

---

# VC 12/13 – T5

## Single Pixel Manipulation

Mestrado em Ciência de Computadores  
Mestrado Integrado em Engenharia de Redes e  
Sistemas Informáticos

*Miguel Tavares Coimbra*

---

# Outline

---

- Dynamic Range Manipulation
- Neighborhoods and Connectivity
- Image Arithmetic
- Example: Background Subtraction

---

# Topic: Dynamic Range Manipulation

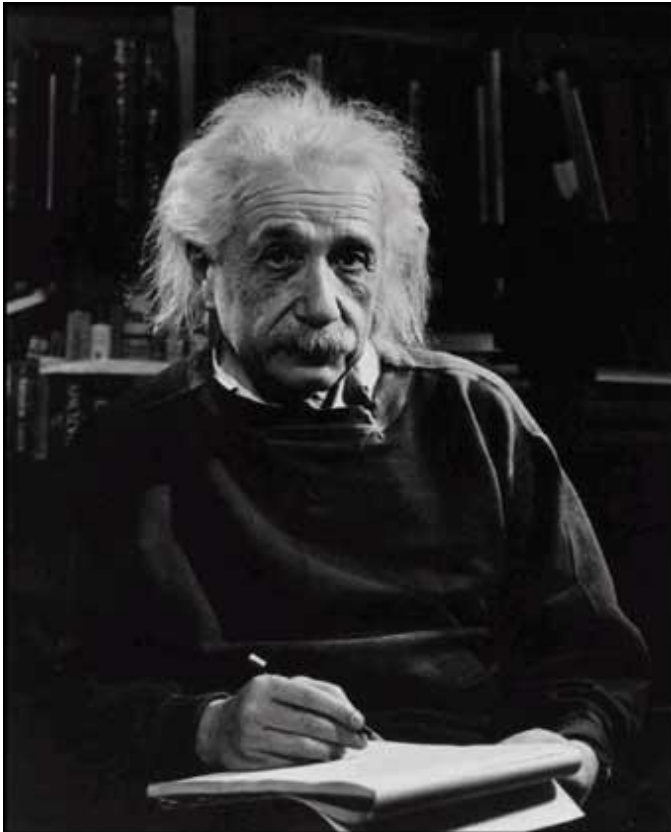
---

- **Dynamic Range Manipulation**
- Neighborhoods and Connectivity
- Image Arithmetic
- Example: Background Subtraction

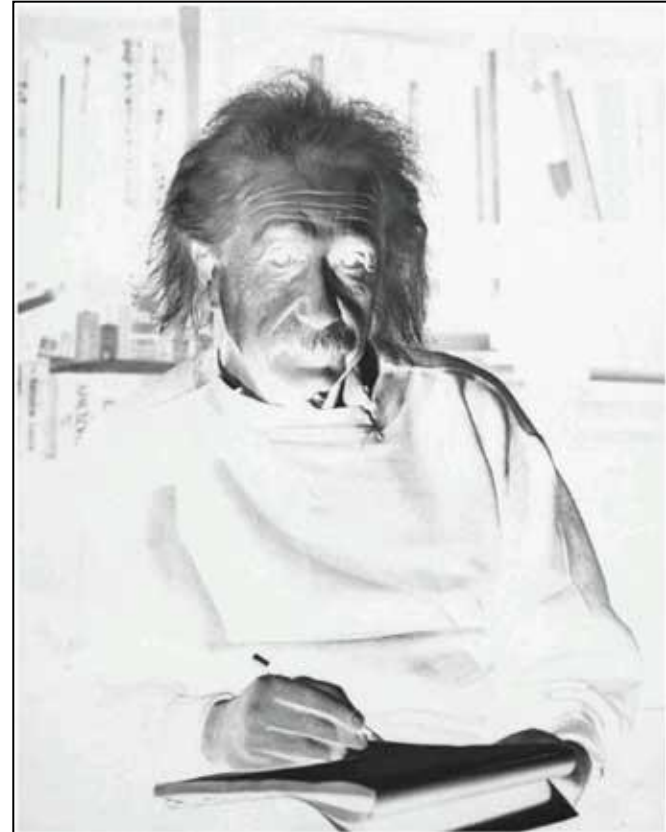
---

# Manipulation

---

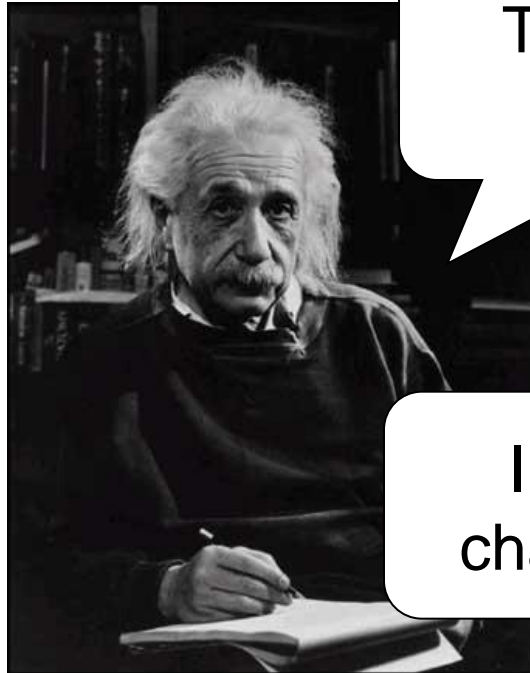


What I see



What I want to see

# Digital Images



To change this...

I need to change this!

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

What we see

What a computer sees

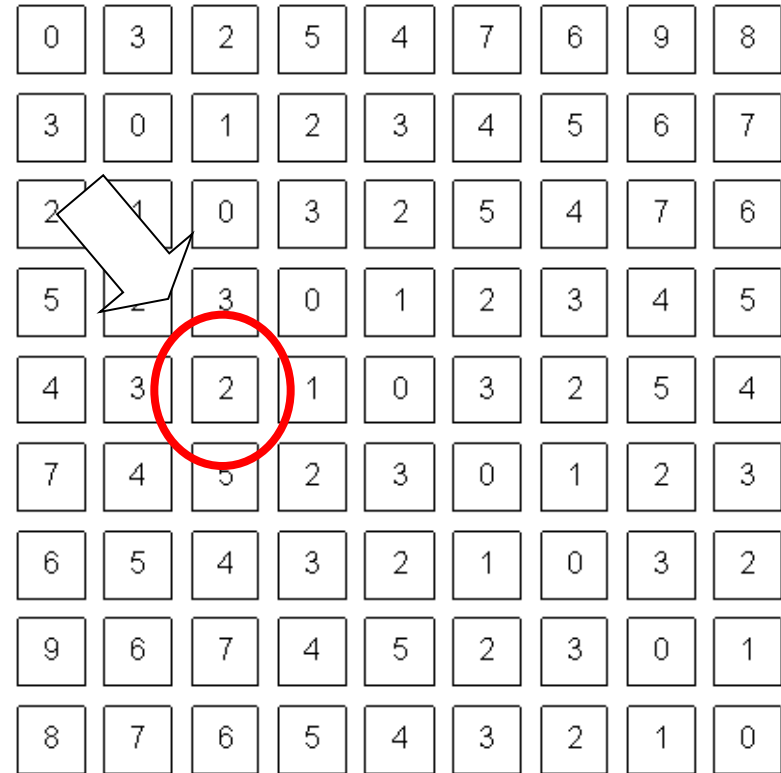
# Pixel Manipulation

- Let's start simple
- I want to change a single Pixel.

$$f(X, Y) = MyNewValue$$

- Or, I can apply a transformation  $T$  to all pixels individually.

$$g(x, y) = T[f(x, y)]$$



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

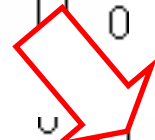
# Image Domain

- I am directly changing values of the image matrix.

