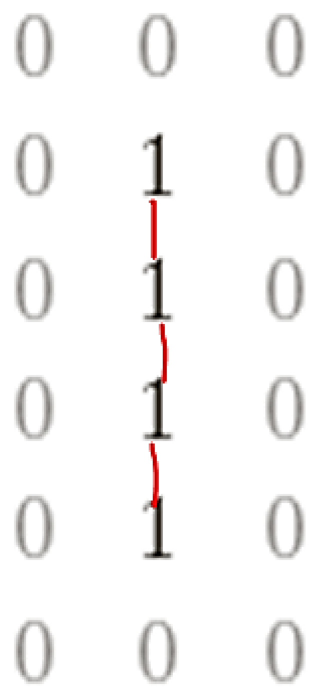


Outer boundary

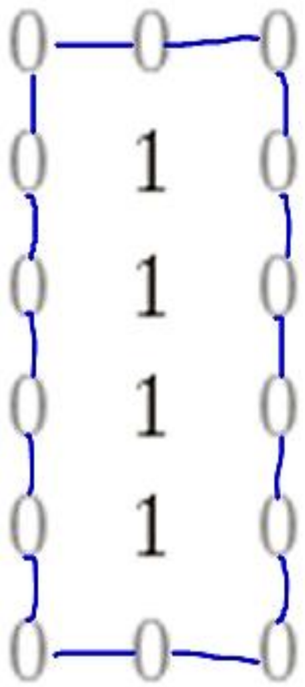
Boundary

- Inner boundary: of the region
- Outer boundary: the border in the background

A region of V={1}, 8-connectivity and its background



Inner boundary



Outer boundary

Boundary vs. Edge

- Boundary: global concept: the boundary of a finite region forms a closed path.

- Edge: local concept, based on a measure of intensity-level discontinuity at a point.
  - Edge point
  - Edge segment

Proximity Relationship

Proximity Relationship

- What is the distance between any two pixels?
  - Distance in domain (2D plane)
  - Distance in range (Gray scale values)

- Meaning of distance is context dependent.

Distance Measures

- For pixels p, q and z, $\mathbf{D}$ is a **distance function** or **metric** if:

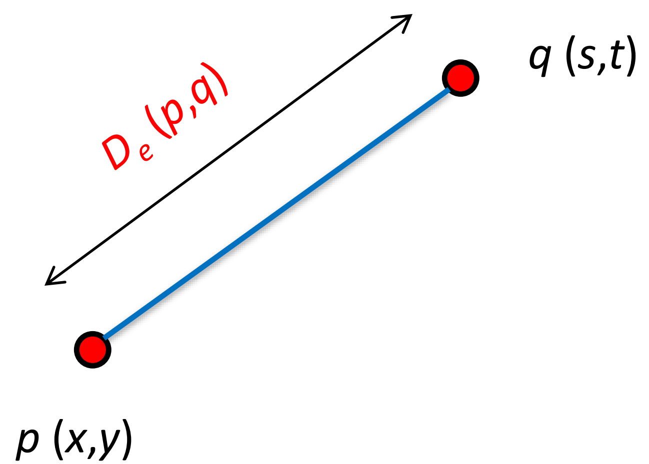
1. $D(p, q) \geq 0$ (non-negativity)
2. $D(p, q) = 0$ if and only if $p = q$ (Identity of indiscernible)
3. $D(p, q) = D(q, p)$ (Symmetry)
4. $D(p, z) \leq D(p, q) + D(q, z)$ (triangle inequality)

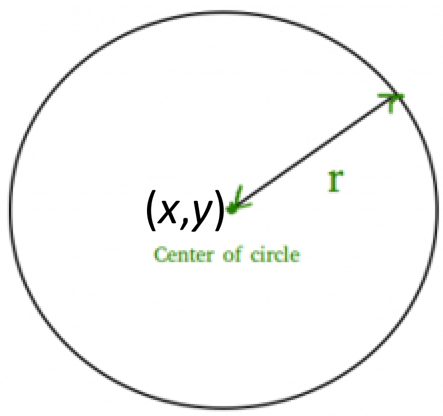
Distance Measures

- The ***Euclidean Distance*** between **p** with coordinates **(x, y)** and **q** with coordinates **(s, t)** is defined as:

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

- Pixels having a distance less than or equal to some value **r** from (x, y) are the points contained in a **disk** of radius **r** centered at (x, y).



