Digital Image Processing

COSC 6380/4393

Lecture – 11

Sept. 26th, 2023

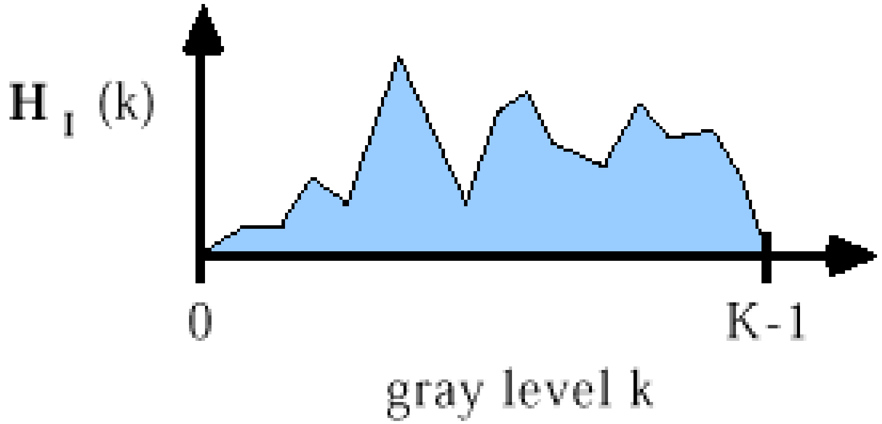
Pranav Mantini

Slides from Dr. Shishir K Shah and Frank (Qingzhong) Liu

Point Operations

SIMPLE HISTOGRAM OPERATIONS

- Recall: the gray-level histogram $\mathbf{H_I}$ of an image **I** is a graph of the frequency of occurrence of each gray level in **I**
- $\mathbf{H_I}$ is a one-dimensional function with domain 0, ... , K-1 :
- $\mathbf{H_I}$ (k) = n if gray-level k occurs (exactly) n times in **I**, for each k = 0, ... K-1



SIMPLE HISTOGRAM OPERATIONS

- The histogram $\mathbf{H_I}$ contains **no spatial information** about **I** - only information about the relative frequency of intensities
- Nevertheless
  - Useful information can be obtained from the histogram
  - Image quality is effected (enhanced, modified) by altering the histogram

Average Optical Density

- A measure of the average intensity of the image $\mathbf{I}$:

$$\mathrm{AOD}(\mathbf{I}) = \left(\frac{1}{N^2}\right) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i,j) = \left(\frac{1}{N^2}\right) \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} I(i,j)$$

- Can compute it from the histogram as well:

Average Optical Density

- A measure of the average intensity of the image **I**:

$$\mathrm{AOD}(\mathbf{I}) = \left(\frac{1}{N^2}\right)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} I(i,j) = \left(\frac{1}{N^2}\right)\sum_{j=0}^{N-1}\sum_{i=0}^{N-1} I(i,j)$$

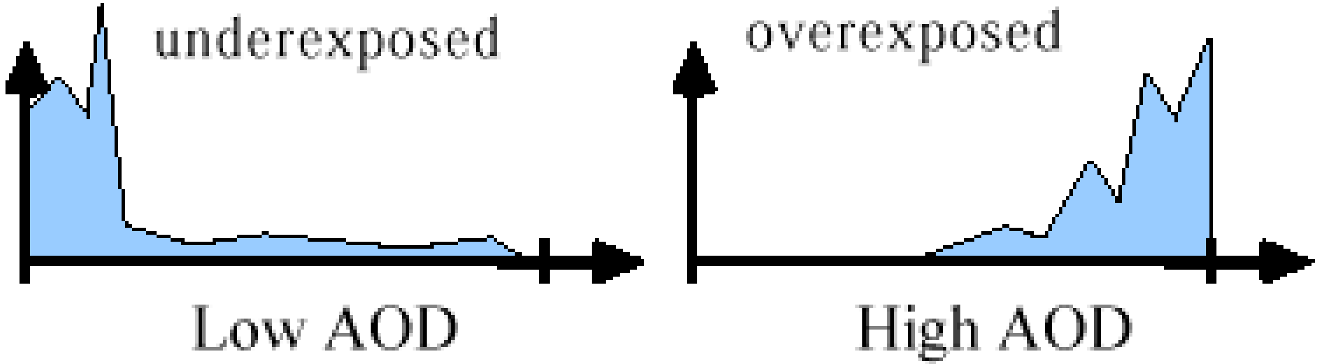
- Can compute it from the histogram as well:

$$\left(\frac{1}{N^2}\right)\sum_{k=0}^{K-1} k\mathbf{H}_{\mathbf{I}}(k)$$

- k^{th} term = (brightness level k) x (# occurrences of k)

