Introduction to Digital Image Processing

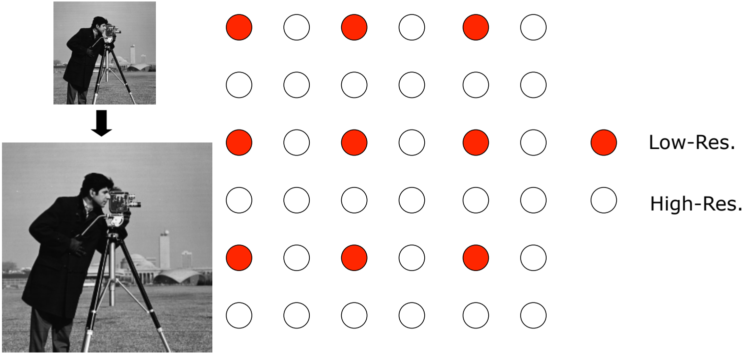
Dr. Kundan Kumar
Ph.D. (IIT Kharagpur)
Assistant Professor
ECE Department (Cabin - E139)

Institute of Technical Education & Research (ITER)
Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India-751030

Image Interpolation

- Interpolation is a basic tool which helps to perform specific tasks in digital image processing like zooming, shrinking, rotating, and geometric corrections.
- Fundamentally, interpolation is the process of using known data to estimate values at unknown locations.
- Type of interpolations:
  - Nearest neighbor interpolation
  - Bilinear interpolation
  - Bicubic interpolation
  - Higher order interpolation: SPLINE & SINC

Nearest Neighbor Interpolation



- This approach is simple but, it has the tendency to produce undesirable *artifacts*.

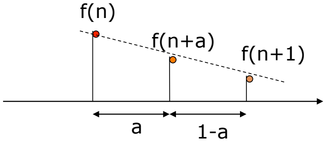
Nearest Neighbor Interpolation: Example

- Upscale 3×3 image by factor two using Nearest Neighbor Interpolation.

3	4	5
6	2	4
2	4	7

3	3	4	4	5	5
3	3	4	4	5	5
6	6	2	2	4	4
6	6	2	2	4	4
2	2	4	4	7	7
2	2	4	4	7	7

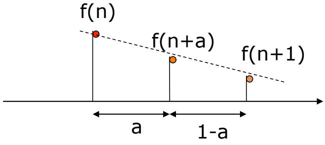
Bilinear Interpolation



$$\boxed{f(n+a) = (1-a) \times f(n) + a \times f(n+1),\ 0<a<1}$$

Note: when a=0.5, we simply have the average of two

Bilinear Interpolation



$$\boxed{f(n+a)=(1-a)\times f(n)+a\times f(n+1),\ 0<a<1}$$

Note: when a=0.5, we simply have the average of two

$f(n)=[0,120,180,120,0]$

↓ Interpolate at 1/2-pixel

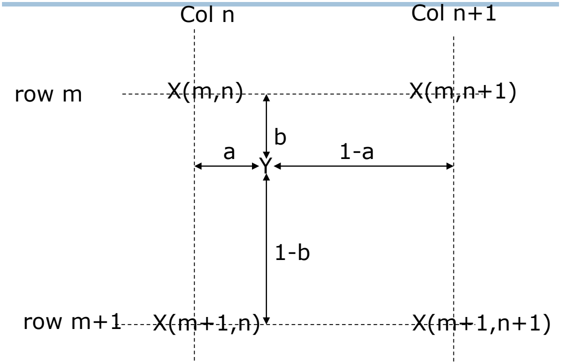
$f(x)=[0,60,120,150,180,150,120,60,0],\ x=n/2$

↓ Interpolate at 1/3-pixel

$f(x)=[0,20,40,60,80,100,120,130,140,150,160,170,180,\ldots],\ x=n/6$

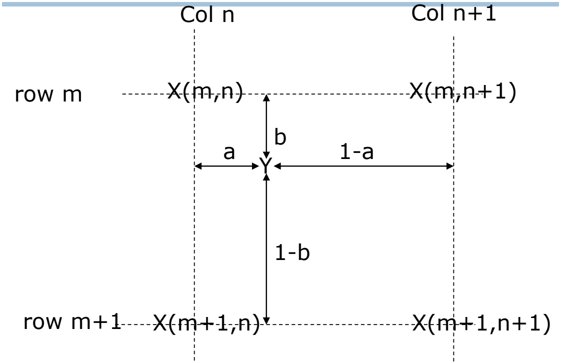
Bilinear Interpolation

- What is the interpolated value at Y?



Bilinear Interpolation

- What is the interpolated value at Y?



Ans.: (1-a)(1-b)X(m,n)+(1-a)bX(m+1,n)
+a(1-b)X(m,n+1)+abX(m+1,n+1)

Image Interpolation

- For bilinear interpolation, the assign value is obtained using the equation

$$v(x,y) = ax + by + cxy + d$$

where the four coefficients are determined from the four equations in four unknowns.

- For bicubic interpolation, the assign value is obtained using the equation

$$v(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j$$

where the sixteen coefficients are determined from the sixteen equations in sixteen unknown.

Bilinear Interpolation: Example

- Upscale 3×3 image by factor two using Bilinear Interpolation.