$$g = T(f)$$

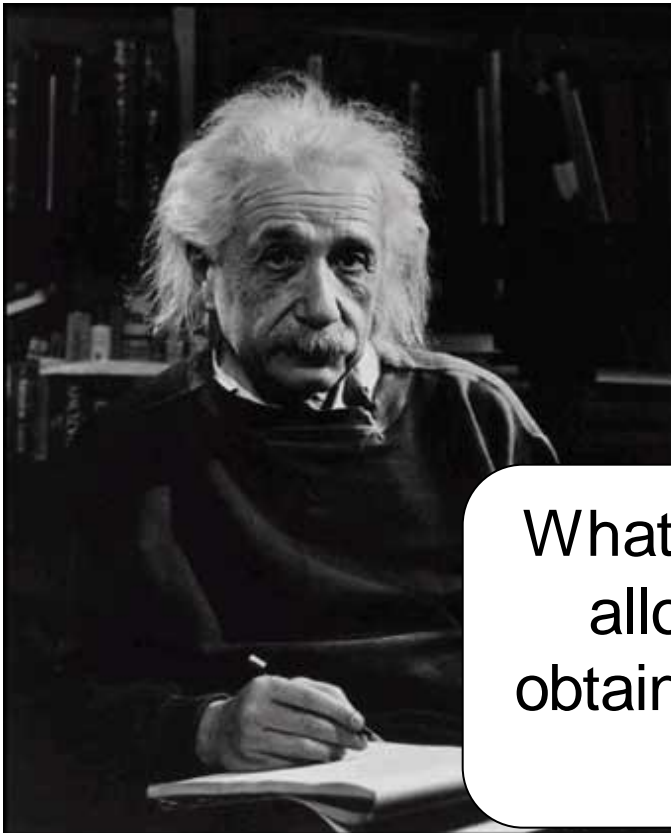
- Image Domain
- So, what is the other possible 'domain'?

0	1	2	3
1	0	3	2
2	3	0	1
3	2	1	0



0	1	2	3
1	3	3	2
2	3	3	1
3	2	1	0

# Image Negative



What I see



What I want to see

What operation **T**  
allows me to  
obtain this result?

$$g=T(f)$$

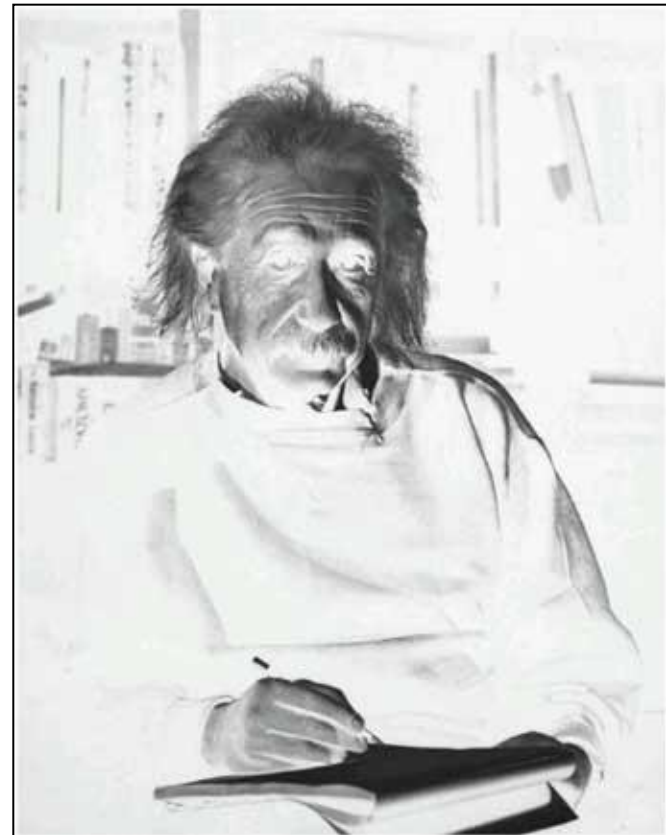


# Image Negative

- Consider the maximum value allowed by quantization (*max*).
- For 8 bits: 255
- Then:

$$g(x, y) = \text{max} - f(x, y)$$

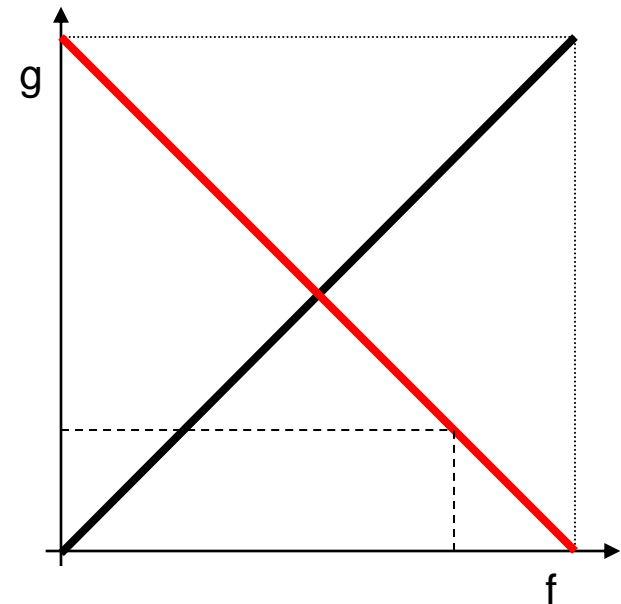
$$g(x, y) = 255 - f(x, y)$$



What I want to see

# Dynamic Range Manipulation

- What am I really doing?
  - Changing the response of my image to the received brightness.
- Dynamic Range Manipulation
  - Represented by a 2D Plot.