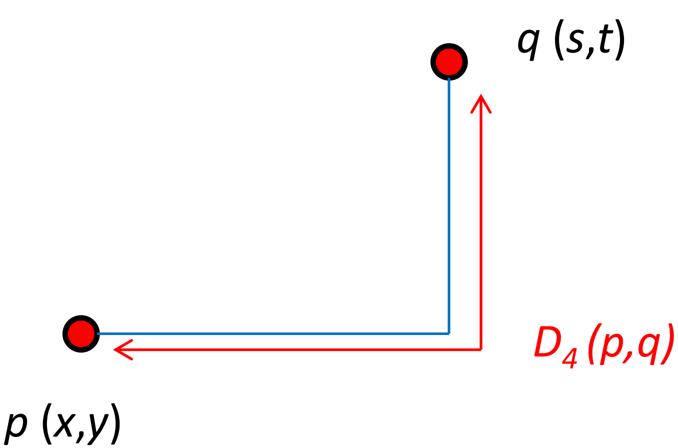


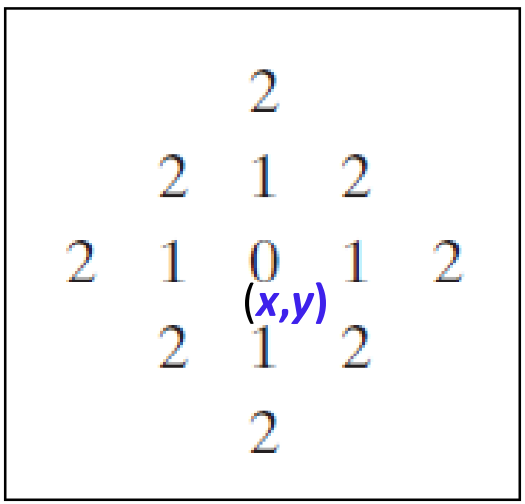
Distance Measures

- The ***D_4 distance*** (also called city-block distance) between **p** and **q** is defined as:

$$D_4(p, q) = |x - s| + |y - t|$$

- Pixels having a D_4 distance from (x,y), less than or equal to some value **r** form a **diamond** centered at (x,y).





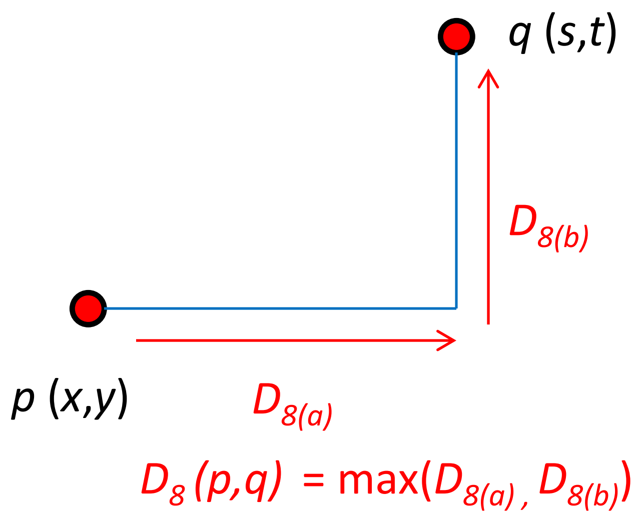
$D_4 \leq 2$

Distance Measures

- The ***D_8 distance*** (also called chessboard distance) between **p** and **q** is defined as:

$$D_8(p,q) = \max(|x - s|, |y - t|)$$

- Pixels having a $\mathbf{D_8}$ distance from (x,y), less than or equal to some value **r** form a **square** Centered at (x,y).



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

(x,y)

$\mathbf{D_8 \le 2}$

Distance Measures

- The ***D_m distance*** is defined as the shortest m-path between the points.

- Consider the following arrangement of pixels and assume that $\mathbf{p}$, $\mathbf{p_2}$, and $\mathbf{p_4}$ have value $\mathbf{1}$ and that $\mathbf{p_1}$ and $\mathbf{p_3}$ can have a value of $\mathbf{0}$ or $\mathbf{1}$.