3	4	5
6	2	4
2	4	7

3	3.5	4	4.5	5
4.5	3.75	3	3.75	4.5
6	4	2	3	4
4	3.5	3	4.25	5.5
2	3	4	5.5	7

Image Interpolation: Comparision

Original image:  x 10

Nearest-neighbor interpolation

Bilinear interpolation

Bicubic interpolation

Relationships between pixels: Neighbors of a pixel

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

- A pixel p at location (x, y) has two horizontal and two vertical neighbors. whose coordinates are

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

- This set of four pixels is called the 4-neighbors of p, denoted as $N_4(p)$.
- Each of these neighbors is at a unit distance from p.
- If p is a boundary pixel then it will have less number of neighbors.

Relationships between pixels: Neighbors of a pixel

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

- A pixel p has four diagonal neighbors

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

  denoted by $N_D(p)$.
- Combining 4-neighbors and diagonal-neighbors gives 8-neighbors of p, denoted as $N_8(p)$.
- If p is a boundary pixel then both $N_D(p)$ and $N_8(p)$ will have less number of pixels.

Adjacency

- Let V be the set of intensity values used to define adjacency.
- For a binary image $V = \{0\}$ or $\{1\}$
- For a grayscale image $V \subset \{0, 1, \ldots, 255\}$
- There are three types of adjacency:
  - *4-adjacency:* Two pixel p and q with values from V are 4-adjacent if $q \in N_4(p)$
  - *8-adjacency:* Two pixel p and q with values from V are 8-adjacent if $q \in N_8(p)$
  - *m-adjacency* (mixed adjacency): Two pixel p and q with values from V are m-adjacent if
    - $q \in N_4(p)$, or
    - $q \in N_D(p)$ and the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

Path

- A path from pixel p with coordinate (x, y) to pixel q with coordinates (s, t) is a sequence of distinct pixels with coordinates
$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$
where $(x_0, y_0) = (x, y)$, $(x_n, y_n) = (s, t)$, and pixels (x_i, y_i) and (x_{i-1}, y_{i-1}) are adjacent for $1 \leq i \leq n$.
- Here, n is the length of the path.
- If $(x_0, y_0) = (x_n, y_n)$, the path is closed path.
- We can define 4-, 8-, or m-paths depending on the type of adjacency specified.

Why m-adjacency?

- Mixed adjacency is a modification of 8-adjacency.
- It is introduced to eliminate the ambiguities that often arise when 8-adjacency is used.
- Example: $V = \{1\}$

$$\begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

4-connected 8-connected m-connected

Connectivity

- Connectivity between pixels is a very important concept.
- Let S represent a subset of pixels in an image.
- Two pixel p and q are said to be *connected* in S if there exits a path between them consisting entirely of pixels in S.
- For any pixel p in S, the set of pixels that are connected to it in S is called a *connected component* of S
- If S only has connected component, then set S is called a *connected set*.
- Let R be a subset of pixels in an image. Then R is a *region* of the image if R is a connected set.

Connected Components

- Two regions, R_i and R_j are said to be adjacent if their union forms a connected set.
- Regions that are not adjacent are said to be *disjoint*.
- We consider 4- and 8-adjacency when referring to regions.

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

- Suppose that an image contains K disjoint regions, R_k, $k = 1, 2, \ldots, K$, none of which touches the image border.
- All the points in R_u the *foreground*, and all the points in $(R_u)^c$ the *background*.
- R_u denote the union of all the K regions.

Boundary

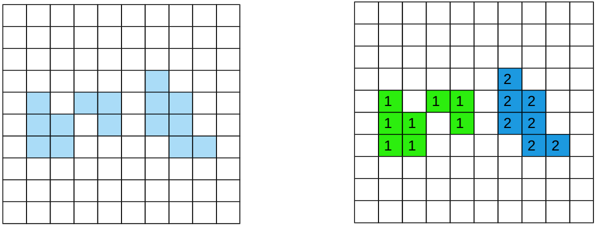
- The *boundary* (also called the *border* or *contour*) of a region R is the set of points that are adjacent to points in the R^c.

$$
\begin{matrix}
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0
\end{matrix}
$$

- Depending on the type of connectivity and edge operators used, the edge extracted from a binary region will be the same as the region boundary.
- Edges as intensity discontinuities and boundaries as closed paths.

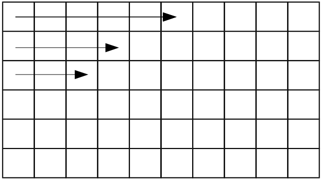
Connected component labeling

- Ability to assign different labels to various disjoint connected set of an image.