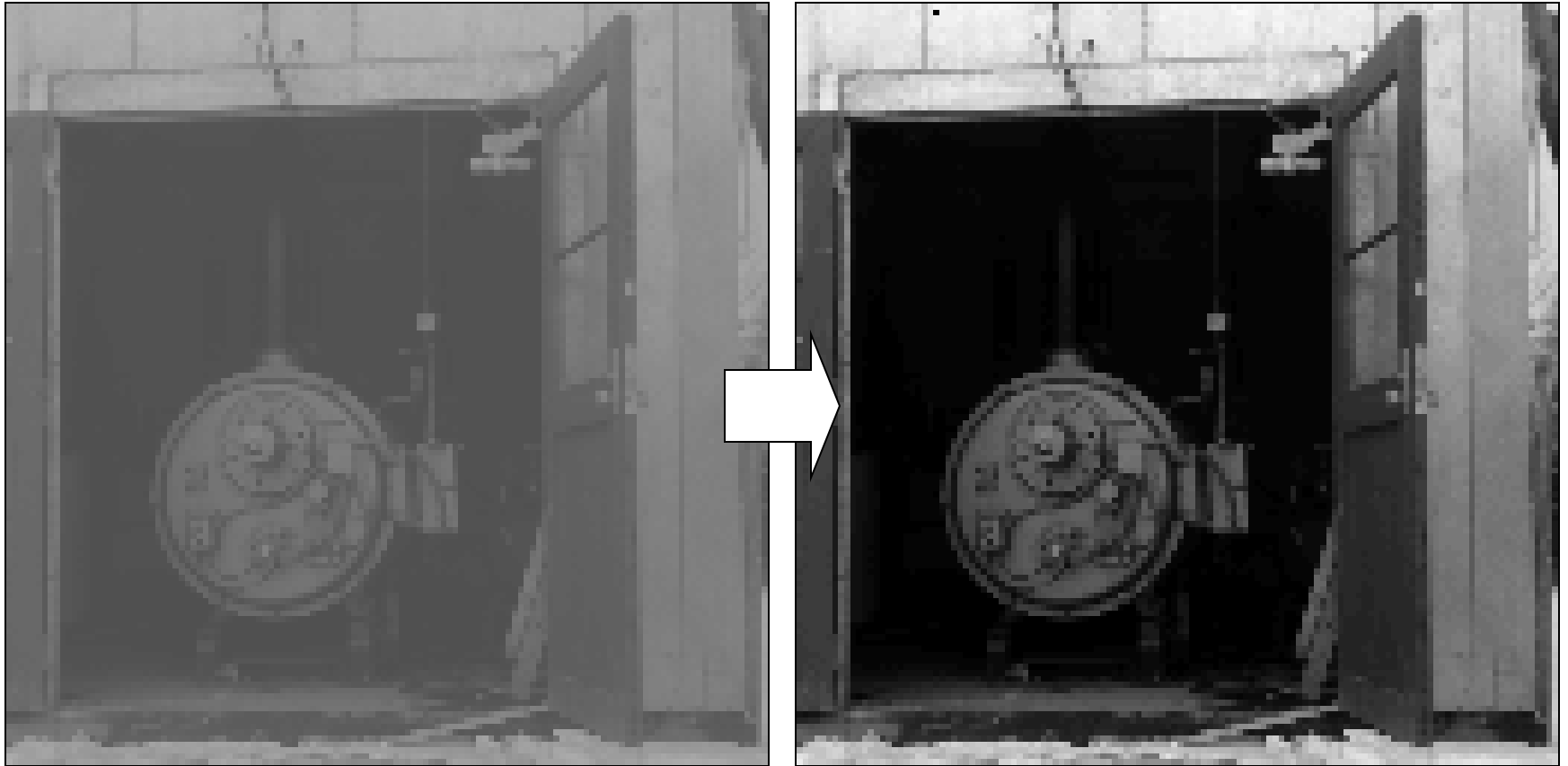


— Normal  
— Inverted

---

# Why DRM?

---



---

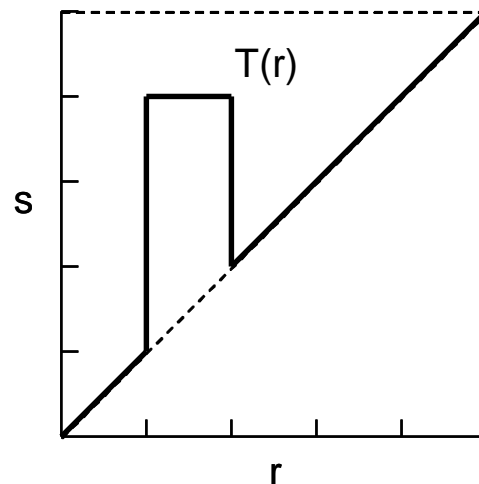
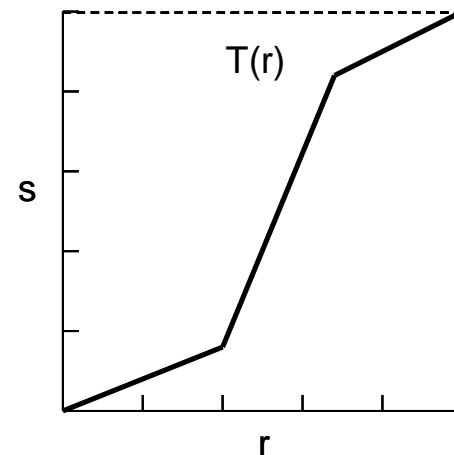
# Why DRM?

---



# Other DRM functions

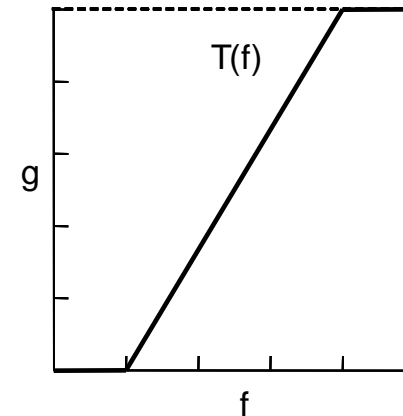
- By manipulating our function we can:
  - Enhance generic image visibility.
  - Enhance specific visual features.
  - Use quantization space a lot better.



# Contrast Stretching

- ‘Stretches’ the dynamic range of an image.
- Corrects some image capture problems:
  - Poor illumination, aperture, poor sensor performance, etc.

$$g = \max \frac{f - \min}{\max - \min}$$



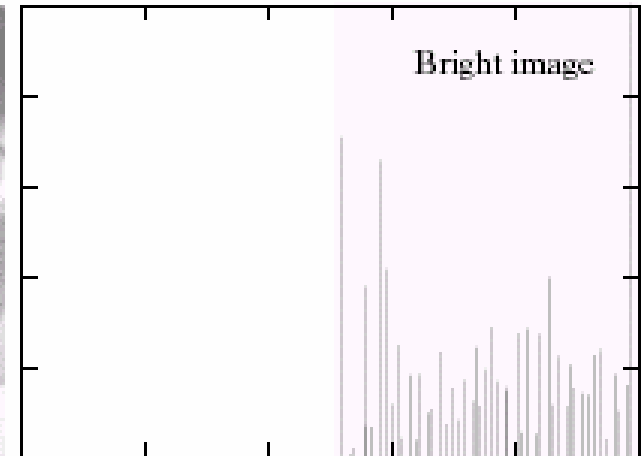
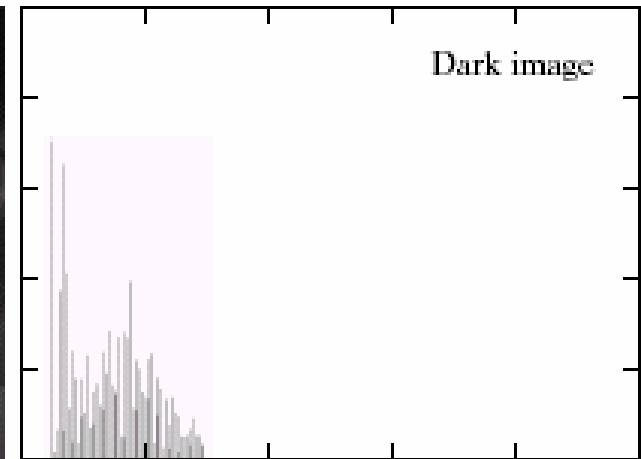
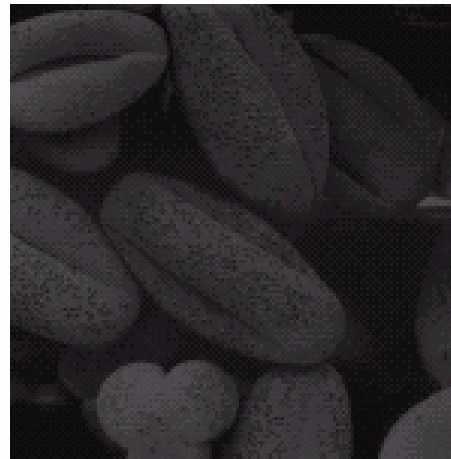
# Histogram Processing

- Histograms give us an idea of how we are using our dynamic range



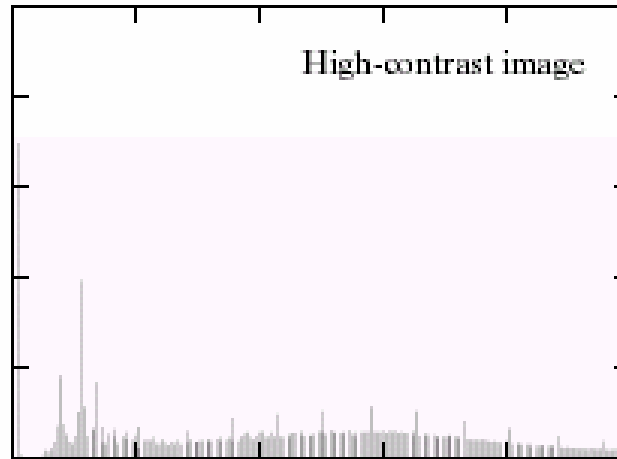
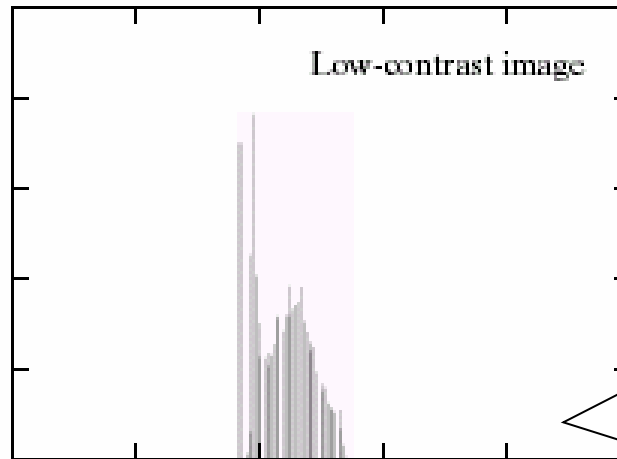
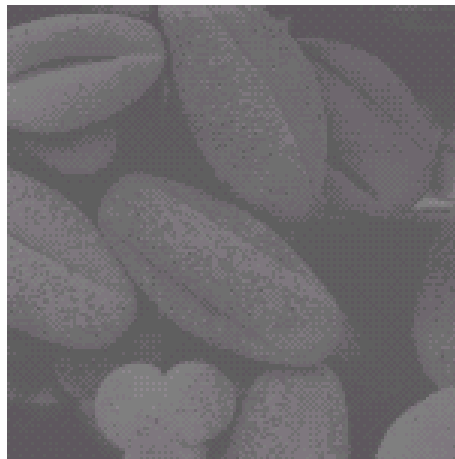
# Types of Image Histograms