- Suppose that we consider the adjacency of pixels values $\mathbf{1}$ (i.e. $\mathrm{V} = \{1\}$), what is the distance between $\mathbf{p}$ and $\mathbf{p_4}$ i.e. $\mathbf{D_m(p,p_4)}$?

	p_3	p_4
p_1	p_2	
p		

Distance Measures

- **Case1**: If p_1=0 and p_3=0, $D_m(p,p_4)$=**2** *($p \to p_2 \to p_4$)*

	0	1
0	1	
1		

- **Case2**: If p_1=1 and p_3=0, $D_m(p,p_4)$=**3** ($p \to p_1 \to p_2 \to p_4$)

	0	1
1	1	
1		

	p_3	p_4
p_1	p_2	
p		

- **Case3**: If p_1=0 and p_3=1, $D_m(p,p_4)$=**3** ($p \to p_2 \to p_3 \to p_4$)

	1	1
0	1	
1		

- **Case4**: If p_1=1 and p_3=1, $D_m(p,p_4)$=**4** ($p \to p_1 \to p_2 \to p_3 \to p_4$)

	1	1
1	1	
1		

Linear Operations

Linear Vs. Nonlinear Operation

- Let $\mathbf{H}$ be a general operator that produces $g(x,y)$ for a given image $f(x,y)$:

$$\mathbf{H[f(x,y)]=g(x,y)}$$

- Given two arbitrary constants, *a* and *b*, and two arbitrary images $f_1(x, y)$ and $f_2(x, y)$, $\mathbf{H}$ is said to be a ***Linear operator*** if:

$$H[af_1(x,y) + bf_2(x,y)] = aH[f_1(x,y)] + bH[f_2(x,y)]$$
$$= ag_1(x,y) + bg_2(x,y)$$

  - Additivity: $H[f_1(x,y)+f_2(x,y)]=H[f_1(x,y)]+H[f_2(x,y)$
  - Homogeneity: $H[c\times f(x,y)]=c\times H[f(x,y)]$

An operator that fails to satisfy above conditions is said to be ***nonlinear***.

Linear Operation : example

- Suppose that $\mathbf{H}$ is the summation operator $\boldsymbol{\Sigma}$ and function performed by this operator is simply to ***sum its inputs***:

$$\begin{aligned}\sum\left[af_1(x,y)+bf_2(x,y)\right] &= \sum af_1(x,y)+\sum bf_2(x,y)\\ &= a\sum f_1(x,y)+b\sum f_2(x,y)\\ &= ag_1(x,y)+bg_2(x,y)\end{aligned}$$

Where the first step follows from the fact that **summation is distributive**. So, an expansion of the left side is equal to the right side, and we conclude that the **sum operator is linear**.

Linear Operation

- How to show an operator H is linear?
  - We need to show that for arbitrary images $\mathrm{f}_1(\mathrm{x,y})$ and $\mathrm{f}_2(\mathrm{x,y})$, H satisfies the following equation:

$$\mathrm{H}[\alpha \mathrm{f}_1(\mathrm{x,y})+\beta \mathrm{f}_2(\mathrm{x,y})] = \alpha \mathrm{H}[\mathrm{f}_1(\mathrm{x,y})]+ \beta \mathrm{H}[\mathrm{f}_2(\mathrm{x,y})]$$

Examples

- $H[f(x,y)]=f(x-x_0,y-y_0)$
- $H[f(x,y)]=[f(x,y)]^2$
- $H[f(x,y)]=\max[f(x,y)]$
- $H[f(x,y)]=f(M\times x,N\times y)$, where $M,N \in Z^+$
- $H[f(x,y)]=af(x,y)+b$, where a,b are arbitrary scalars

Non-linear Operation

$$\mathcal{H}\left[a f_1(x,y) + b f_2(x,y)\right] = a\mathcal{H}\left[f_1(x,y)\right] + b\mathcal{H}\left[f_2(x,y)\right]$$

- If **H** ***does not*** satisfy additive and homogeneity property, then it is a non-linear operator.

Non-Linear Operation : example

- Suppose that we are working with the **max operation**, whose function is to find the *maximum value of the pixels* in an image.

- Consider the following two images and suppose that we let ***a* = 1** and ***b* = −1**:

$$\mathcal{H}\left[a f_1(x,y) + b f_2(x,y)\right] = a\mathcal{H}\left[f_1(x,y)\right] + b\mathcal{H}\left[f_2(x,y)\right]$$

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

LHS

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\}$$
$$= -2$$

RHS

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7 = -4$$

The LHS and RHS of the linear equation are <u>not equal</u> in this case, so we have proved that the **max operator is nonlinear**.

Next Lecture

- Elementwise Versus Matrix Operations

- Operations on Images
  - Arithmetic Operations
  - Set and Logical Operations

- Spatial Operations
  - Single-pixel Operations
  - Neighborhood Operations