- Connected component labeling is a fundamental step in automated image analysis
  - Shape
  - Area
  - Boundary
  - Shape/Area/Boundary based features

Connected component labeling: Algorithm



- Scan an image from left to right and from top to bottom.
- Assume 4-connectivity
- Suppose p be a pixel at any step in the scanning process

	r
t	p

- Before p, point r and t are scanned

Connected component labeling: Algorithm

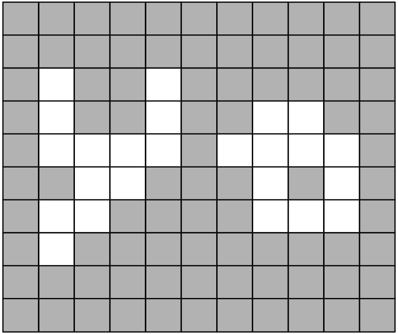
	r
t	p

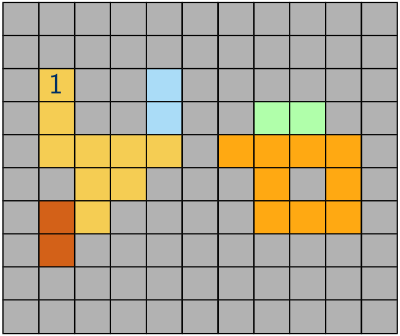
1. $I(p) \Rightarrow$ Pixel value at position p.
2. $L(p) \Rightarrow$ Label assigned to pixel location p.
3. If $I(p) = 0$, move to next scanning position.
   if $I(p) = 1$ and $I(r) = I(t) = 0$ then assign a new label to position p
4. If $I(p) = 1$ and only one of the two neighbor is 1. Then assign its label to p.
5. If $I(p) = 1$ and both r and t are 1's, then
   If $L(r) = L(t)$ then $L(p) = L(r)$
   If $L(r) \neq L(t)$ then assign one of the labels to p and make a note that the two labels are equivalent.

Connected component labeling: Algorithm

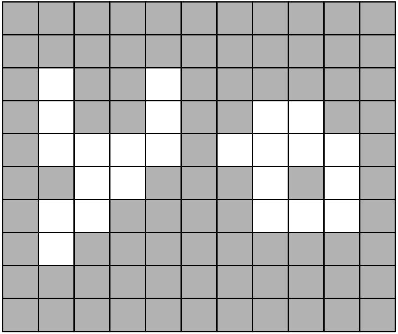
- At end of the scan all pixels with value 1 are labeled.
- Some labels are equivalent.
- During second pass process equivalent pairs to from equivalence classes.
- Assign a different label to each class. In the second pass through the image replace each label by the label assigned to its equivalence class.