- Images can be classified into types according to their histogram
  - Dark
  - Bright
  - Low-contrast
  - High-contrast





# Types of Image Histograms

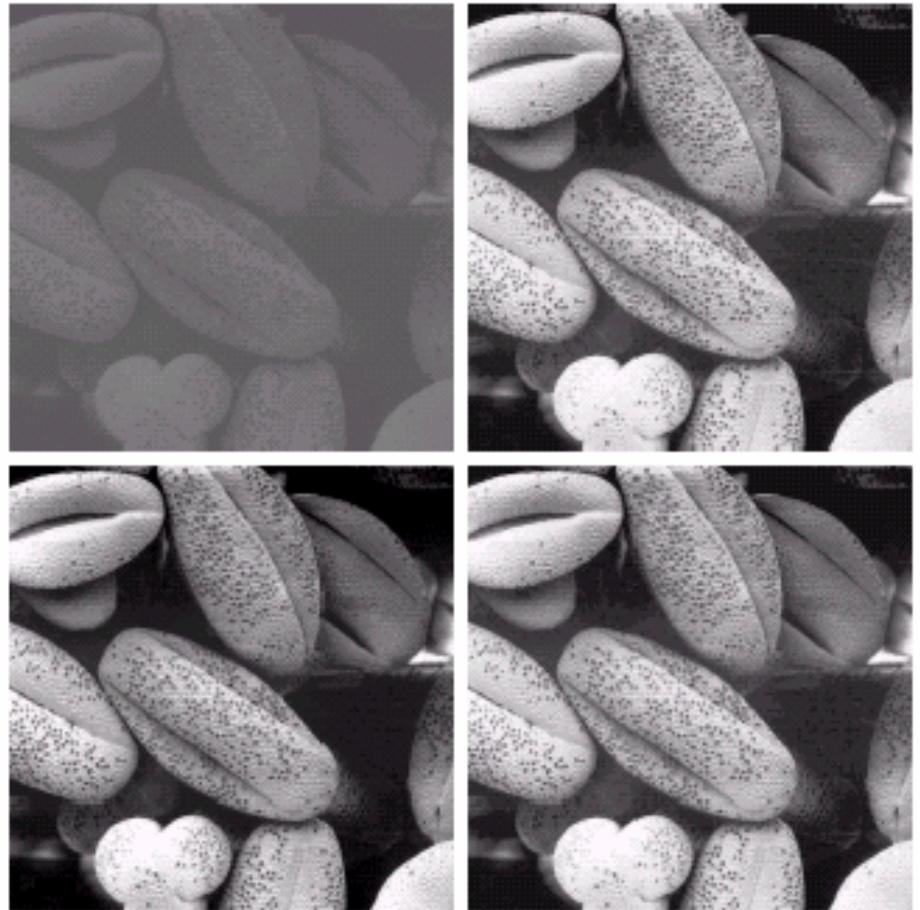


I can  
manipulate  
this using  
single Pixel  
operations!

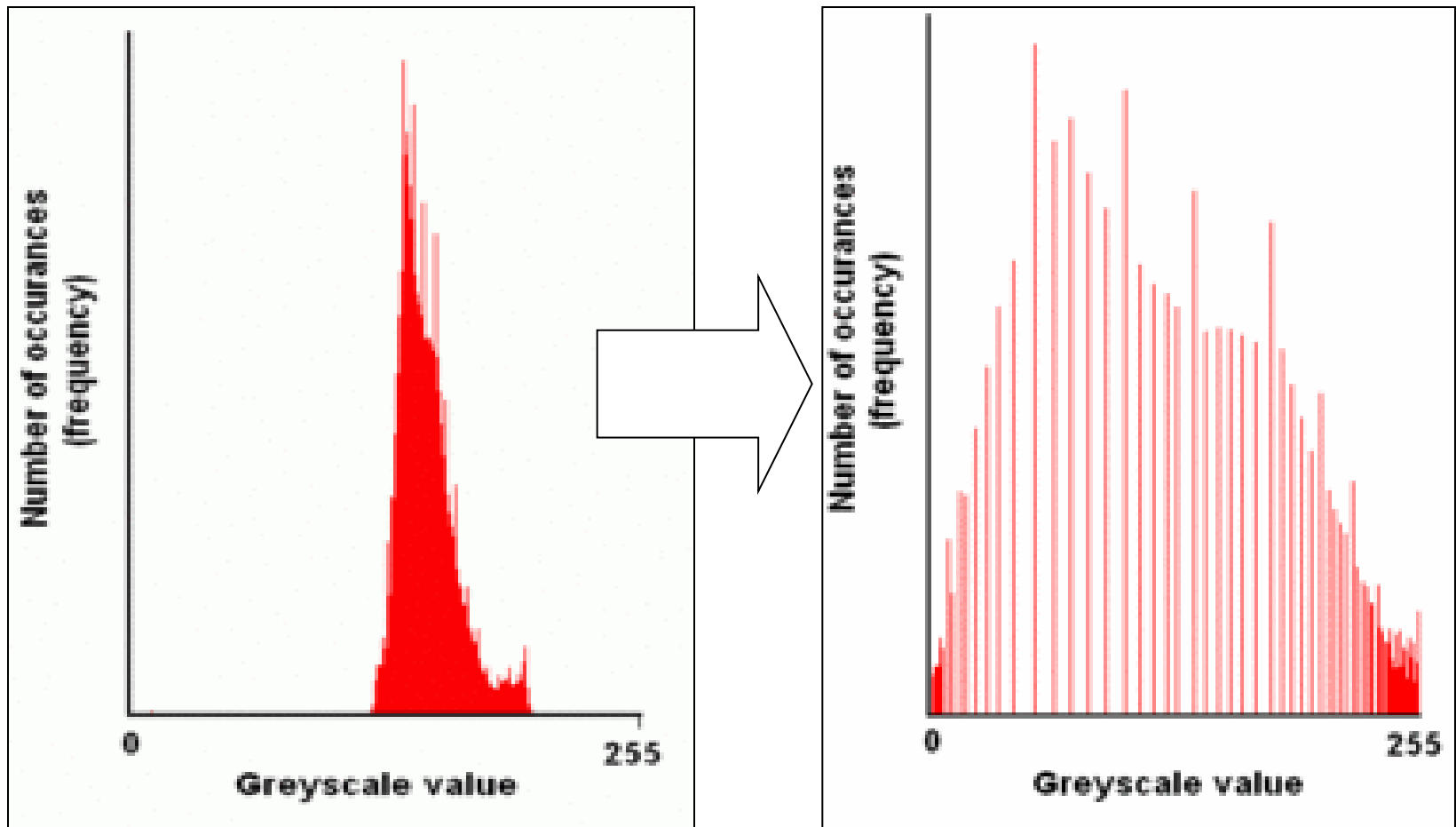
# Histogram Equalization

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j)$$

- **Objective:**
  - Obtain a ‘flat’ histogram.
  - Enhance visual contrast.
- **Digital histogram**
  - Result is a ‘flat-ish’ histogram.
  - Why?