Average Optical Density

- Examining the histogram can reveal possible errors in the imaging process:



- Methods for correcting such errors utilize the histogram
- The histogram will arise throughout this lecture

POINT OPERATIONS

- A **point operation** on an image **I** is a **function** f that maps **I** to another image **J** by operating on **individual pixels** in **I**:

$$J(i, j) = f[I(i, j)], \; 0 <= i, j <= N-1$$

- The same function f is applied at every image coordinate
- This is different from **local operations** such as OPEN, CLOSE, etc., since those are functions of both I(i, j) **and its neighbors**

POINT OPERATIONS

- Point operations **do not** modify **spatial relationships** between pixels
- They **do** modify the **image histogram**, and therefore the overall appearance of the image

LINEAR POINT OPERATIONS

- **Linear point operations** are the simplest class of point operations

$$F(X) = P.X + L$$

Image Offset

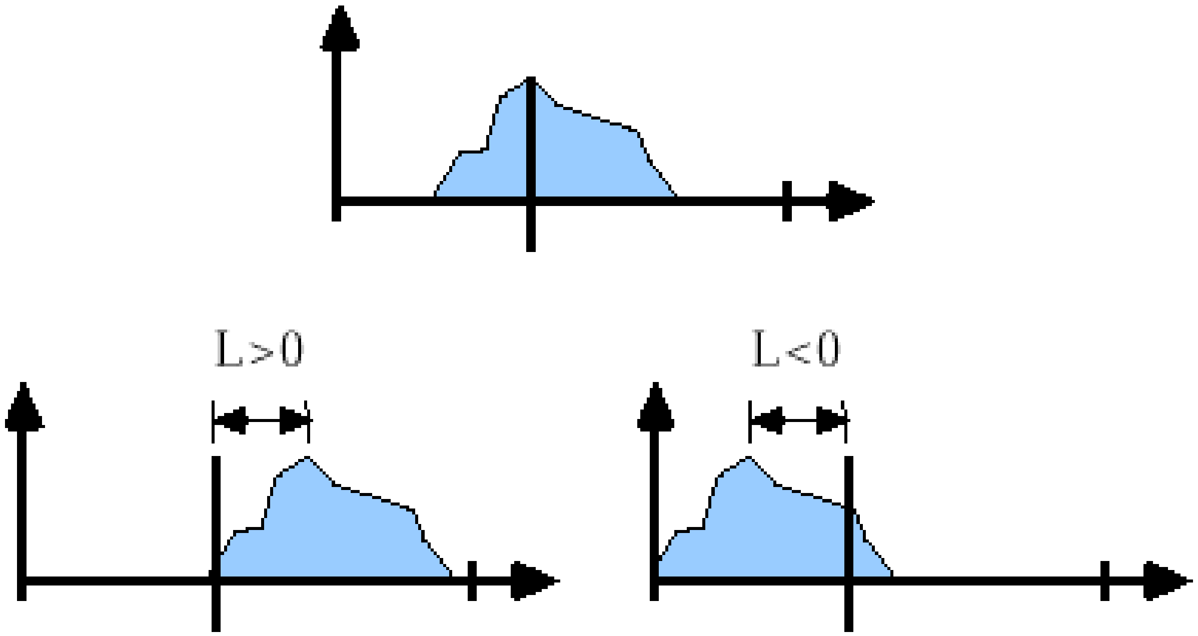
- Suppose L falls in the range -(K-1) <= L <= K-1 (± the nominal gray scale)
- An **additive image offset** is defined by the function