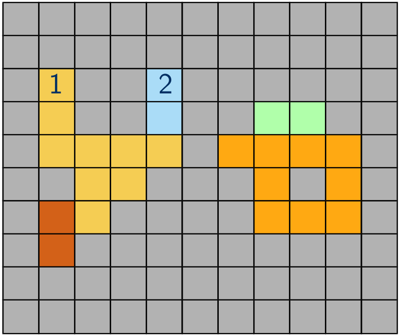
Connected component labeling: Example



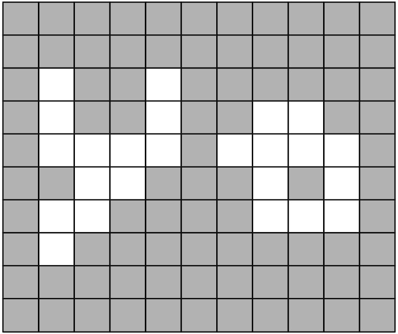


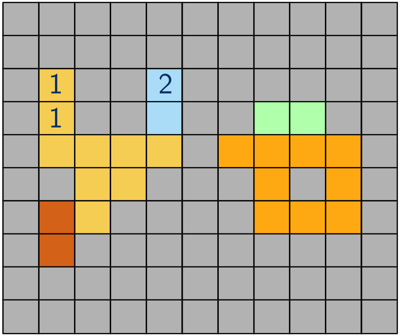
Connected component labeling: Example



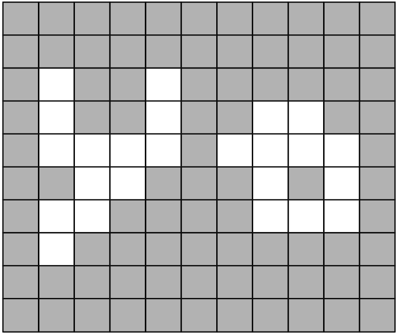


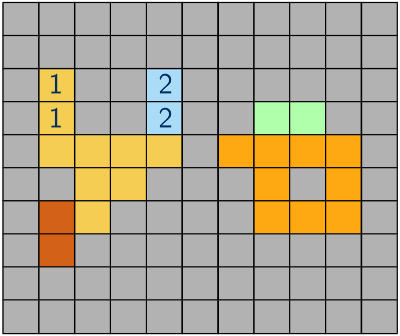
Connected component labeling: Example



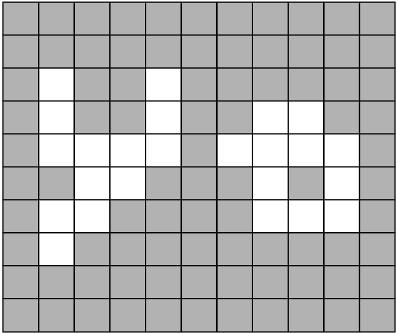


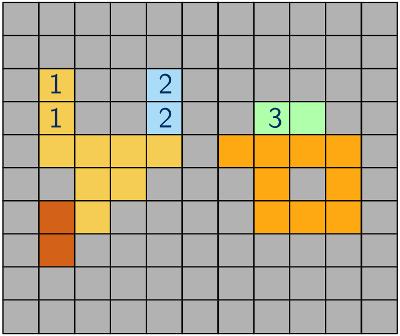
Connected component labeling: Example



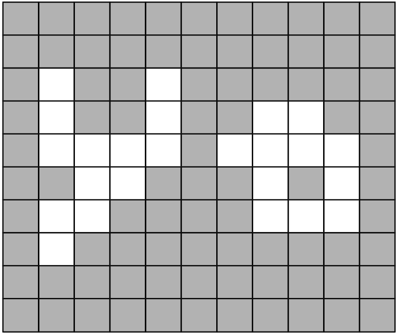


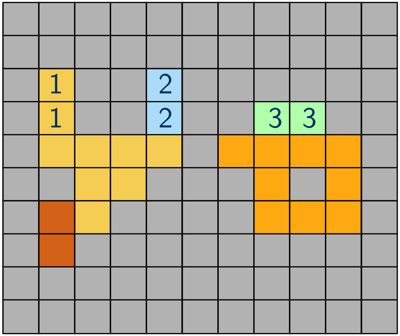
Connected component labeling: Example



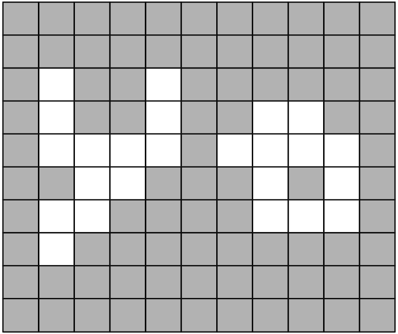


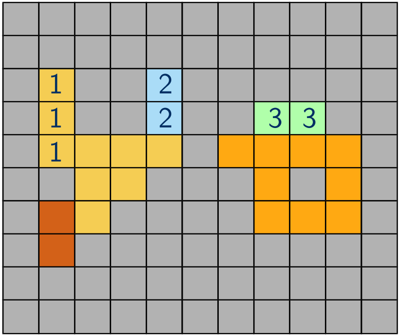
Connected component labeling: Example



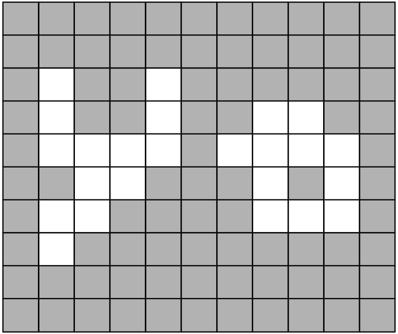


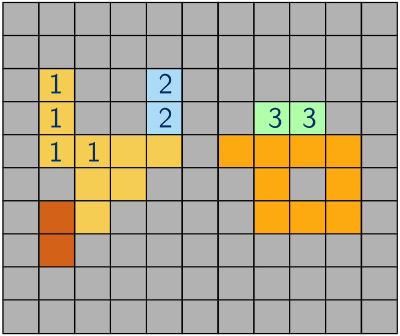
Connected component labeling: Example



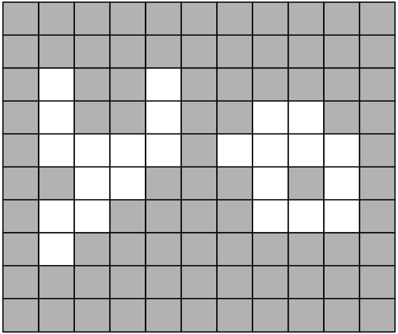


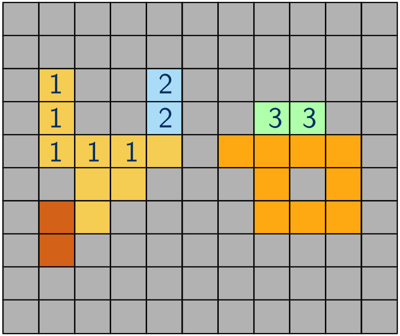
Connected component labeling: Example



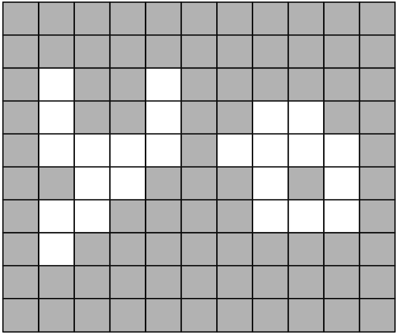


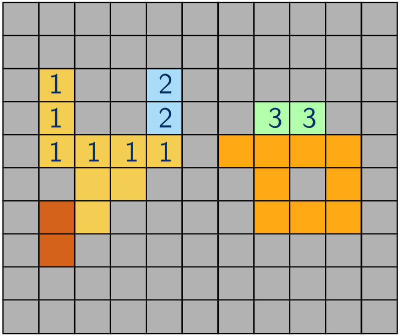
Connected component labeling: Example





Connected component labeling: Example

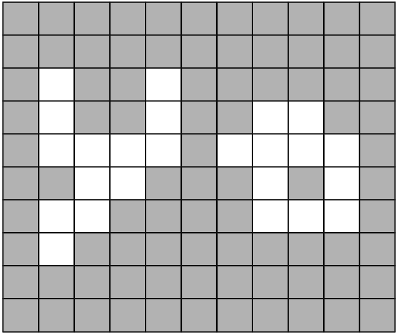


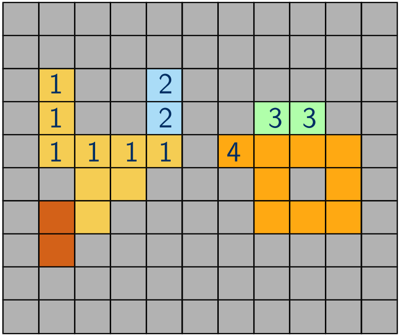


Equivalent pairs:

(1,2)

Connected component labeling: Example

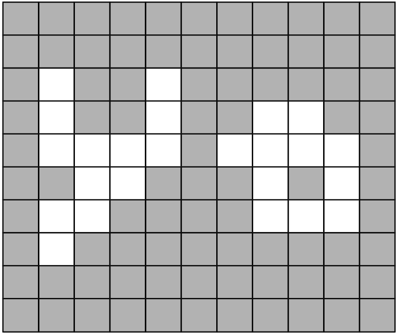




Equivalent pairs:

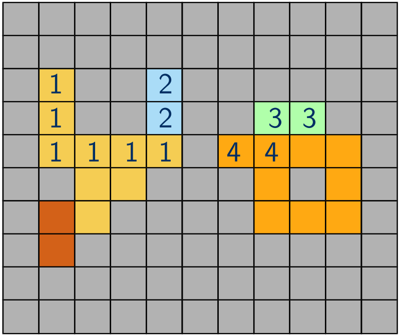
(1,2)

Connected component labeling: Example

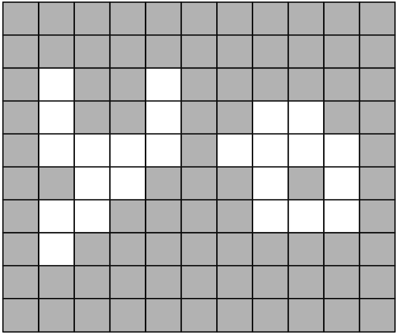


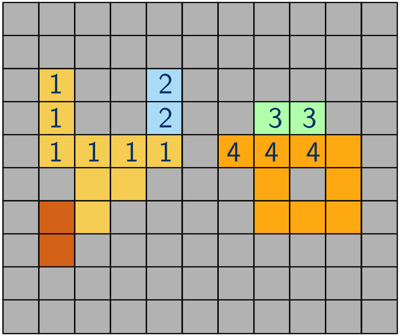
Equivalent pairs:

(1,2) (3,4)



Connected component labeling: Example

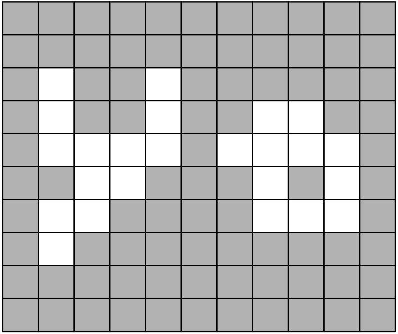


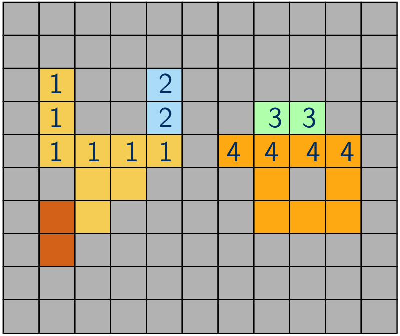


Equivalent pairs:

(1,2) (3,4)

Connected component labeling: Example

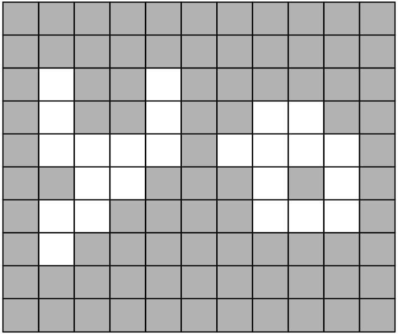


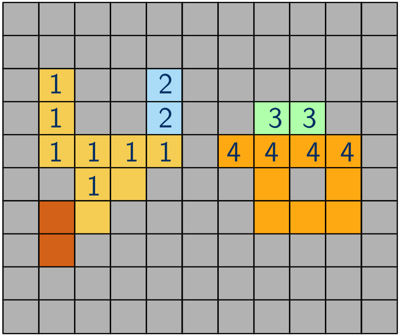


Equivalent pairs:

(1,2) (3,4)

Connected component labeling: Example

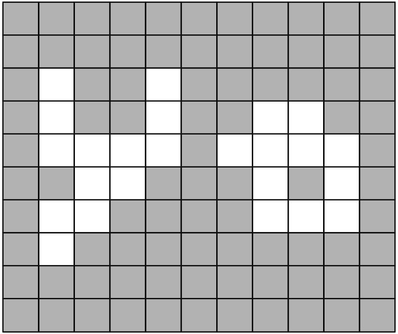




Equivalent pairs:

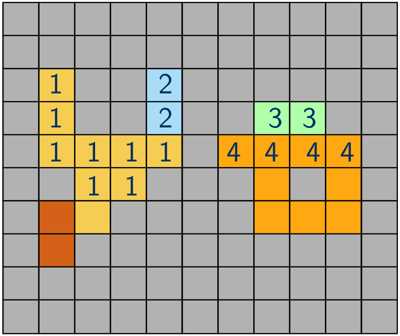
(1,2) (3,4)

Connected component labeling: Example

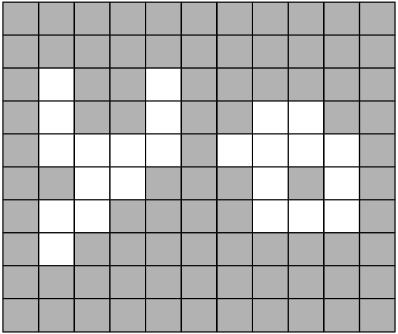


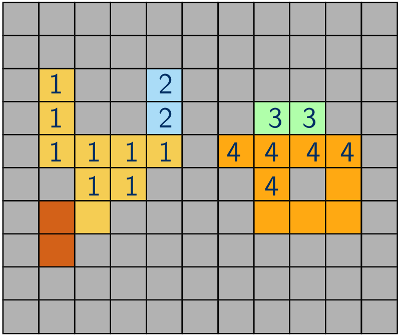
Equivalent pairs:

(1,2) (3,4)



Connected component labeling: Example

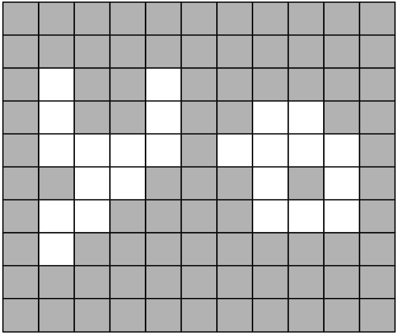


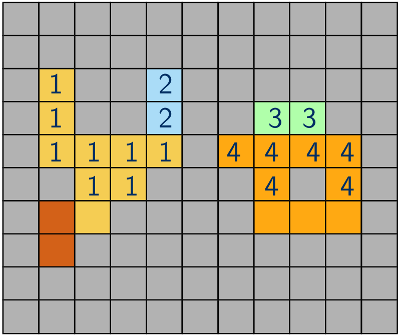


Equivalent pairs:

(1,2) (3,4)

Connected component labeling: Example

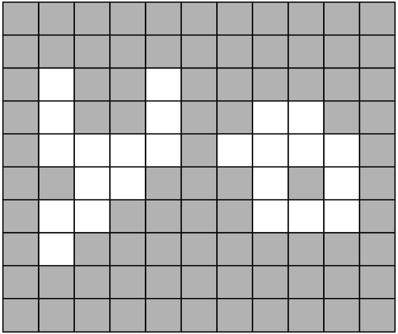


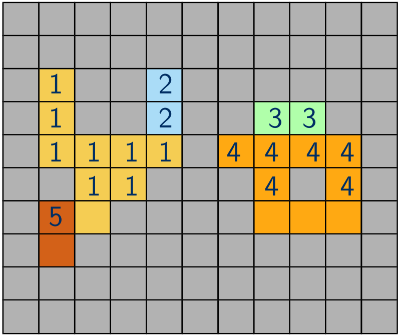


Equivalent pairs:

(1,2) (3,4)

Connected component labeling: Example

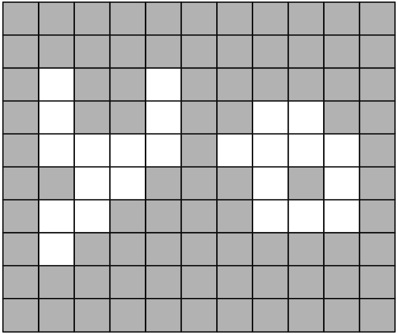




Equivalent pairs:

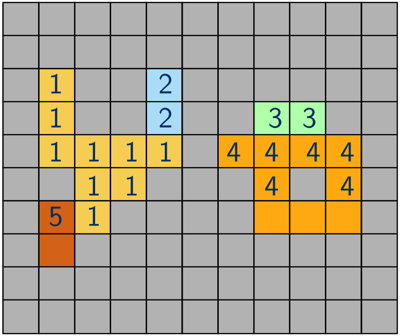
(1,2) (3,4)

Connected component labeling: Example

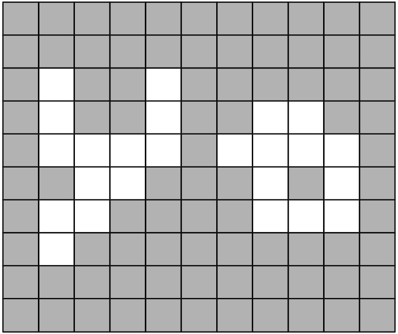


Equivalent pairs:

(1,2) (3,4) (1,5)

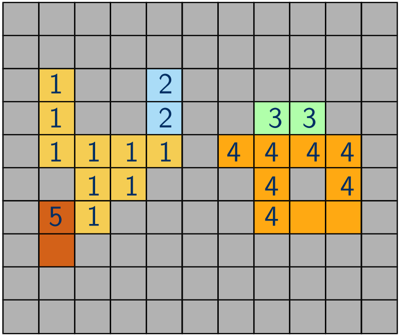


Connected component labeling: Example

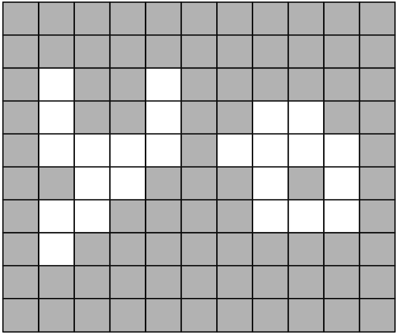


Equivalent pairs:

(1,2) (3,4) (1,5)

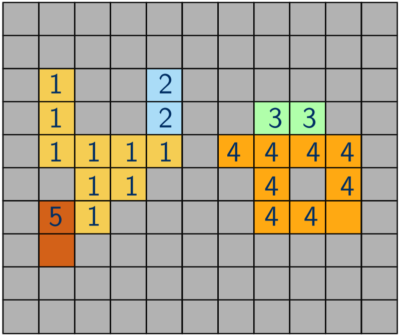


Connected component labeling: Example

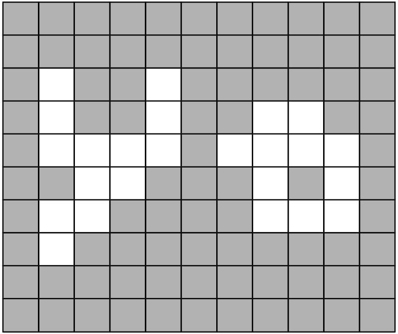


Equivalent pairs:

(1,2) (3,4) (1,5)

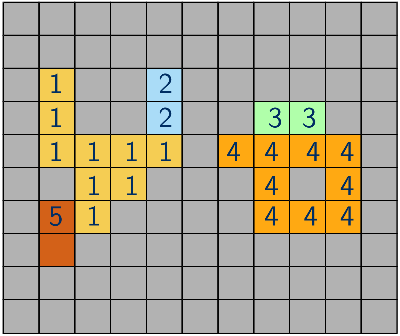


Connected component labeling: Example

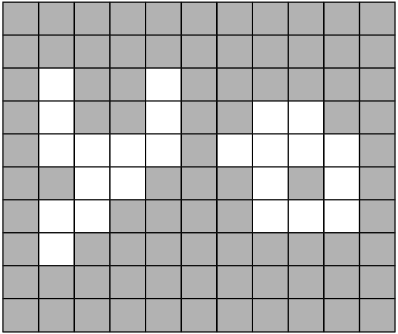


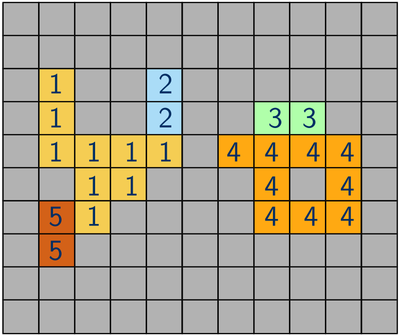
Equivalent pairs:

(1,2) (3,4) (1,5)



Connected component labeling: Example





Equivalent pairs:

(1,2) (3,4) (1,5)

Connected component labeling: Example

- Final result obtained by updating labels as

$$(1,2),(1,5) = 1$$

$$(3,4) = 3$$

	1			1						
	1			1			3	3		
	1	1	1	1		3	3	3	3	
		1	1				3		3	
	1	1					3	3	3	
	1									

Distance Measures

- For pixel p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a *distance function* or *metric* if
  (a) $D(p, q) \geq 0$ ($D(p, q) = 0$ iff $p = q$),
  (b) $D(p, q) = D(q, p)$, and
  (c) $D(p, z) \leq D(p, q) + D(q, z)$
- Types of distance
  - Euclidean distance
  - City-block distance
  - Chessboard distance
  - M-distance

Euclidean Distance

- The Euclidean distance between p and q is defined as

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

- For this distance measure, the 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).

Euclidean Distance

- The Euclidean distance between p and q is defined as

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

- For this distance measure, the 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).
- Set of points $S = \{q \mid D_e(p,q) \leq r\}$ are the points contained in a disk of radius r centered at p.

City-Block Distance

- Also called D_4-distance/*Manhattan distance*.
- City-block distance between p and q is defined as

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

- The pixels having a D_4 distance from (x, y) less than or equal to r form a diamond centered at (x, y).
- For example D_4 distance ≤ 2

			2			
		2	1	2		
	2	1	p	1	2	
		2	1	2		
			2			

Chessboard Distance

- Also called D_8 distance.
- Chessboard distance between p and q is defined as

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

- Set of points $S = \{q \mid D_8(p, q) \leq r\}$ forms a square centered at p
- For example D_8 distance ≤ 2

	2	2	2	2	2	
	2	1	1	1	2	
	2	1	p	1	2	
	2	1	1	1	2	
	2	2	2	2	2	

- The pixels with $D_8 = 1$ are the 8-neighbors of (x, y).