# Histogram Equalization



---

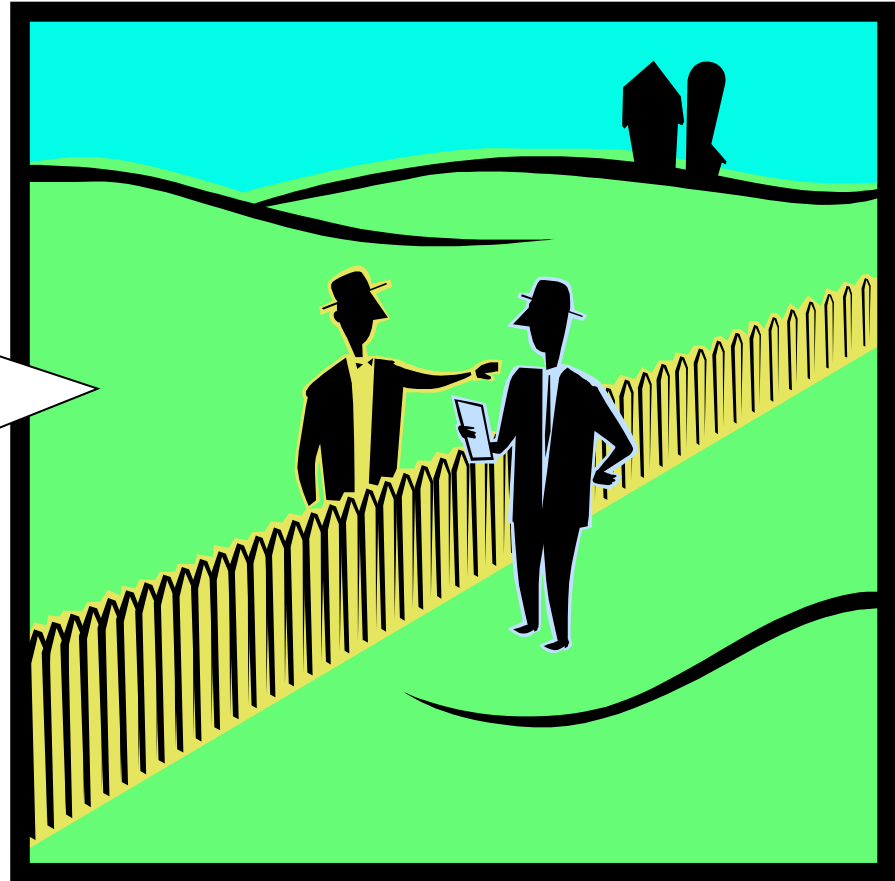
# Topic: Neighborhoods and Connectivity

---

- Dynamic Range Manipulation
- **Neighborhoods and Connectivity**
- Image Arithmetic
- Example: Background Subtraction

# Neighbors

Why do we care at all?

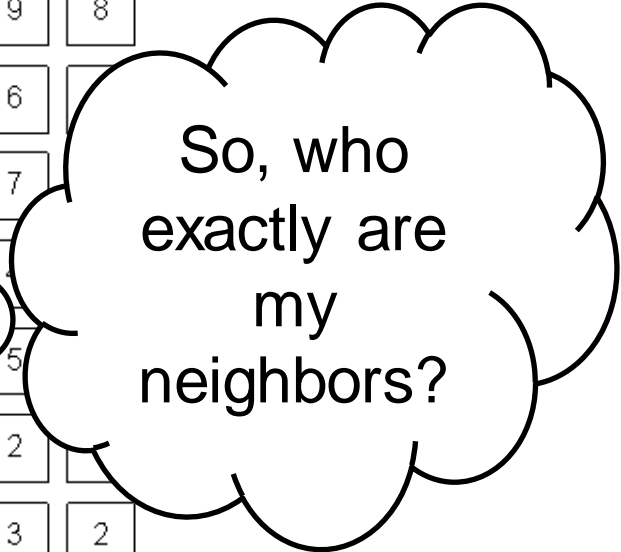


---

# Digital Images

---

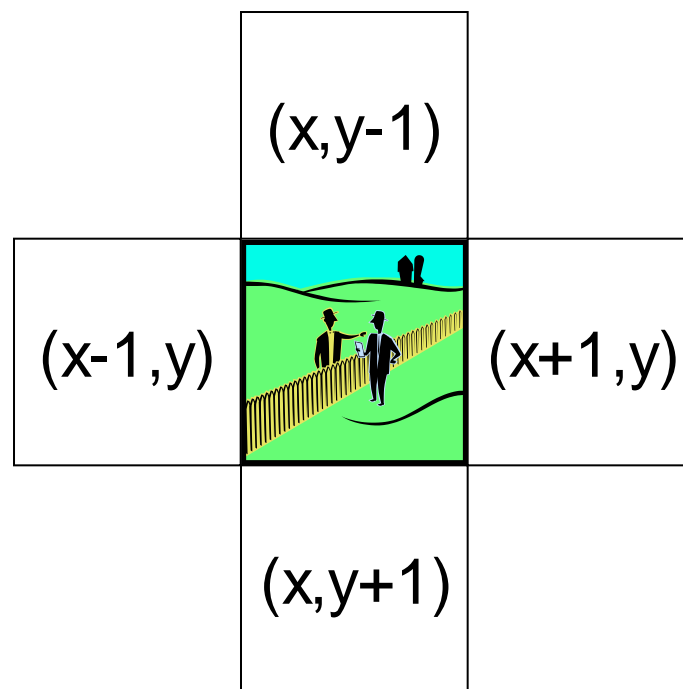
0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0



What a computer sees

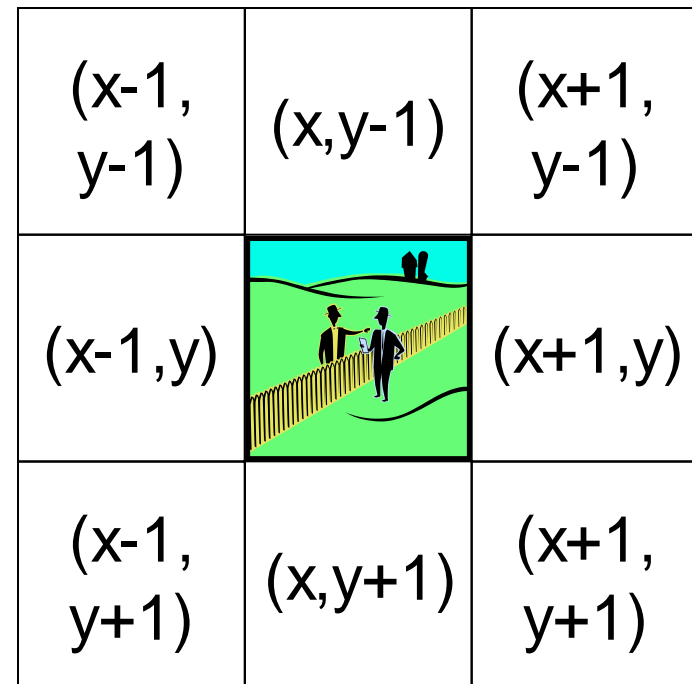
# 4-Neighbors

- A pixel  $p$  at  $(x,y)$  has 2 horizontal and 2 vertical neighbors:
  - $(x+1,y)$ ,  $(x-1,y)$ ,  
 $(x,y+1)$ ,  $(x,y-1)$
  - $N_4(p)$ : Set of the 4-neighbors of  $p$ .
- Limitations?



# 8-Neighbors

- A pixel has 4 diagonal neighbors
  - $(x+1, y+1)$ ,  $(x+1, y-1)$ ,  
 $(x-1, y+1)$ ,  $(x-1, y-1)$
  - $N_D(p)$ : Diagonal set of neighbors
- $N_8(p) = N_4(p) + N_D(p)$
- Limitations?



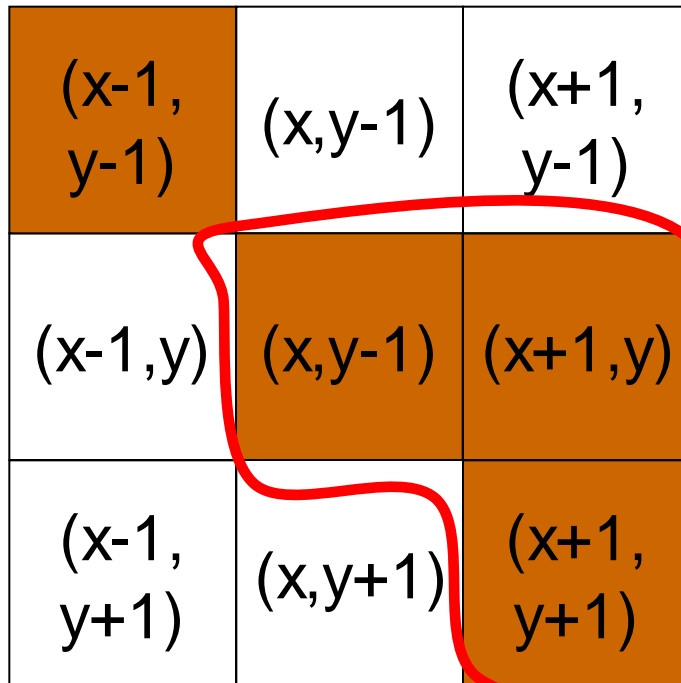


# Connectivity

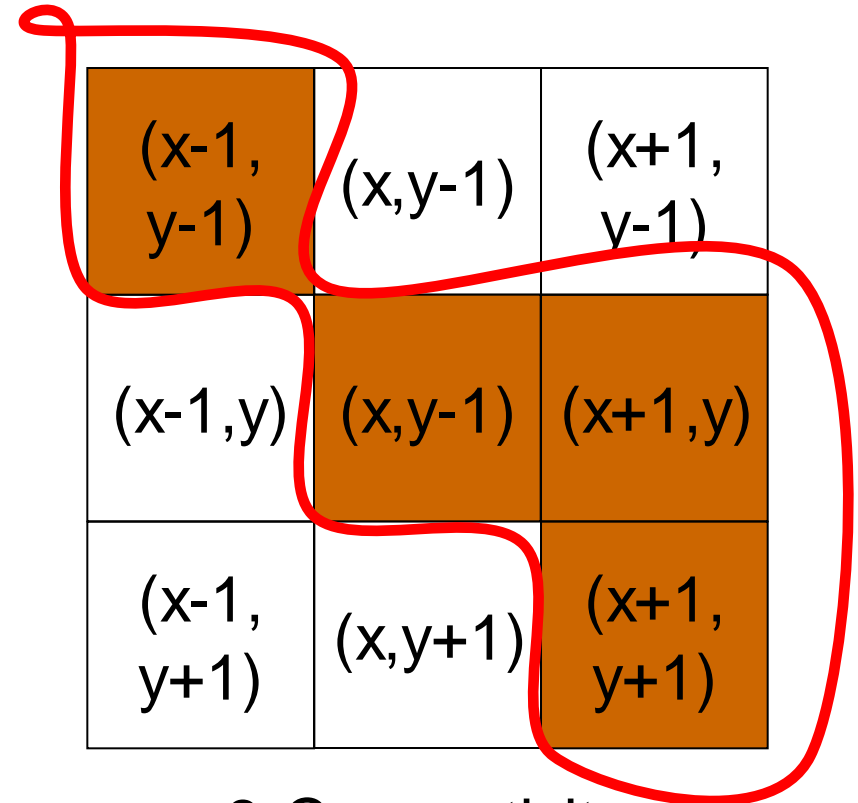
- Two pixels are connected if:
  - They are neighbors (i.e. adjacent in some sense -- e.g.  $N_4(p)$ ,  $N_8(p)$ , ...)
  - Their gray levels satisfy a specified criterion of similarity (e.g. equality, ...)

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

# 4 and 8-Connectivity

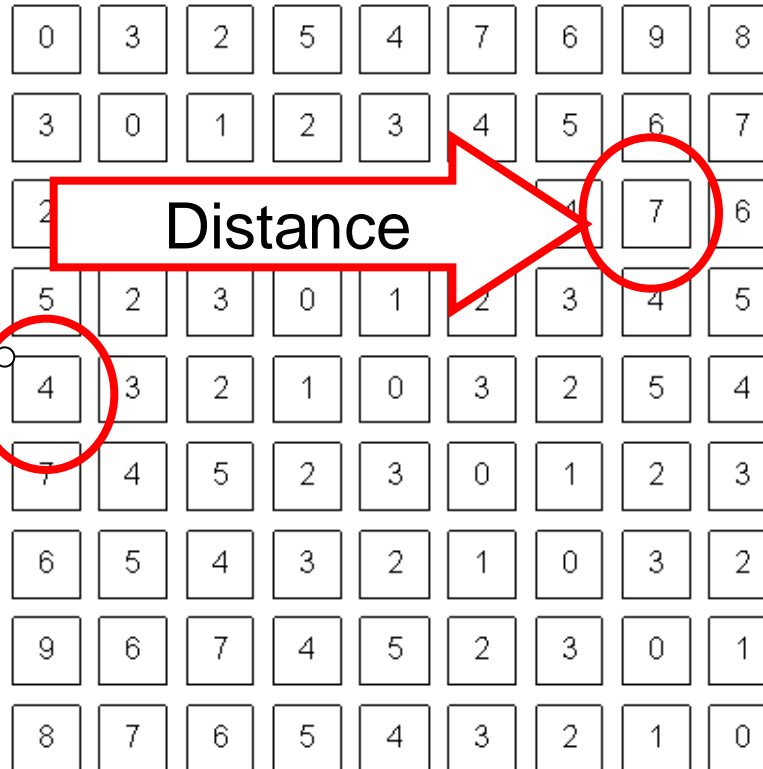
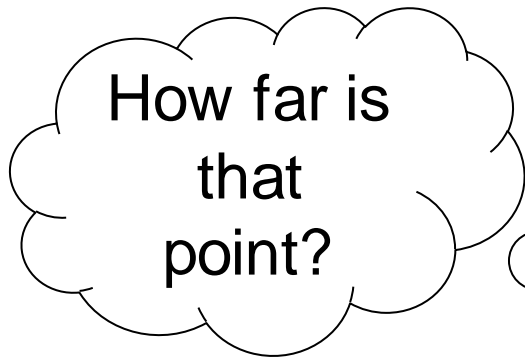


4-Connectivity



8-Connectivity

# Distances



---

# D4 Distance

---

- $D_4$  distance (city-block distance):

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

- forms a diamond centered at  $(x,y)$

- e.g. pixels with  $D_4 \leq 2$  from  $p$

```
      2
     2 1 2
    2 1 0 1 2
     2 1 2
      2
```

$D_4 = 1$  are the 4-neighbors of  $p$

---

# D8 Distance

---

- $D_8$  distance (chessboard distance):
  - $D_8(p,q) = \max(|x-s|, |y-t|)$
  - Forms a square centered at  $p$
  - e.g. pixels with  $D_8 \leq 2$  from  $p$

```
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
```

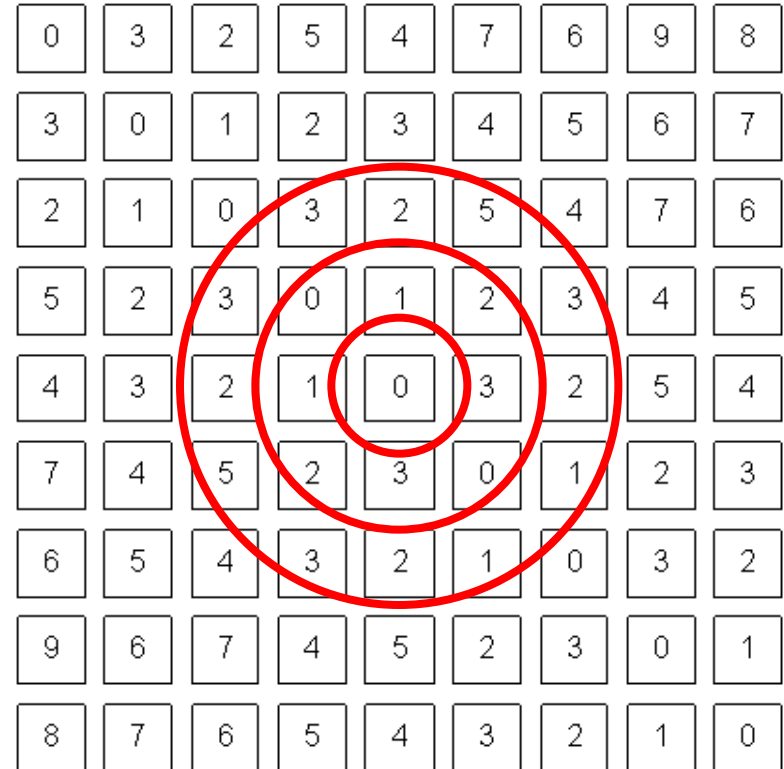
$D_8 = 1$  are the 8-neighbors of  $p$

# Euclidean Distance

- Euclidean distance:

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

- Points (pixels) having a distance less than or equal to  $r$  from  $(x,y)$  are contained in a disk of radius  $r$  centered at  $(x,y)$ .