  J(i, j) = I(i, j) + L, for 0 <= i, j <= N-1

- Thus, the same constant L is added to every image pixel value
- If L > 0, **J** will be a **brightened** version of the image **I**
- Otherwise its appearance will be essentially the same

Image Offset

- If $L < 0$, **J** will be a **dimmed** version of the image **I**
- Adding offset L **shifts** the histogram by amount L to left or right:



Histograms of additive image offsets

- The input and output histograms are related by:

$$\mathbf{H}_\mathbf{J}(k) = \mathbf{H}_\mathbf{I}(k-L)$$

Image Offset Example

- Suppose it is desired to compare multiple images I_1, $\mathbf{I}_2$,..., $\mathbf{I}_n$ of the same scene
- However, the images were taken with a variety of different exposures or lighting conditions
- One solution: **equalize** the AOD's of the images
- If the gray-scale range of the images is 0 ,..., K-1, a reasonable AOD is K/2
- Let L_m = AOD(I_m), for m = 1 ,..., n
- Then define "AOD-equalized" images J_1, $\mathbf{J}_2$,..., $\mathbf{J}_n$ according to
  $J_m(i, j) = I_m(i, j) - L_m + K/2$, for $0 <= i, j <= N-1$

Image Offset Example

- The effect:

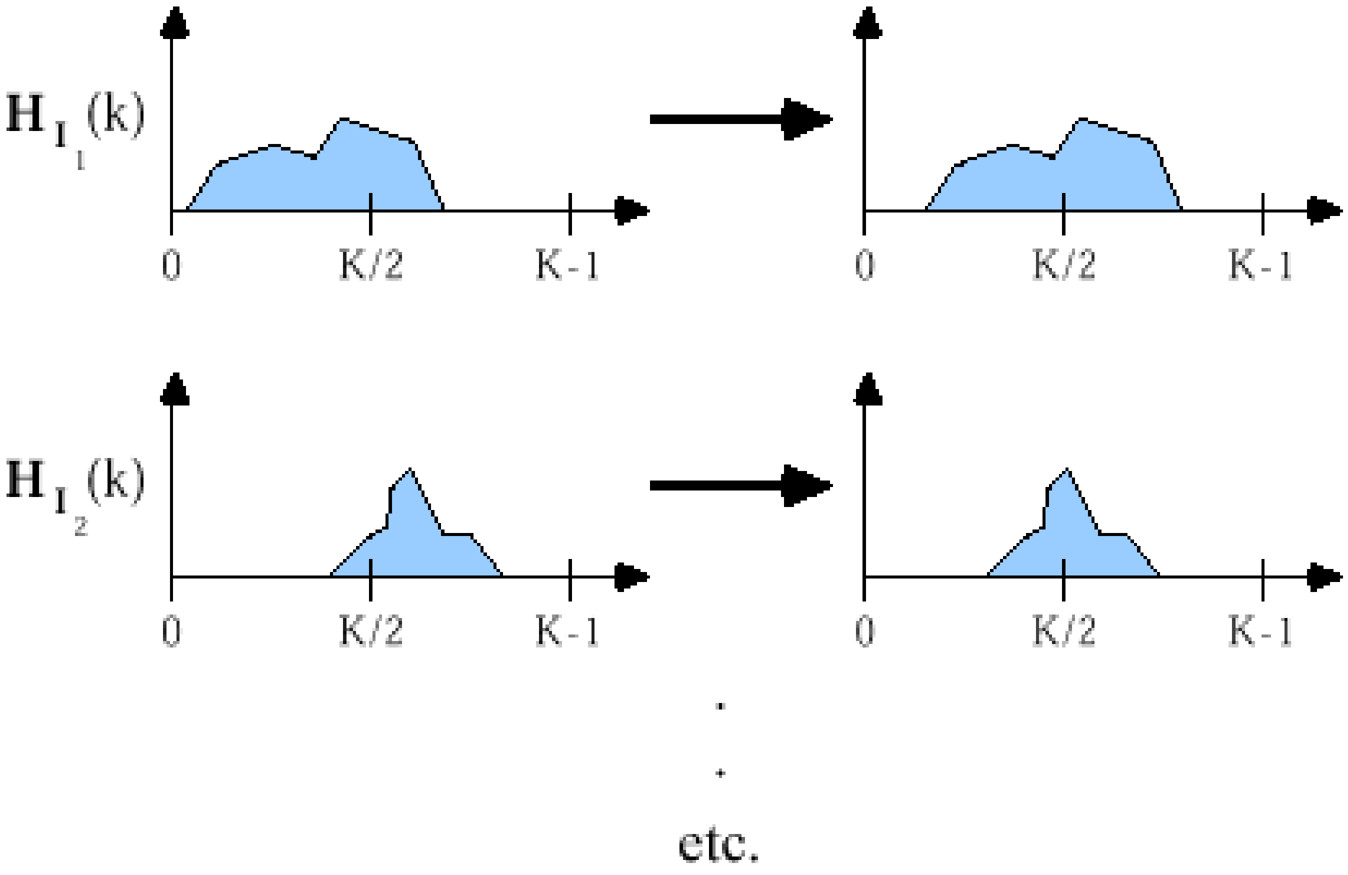


Image Scaling

- Suppose P > 0 (not necessarily an integer)
- **Image scaling** is defined by the function

$$J(i, j) = P \cdot I(i, j) \text{ , for } 0 <= i, j <= N-1$$

- Thus, P multiplies every image pixel value
- In practice:

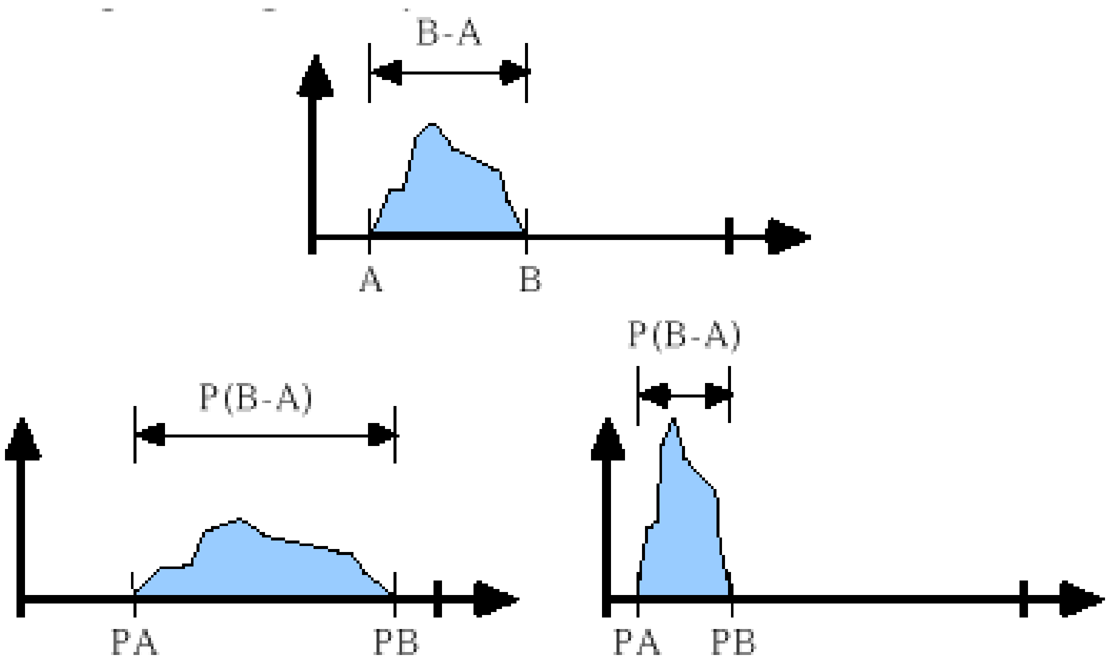
$$J(i, j) = INT[\, P \cdot I(i, j) + 0.5 \,] \text{ , for } 0 <= i, j <= N-1$$

where INT[R] = nearest integer that is <= R

- If P > 1, **J** will have a **broader grey level range** than image **I**

Image Scaling

- If P < 1, **J** will have a **narrower grey-level range** than **I**
- Multiplying by a constant P **stretches** or **compresses** the "width" of the image histogram by a factor P:



Comments

- An image with a compressed gray level range generally has a **reduced visual contrast**
- Such an image may have a **washed-out** appearance
- An image with a wide range of gray levels generally has an **increased visual contrast**
- Such an image may have a more striking, viewable appearance

Linear Point Operations: Offset & Scaling

- Suppose L and P are real numbers (not necessarily integers)
- A **linear point operation** on **I** is defined by the function

$$J(i, j) = P \cdot I(i, j) + L, \text{ for } 0 <= i, j <= N-1$$

- In practice:

$$J(i, j) = \text{INT}[\ P \cdot I(i, j) + L + 0.5\ ]\ , \text{ for } 0 <= i, j <= N-1$$

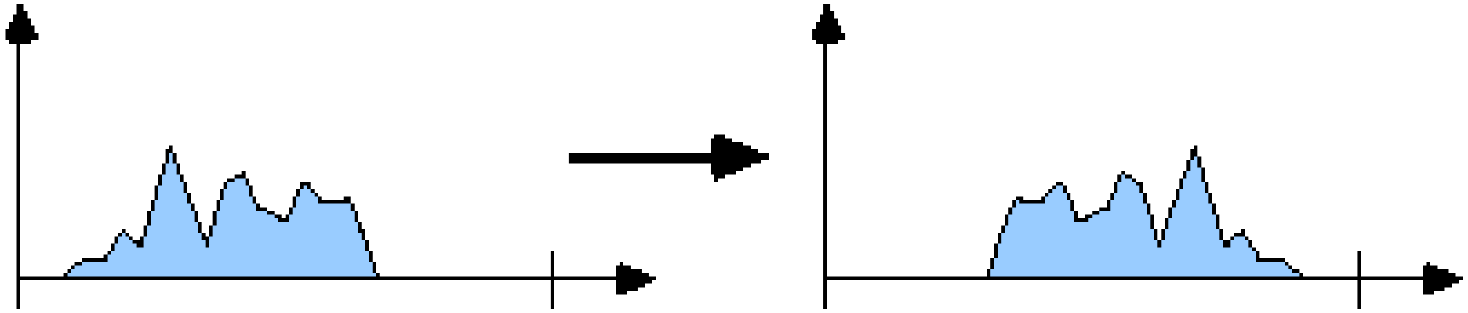
- The image **J** is a version of **I** that has been scaled and given an additive offset

Linear Point Operations: Offset & Scaling

- If P < 0, the histogram is **reversed**, creating a **negative** image
- By far the most common use is P = -1 and L = K-1:

  J(i, j) = (K-1) -I(i, j) , for 0 <= i, j <= N-1

- Hereafter w



Histogram of "**digital negative image**"

Caveat

- Generally, the available gray-scale of the transformed image **J** is the same as that of the original image **I**: {0 ,..., K-1}
- When making the transformation

  $$J(i, j) = P \cdot I(i, j) + L, \text{ for } 0 <= i, j <= N-1$$

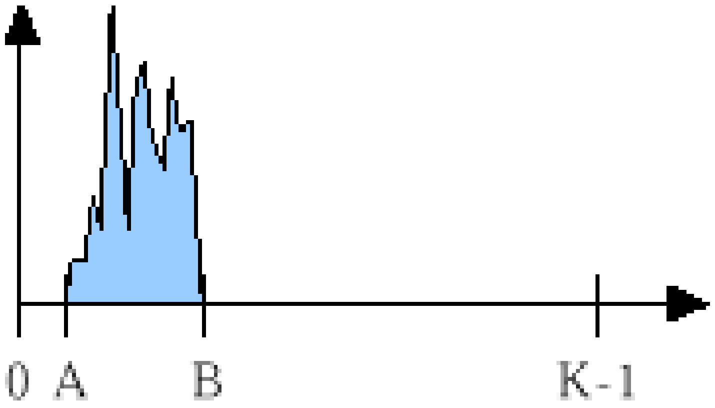
- care must be taken that the maximum and minimum values J_{max} and J_{min} satisfy

  $$J_{max} <= K-1 \text{ and } J_{min} >= 0$$

- **At best**, values outside these ranges will be "clipped"
- At worst, an overflow or sign-error condition may occur
- In that instance, the gray-scale value assigned to an error pixel will be highly unpredictable