D_m-distance

- The D_4 and D_8 distances between p and q are independent of any path that might between the points.
- D_m distance between two points is defined as the shortest m-path between the points.
- The D_m-distance between two pixels will depend on the values of the pixels along the path, as well as the values of their neighbors.
- Assume that p, p_2, and p_4 have value 1 and that p_1 and p_3 can have a value of 0 or 1.

$$
\begin{array}{ccc}
 & p_3 & p_4 \\
p_1 & p_2 & \\
p & &
\end{array}
$$

D_m-distance

- Consider $V = \{1\}$.
- If p_1 and p_3 are 0, the length of the shortest m-path between p and p_4 is 2.
- If p_1 is 1, then p_2 and p will no longer be m-adjacent and the length of the shortest m-path becomes 3.
  - the path goes through the points $pp_1p_2p_4$
- If p_3 is 1 and p_1 is 0; the length of the shortest m-path also is 3.
- If both p_1 and p_3 are 1, the length of the shortest m-path between p and p_4 is 4.
  - The path goes through the sequence of points $pp_1p_2p_3p_4$

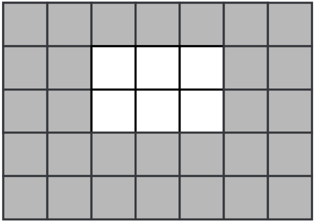
Arithmetic/Logical Operations

- Following Arithmetic/Logical Operations between two pixels p and q are used extensively

Arithmetic	Logical
$p + q$	$p \cdot q$
$p - q$	$p + q$
$p * q$	p'
$p\%q$	

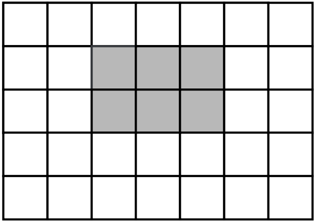
- Logical operations apply to binary images only $\Rightarrow$ Usually pixel by pixel.

Arithmetic/Logical Operations

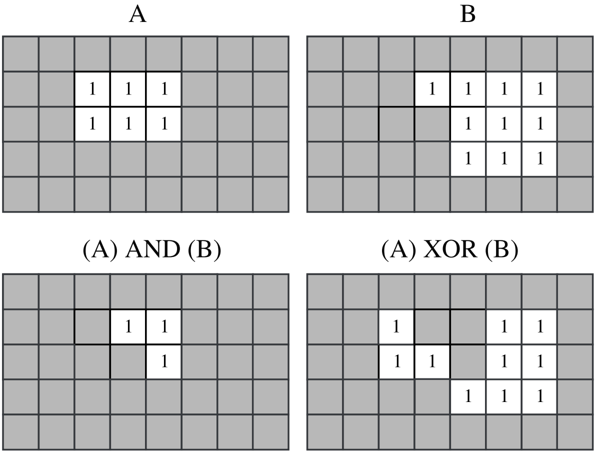


A

NOT(A)



Arithmetic/Logical Operations



Neighborhood Operations

- The value assigned to a pixel is a function of its gray label and the gray labels of its neighbors.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$z = \frac{1}{9}(z_1 + z_2 + \ldots + z_9) = \text{Average}$$

Template

- More general form

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$z = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9$$
$$= \sum_{i=1}^{9} w_i z_i$$

- Same as averaging if $w_i = \frac{1}{9}$

Neighborhood Operations

- Various important operations can be implemented by proper selection of coefficient w_i.
  - Noise filtering,
  - Thinning,
  - Edge detection, etc.

Quiz Questions

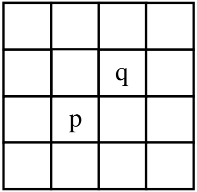
Question 01:

In the following figure which of the option are true?

(a) $q \in N_4(p)$

(b) $q \in N_8(p)$

(b) $q \in N_D(p)$



Quiz Questions

Question 02:

Find out

(a) Euclidean

(b) City Block

(b) Chess Board

distances between p and q in the given figures.

			q
p			

Quiz Questions

Question 02:

Find out

(a) Euclidean

(b) City Block

(b) Chess Board

distances between p and q in the given figures.

			q
p			

			q
p			

Quiz Questions

Question 03:

Consider the two image subset, S_1 and S_2, shown in the following figure. For $V = \{1\}$, determine whether these two subsets are

(a) 4-connected,

(b) 8-connected,

(c) m-connected

$$
\begin{array}{c|cccc|cccc|c}
\multicolumn{1}{c}{} & \multicolumn{4}{c}{S_1} & \multicolumn{4}{c}{S_2} & \multicolumn{1}{c}{} \\
\cline{2-9}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\cline{2-9}
\multicolumn{1}{c}{0} & 0 & 1 & 1 & \multicolumn{1}{c}{1} & 0 & 0 & 1 & \multicolumn{1}{c}{1} & 1
\end{array}
$$

Quiz Questions

Question 04:

Consider the image segment shown.

(a) Let $V = \{0, 1\}$ and compute the lengths of the shortest $4-$, $8-$, and $m-$path between p and q. If a particular path does not exist between these two points, explain why.

$$\begin{array}{rccl} & 3 & 1 & 2 \quad 1\,(q) \\ & 2 & 2 & 0 \quad 2 \\ & 1 & 2 & 1 \quad 1 \\ (p) & 1 & 0 & 1 \quad 2 \end{array}$$

Quiz Questions

Question 04:

Consider the image segment shown.

(a) Let $V = \{0, 1\}$ and compute the lengths of the shortest $4-$, $8-$, and $m-$path between p and q. If a particular path does not exist between these two points, explain why.

(b) Repeat for $V = \{1, 2\}$.

$$\begin{array}{rcccl} & 3 & 1 & 2 & 1\,(q) \\ & 2 & 2 & 0 & 2 \\ & 1 & 2 & 1 & 1 \\ (p) & 1 & 0 & 1 & 2 \end{array}$$

Thank You

Queries?