---

# Topic: Image Arithmetic

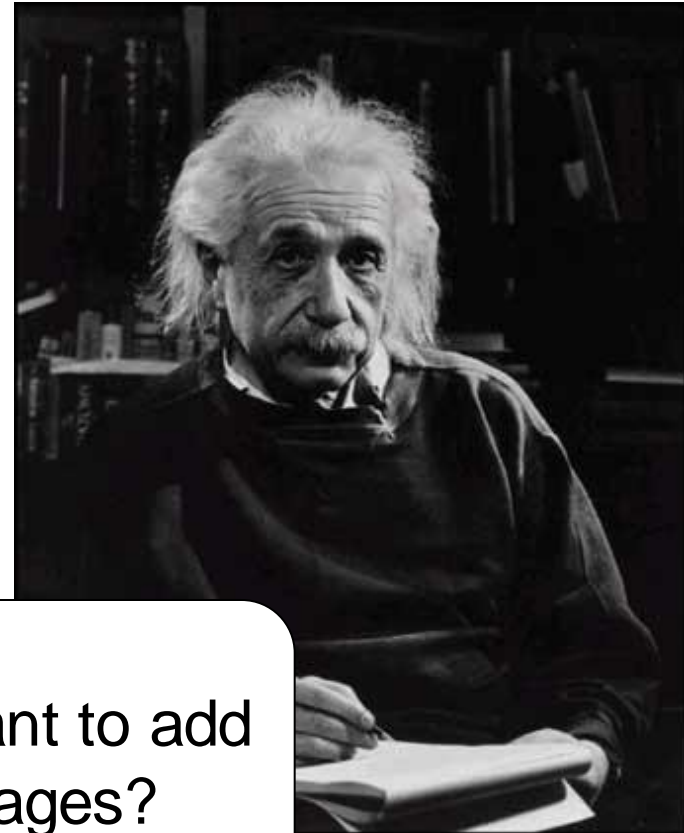
---

- Dynamic Range Manipulation
- Neighborhoods and Connectivity
- **Image Arithmetic**
- Example: Background Subtraction

---

# Arithmetic operations between images

---



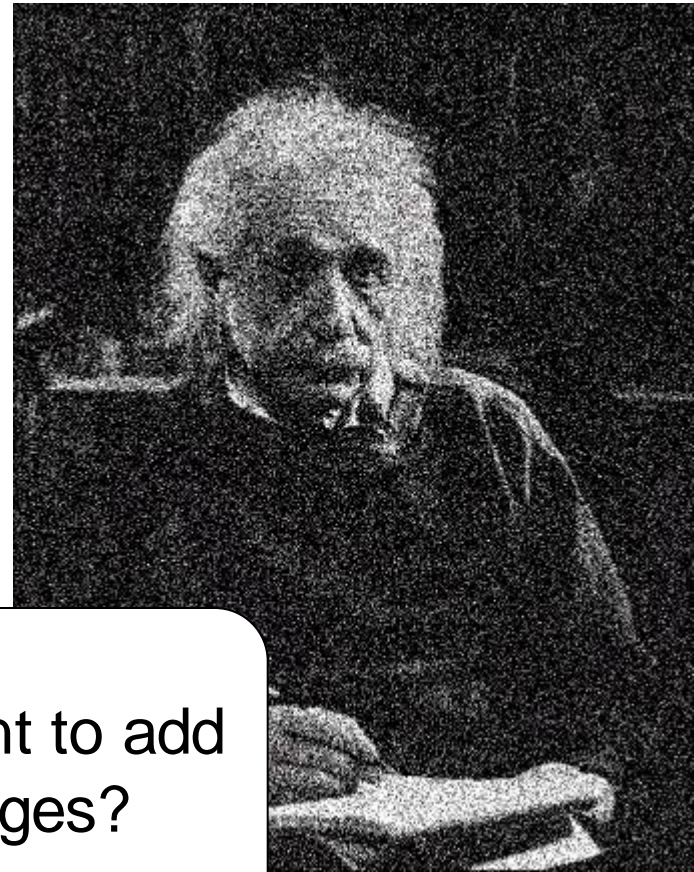
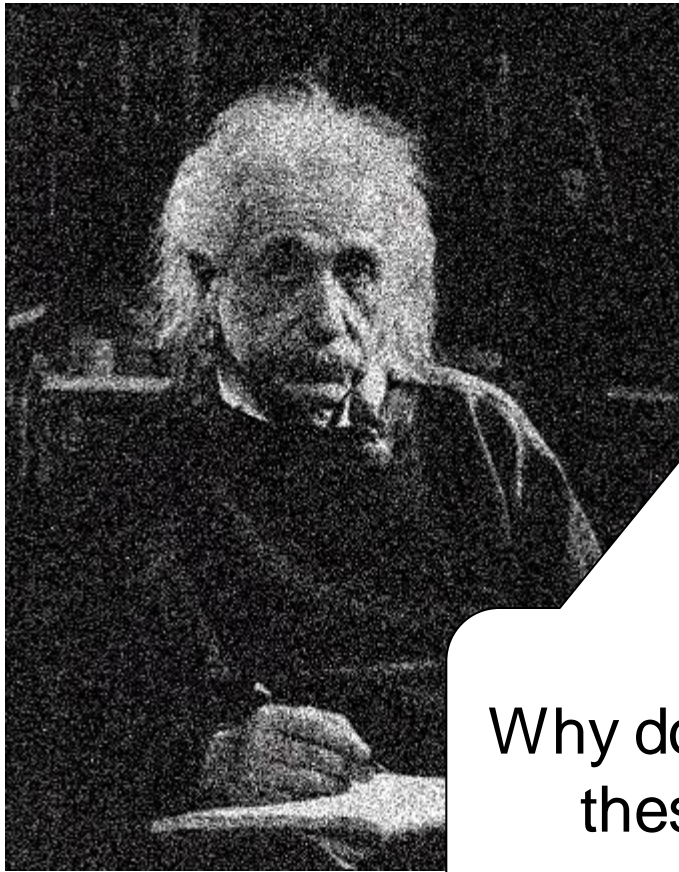
Why do I want to add these images?



---

# Arithmetic operations between images

---

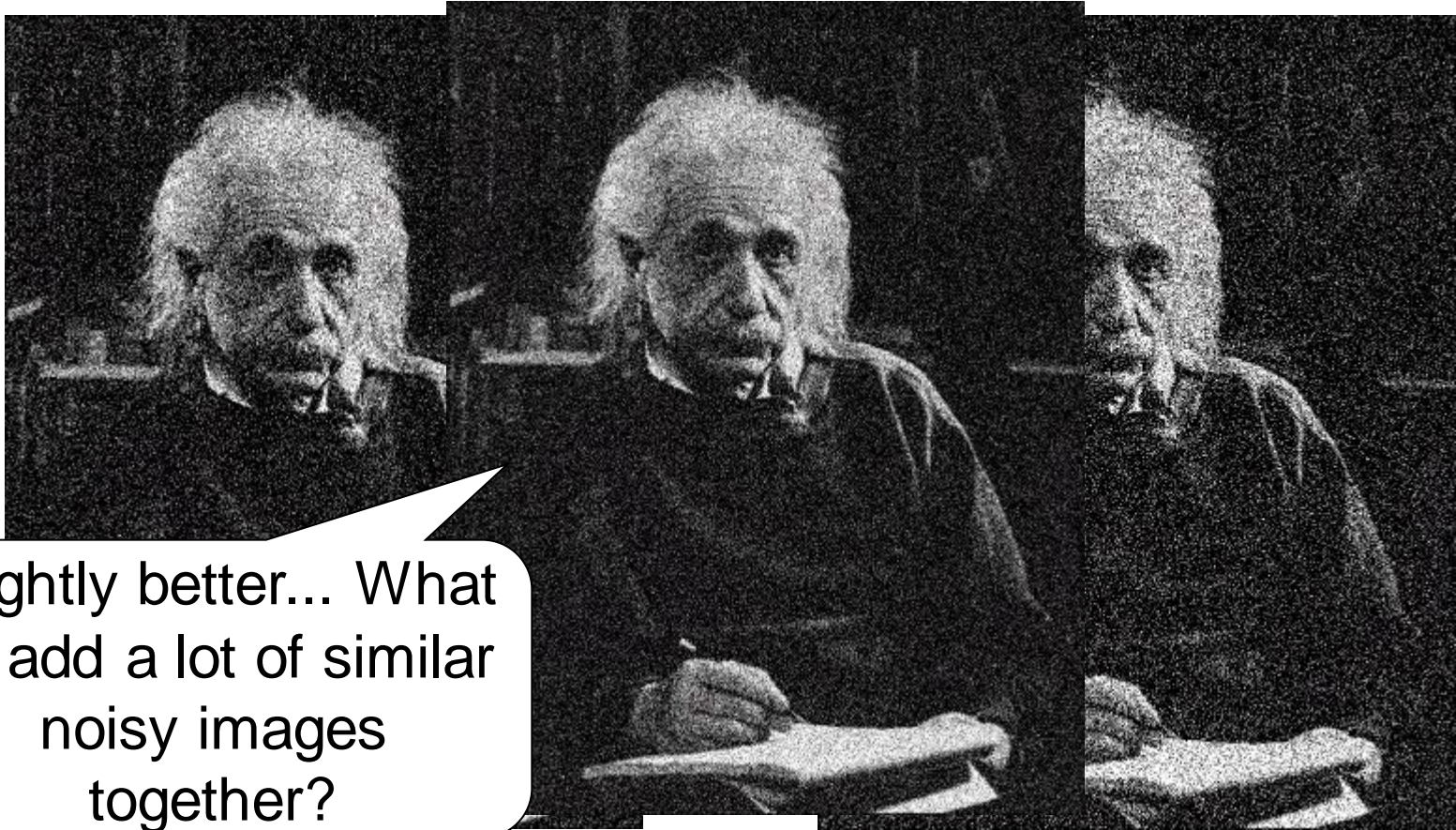


Why do I want to add these images?

---

# Arithmetic operations between images

---



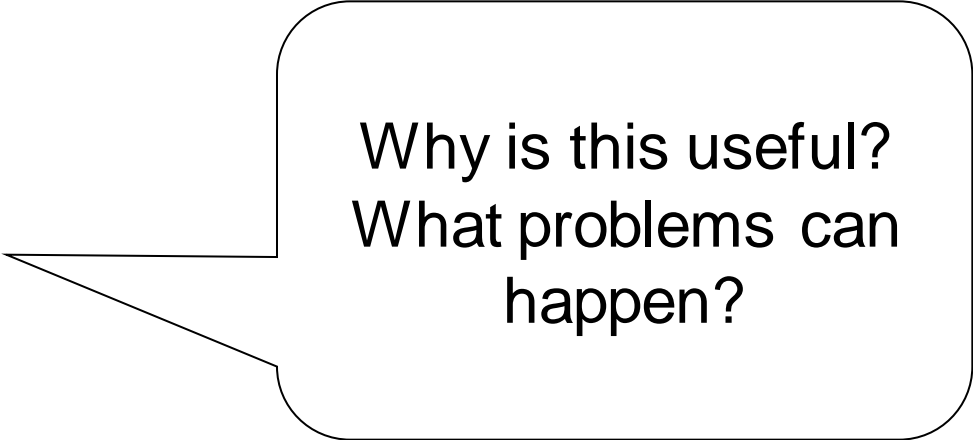
Slightly better... What if I add a lot of similar noisy images together?

---

# Image Arithmetic

---

- Image 1:  $a(x,y)$
- Image 2:  $b(x,y)$
- Result:  $c(x,y) = a(x,y)$  OPERATION  $b(x,y)$
- Possibilities:
  - Addition
  - Subtraction
  - Multiplication
  - Division
  - Etc..



Why is this useful?  
What problems can  
happen?

# Example

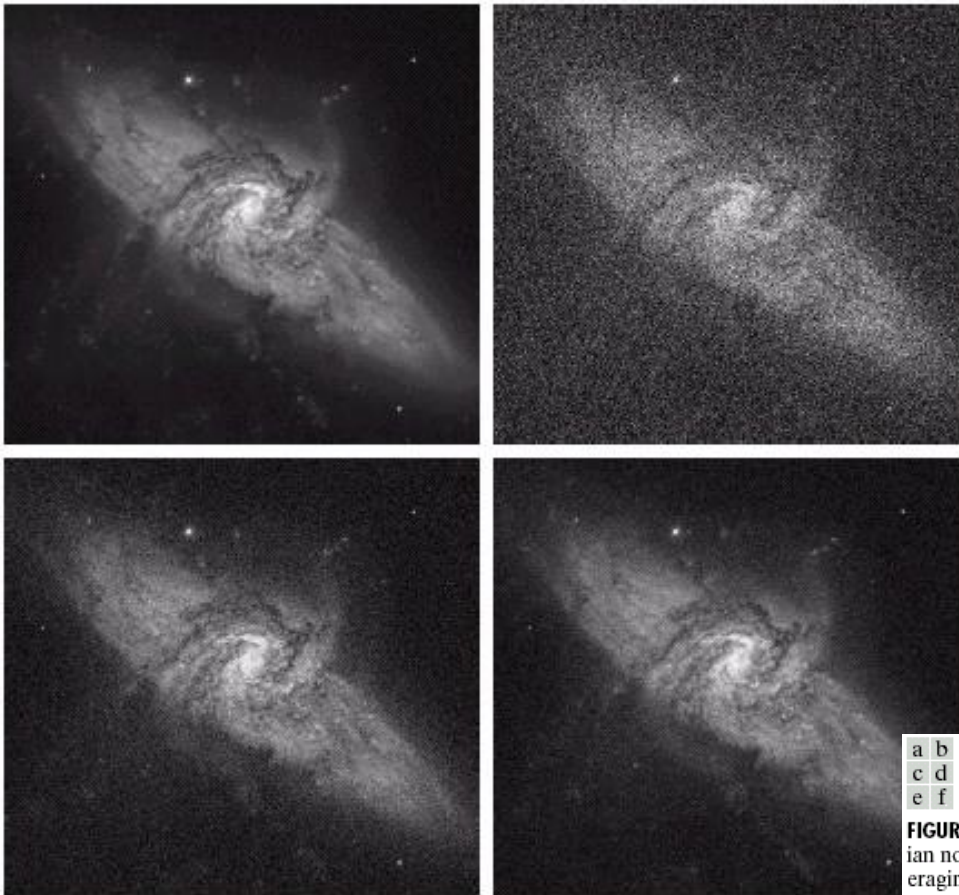


Image  
Addition

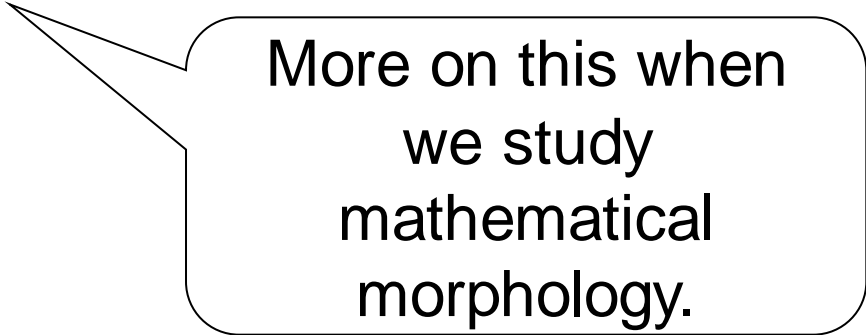
**FIGURE 3.30** (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging  $K = 8, 16, 64,$  and  $128$  noisy images. (Original image courtesy of NASA.)

---

# Logic Operations

---

- Binary Images
- We can use Boolean Logic
- Operations:
  - AND
  - OR
  - NOT



More on this when  
we study  
mathematical  
morphology.

---

# Topic: Example: Background Subtraction

---

- Dynamic Range Manipulation
- Neighborhoods and Connectivity
- Image Arithmetic
- **Example: Background Subtraction**

---

# Example: Background Subtraction

---

- Image arithmetic is simple and powerful.

Is there a  
person  
here?  
Where?



---

# Background Subtraction

---

- Remember: We can only see numbers!

Is there a  
person  
here?  
Where?

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



---

# Background Subtraction

---

- What if I know this?



---

# Background Subtraction

---

- Subtract!
- Limitations?



---

# Background Subtraction

---

- **Objective:**
  - Separate the foreground objects from a static background.
- **Large variety of methods:**
  - Mean & Threshold [CD04]
  - Normalized Block Correlation [Mats00]
  - Temporal Derivative [Hari98]
  - Single Gaussian [Wren97]
  - Mixture of Gaussians [Grim98]

Segmentation!!  
More on this  
later.

---

# Resources

---

- R. Gonzalez, and R. Woods – Chapter 2
- R. Gonzalez, and R. Woods – Chapter 4
- K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: Principles and practice of background maintenance”, in Proc. of IEEE ICCV, Corfu, Greece, 1999.