Full-Scale Contrast Stretch

- The **most common** linear point operation. Suppose **I** has a compressed histogram:



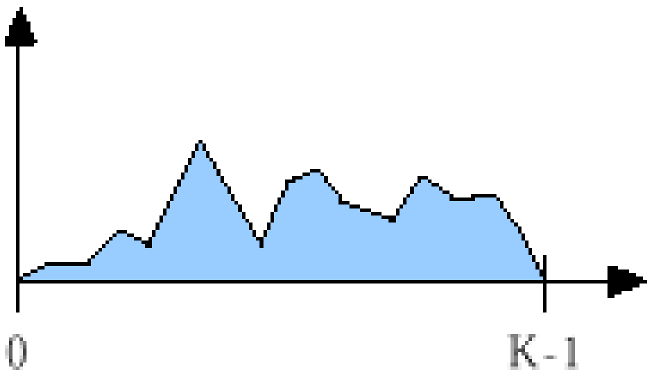
- Let A and B be the min and max gray levels in **I**
- Define

$$J(i, j) = P \cdot I(i, j) + L$$

- such that $P \cdot A + L = 0$ and $P \cdot B + L = (K-1)$

Full-Scale Contrast Stretch

- The result of solving these **2 equations in 2 unknowns** (P, L) is an image **J** with a full-range histogram:



- The solution to the above equations is

$$P = \left(\frac{K-1}{B-A}\right) \quad \text{and} \quad L = -A\left(\frac{K-1}{B-A}\right)$$

or

$$J(i, j) = \left(\frac{K-1}{B-A}\right)\left[I(i, j) - A\right]$$

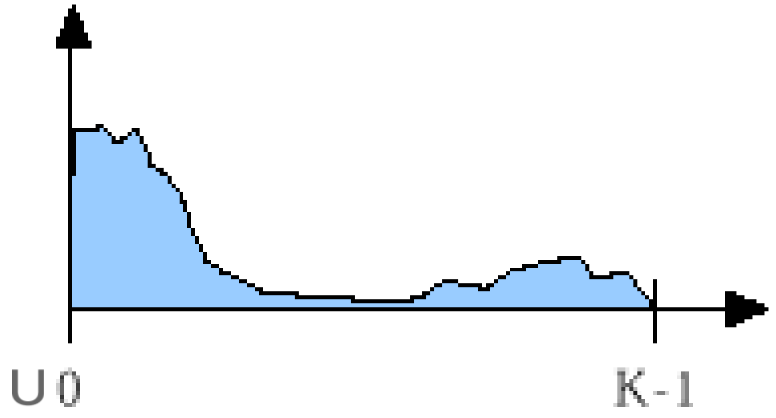
NONLINEAR POINT OPERATIONS

- A **nonlinear point operation** on **I** is a **pointwise function f** mapping **I** to **J**:

  $J(i, j) = f[I(i, j)]$ for $0 <= i, j <= N-1$
- where **f** is a **nonlinear function**.
- This is of course a very broad class of functions
- However, only a few are used much:
  - $J(i, j) = |I(i, j)|$ (absolute value or magnitude)
  - $J(i, j) = [I(i, j)]^2$ (square-law)
  - $J(i, j) = I(i, j)^{1/2}$ (square root)
  - $J(i, j) = \log[1+I(i, j)]$ (logarithm)
  - $J(i, j) = \exp[I(i, j)] = e^{I(i,j)}$ (exponential)

Logarithmic Range Compression

- **Motivation**: An image may contain information-rich, smoothly-changing low intensities - and small very bright regions
- Useful for detecting faint **objects**
- The bright pixels will dominate our visual perception of the image
- A typical histogram:



Logarithmic Range Compression

- **Logarithmic transformation** J(i, j) = log[1+I(i, j)] nonlinearly **compresses** and **equalizes** the gray-scales
- Bright intensities are compressed much more heavily - thus **faint details** emerge
- A full-scale contrast stretch then utilizes the full gray-scale range: