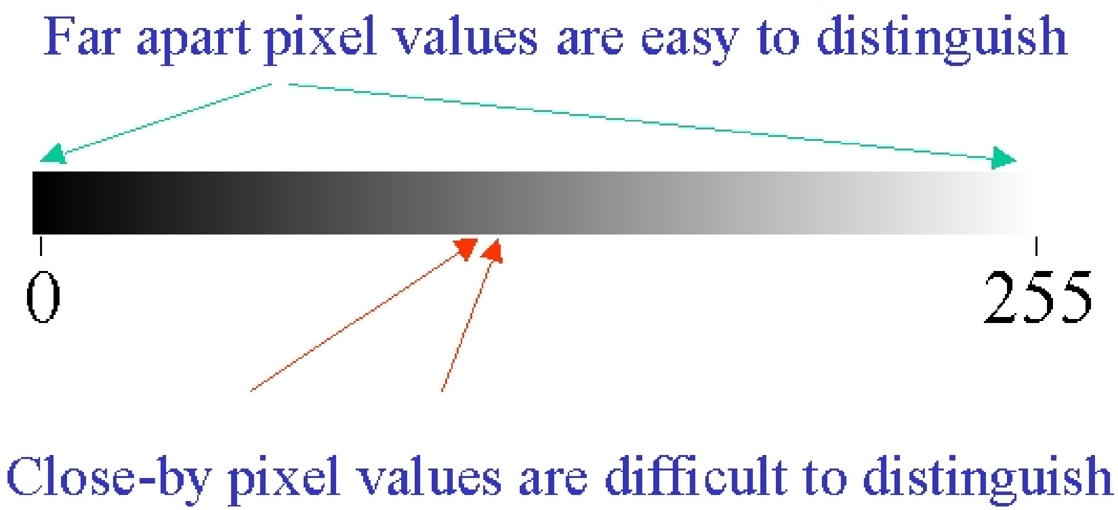
Summary of Lecture 2

- Simple structural processing techniques like transposing, flipping and cropping.
- Simple image statistics like sample mean and sample variance.
- Histograms.
  - $h_A(l)$: number of pixels in image $\mathbf{A}$ that have the value l.
  - Histograms tell us how the values of individual pixels in an image are "distributed".
  - Two different images may have the same histogram.
- Point processing techniques.
  - $B(i,j) = g(A(i,j))$
- In matlab you can also compute $g(l)$ as an array and then utilize $>> B = g(A+1);$.

Dynamic Range, Visibility and Contrast Enhancement



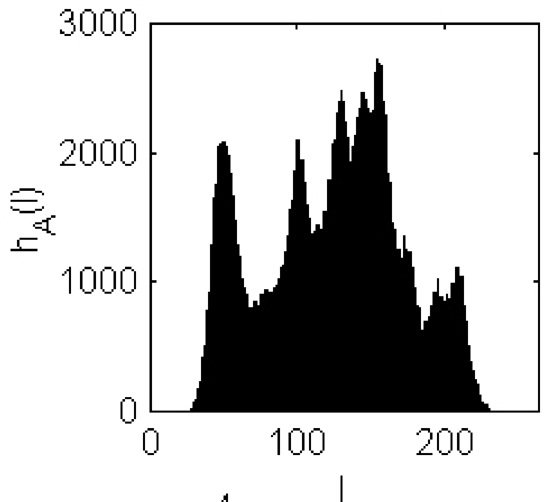
- Contrast enhancing point functions we have discussed earlier expand the dynamic range occupied by certain “interesting” pixel values in the input image.
- These pixel values in the input image may be difficult to distinguish and the goal of contrast enhancement is to make them “more visible” in the output image.
- Don’t forget we have a limited dynamic range $(0-255)$ at our disposal.

Point Functions and Histograms

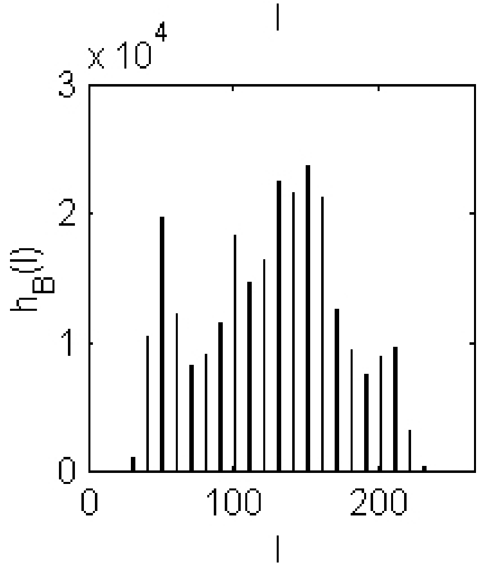
- In general a point operation/function $B(i,j) = g(A(i,j))$ results in a new histogram $h_B(l)$ for the output image that is different from $h_A(l)$.
- The relationship between $h_B(l)$ and $h_A(l)$ may not be straightforward as we have already discussed in Lecture 2.
- You *must* learn how to calculate $h_B(l)$ given $h_A(l)$ and the point function $g(l)$:
  - Exactly: Usually via writing a matlab script that computes $h_B(l)$ from $h_A(l)$ and $g(l)$.
  - Approximately: By sketching $h_B(l)$ given the sketches for $h_A(l)$ and $g(l)$.

“Unexpected” Effect of some Point Functions

A



B(i,j)=10 round(A(i,j)/10)



– **B** has ∼ 10 times as few distinct pixel values.
– Note also the vertical axis scaling in $h_B(l)$.

Stretched/Compressed Pixel Value Ranges

- $B(i, j) = g(A(i, j))$
  Suppose $g(l)$ represents an *overall* point function which includes contrast stretching/compression, emphasis/de-emphasis, rounding, normalizing etc.
- Given an image matrix $\mathbf{A}$, $B(i, j) = g(A(i, j))$ is also an image matrix.
- $g(l)$ may not be "continuous" or connected and it also may not be composed of connected line segments.
- How do we determine which pixel value ranges $g(l)$ stretches/compresses?
  - We can usually assume that in small ranges $g(l)$ may be approximated by piecewise linear, connected line segments. Calculating the implied α_i and testing $|\alpha_i| \gtrless 1$ should help us determine stretched/compressed ranges.

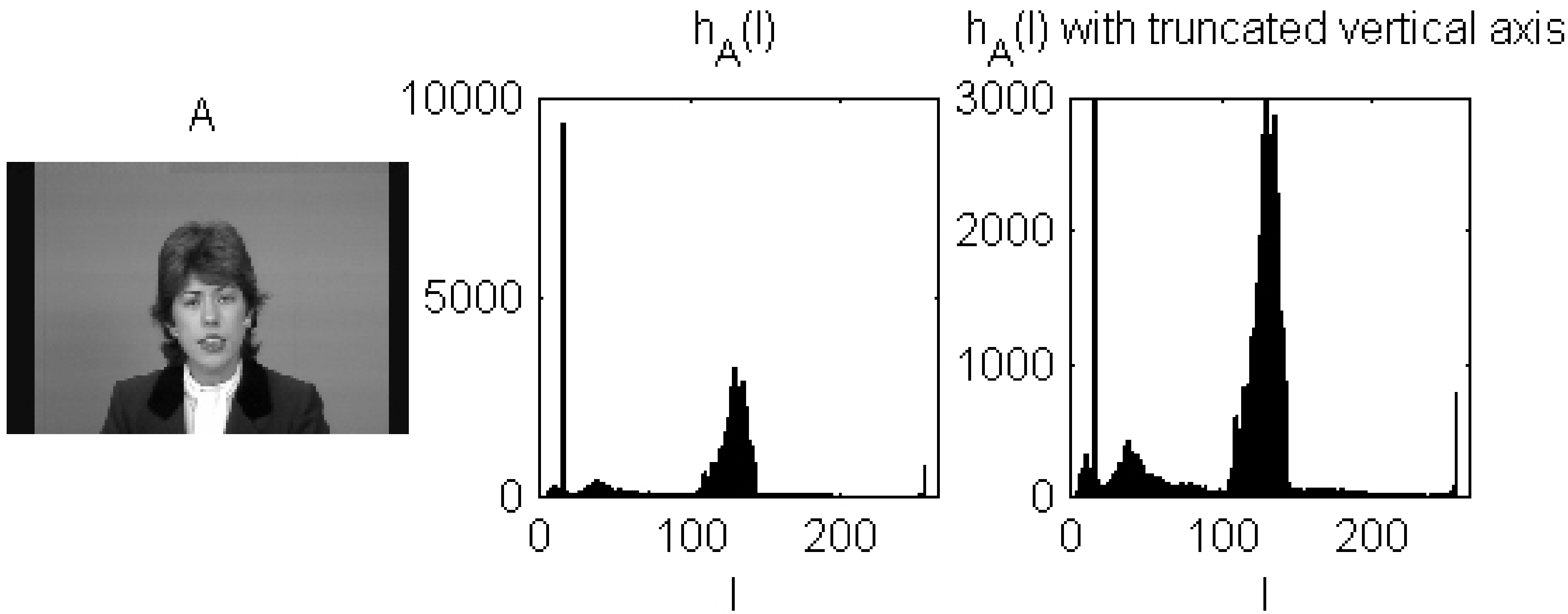
Brief Note on Image Segmentation

- If one views an image as depicting a scene composed of different objects, regions, etc. then segmentation is the decomposition of an image into these objects and regions by associating or "labelling" each pixel with the object that it corresponds to.
- Most humans can easily segment an image.
- Computer automated segmentation is a difficult problem, requiring sophisticated algorithms that work in tandem.
- "High level" segmentation, such as segmenting humans, cars etc., from an image is a very difficult problem. It is still considered unsolved and is actively researched.
- Based on point processing, histogram based image segmentation is a very simple algorithm that is sometimes utilized as an initial guess at the "true" segmentation of an image.

Histogram Based Image Segmentation

- For a given image, decompose the range of pixel values $(0, \ldots, 255)$ into "discrete" intervals $R_t = [a_t, b_t],\ t = 1, \ldots, T$, where T is the total number of segments.
- Each R_t is typically obtained as a range of pixel values that correspond to a hill of $h_A(l)$.
- "Label" the pixels with pixel values within each R_t via a point function.
- Main Assumption: Each object is assumed to be composed of pixels with *similar* pixel values.

Example



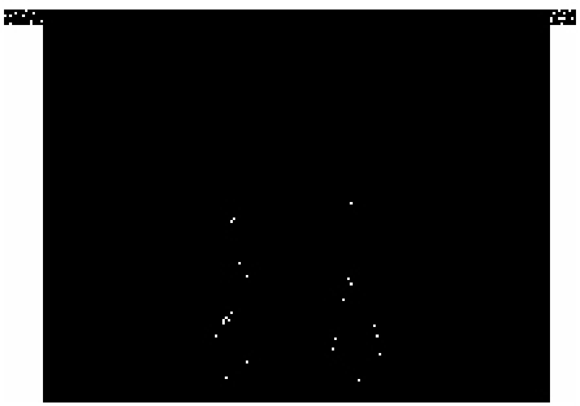
- $R_1 = [0, 14], R_2 = [15, 15], R_3 = [16, 99], R_4 = [100, 149], R_5 = [150, 220], R_6 = [221, 255]$.
- Labeling in matlab: $>> B1 = 255 * ((A >= 0)\&(A <= 14));$, etc.

Example - contd.

A

B1 (R_1=[0,14])

B2 (R_2=[15,15])



B3 (R_3=[16,99])

Example - contd.

A

B4 (R_4=[100,149])

B5 (R_5=[150,220])

B6 (R_6=[221,255])

Example - contd.

A

B (Histogram Segmented)

- Compute the sample mean of each segment ($>> m1 = sum(sum(B1.*A))/sum(sum(B1))$, etc.).
- $C = m1 \times B1 + m2 \times B2 + m3 \times B3 + m4 \times B4 + m5 \times B5 + m6 \times B6.$
  $B(i,j) = g_s^C(C(i,j))$.

Limitations

- Histogram based segmentation operates on each image pixel independently. As mentioned earlier, the main assumption is that objects must be composed of pixels with similar pixel values.
- This independent processing ignores a second important property: Pixels within an object should be *spatially* connected. For example, B3, B4, B5 group spatially disconnected objects/regions into the same segment.
- In practice, one would use histogram based segmentation in tandem with other algorithms that make sure that computed objects/regions are spatially connected.

Histogram Equalization

- For a given image $\mathbf{A}$, we will now design a special point function $g_A^e(l)$ which is called the histogram equalizing point function for $\mathbf{A}$.
- If $B(i,j) = g_A^e(A(i,j))$, then our aim is to make $h_B(l)$ as uniform/flat as possible *irrespective* of $h_A(l)$!
- Histogram equalization will help us:
  - Stretch/Compress an image such that:
    - Pixel values that occur frequently in $\mathbf{A}$ occupy a bigger dynamic range in $\mathbf{B}$, i.e., get stretched and become more visible.
    - Pixel values that occur infrequently in $\mathbf{A}$ occupy a smaller dynamic range in $\mathbf{B}$, i.e., get compressed and become less visible.
  - Compare images by "mapping" their histograms into a standard histogram and sometimes "undo" the effects of some unknown processing.
- The techniques we are going to use to get $g_A^e(l)$ are also applicable in histogram modification/specification.

Continuous Amplitude Random Variables

- Let χ be a continuous amplitude random variable $\chi \in (-\infty, +\infty)$.

  $f_\chi(x)$: the probability density function of χ,
  $F_\chi(x)$: the probability distribution function of χ.

$$f_\chi(x)dx = \text{Probability}(x \leq \chi < x + dx) \quad (1)$$

$$F_\chi(x) = \text{Probability}(\chi \leq x) \quad (2)$$

- Properties:

$$F_\chi(x) = \int_{-\infty}^{x} f_\chi(t)dt \Rightarrow \frac{dF_\chi(x)}{dx} = f_\chi(x) \quad (3)$$

$$f_\chi(x) \geq 0 \Rightarrow F_\chi(x) \geq 0, \;\; F_\chi(x+dx) - F_\chi(x) \geq 0 \quad (4)$$

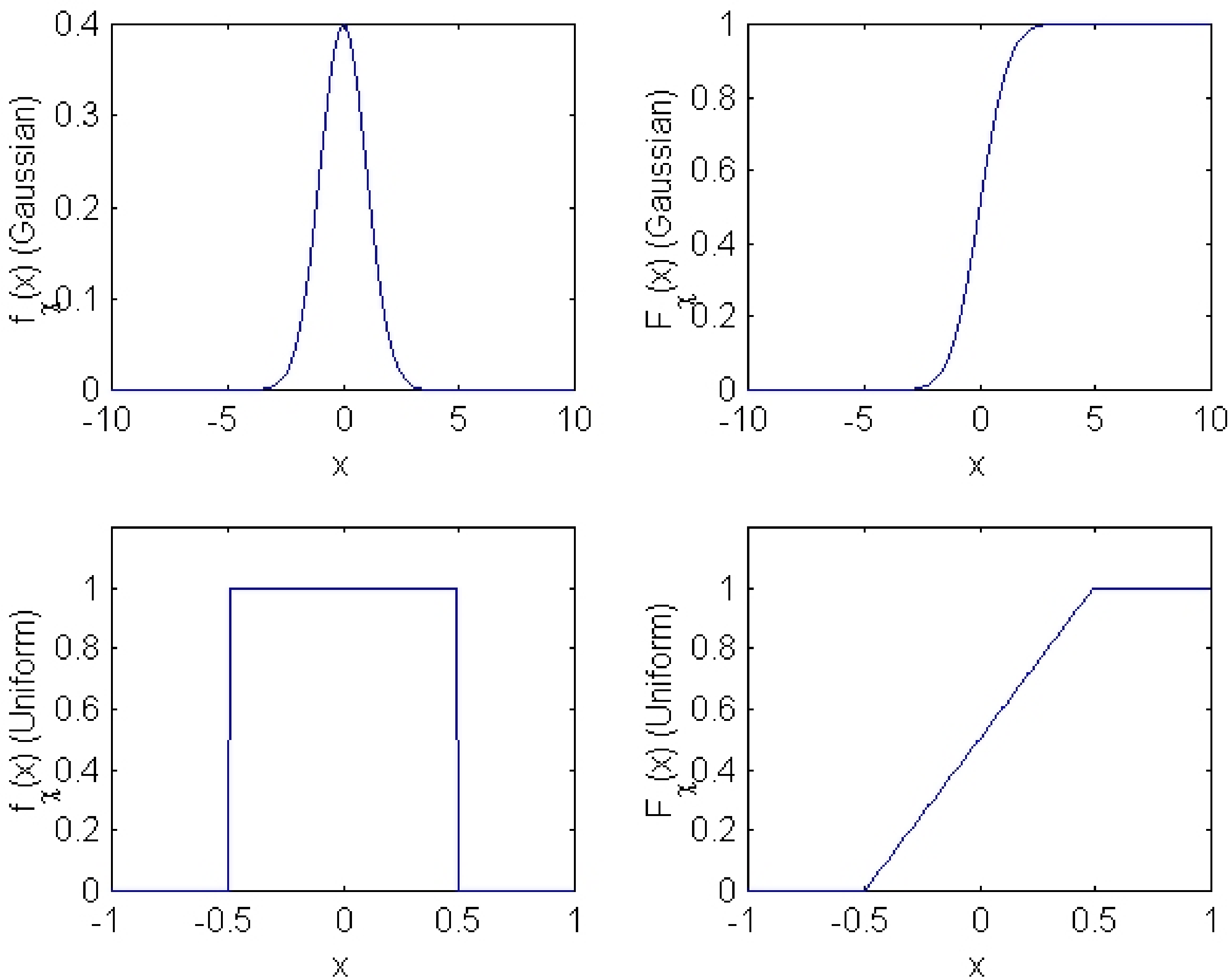
$F_\chi(x)$ is a non-decreasing function.

$$\int_{-\infty}^{+\infty} f_\chi(t)dt = 1 \Rightarrow f_\chi(x)|_{x=+/-\infty} = 0 \quad (5)$$

$$F_\chi(x)|_{x=+\infty} = 1 \quad (6)$$

$$F_\chi(x)|_{x=-\infty} = 0 \quad (7)$$

Example



Gaussian:

$$f_\chi(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Uniform $(a < b)$:

$$f_\chi(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & \text{otherwise} \end{cases}$$

Calculating the Mean and Variance

- Mean (μ):

$$\mu = \int_{-\infty}^{+\infty} x f_\chi(x) dx \tag{8}$$

Analogy: Average price of apples

  - "I bought $f_\chi(x)dx$ many apples at a price of x, ..."
  - "Total price I paid: $P = \int_{-\infty}^{+\infty} x f_\chi(x)dx$.
  - "Total number of apples I purchased: $N = \int_{-\infty}^{+\infty} f_\chi(x)dx = 1$.
  - "My average price for the overall purchase: $\mu = P/N$.

- Variance (σ^2):

$$\sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 f_\chi(x) dx \tag{9}$$

Main Derivation

- We will now obtain a new random variable Y $(f_Y(y), F_Y(y))$ as a *function* of the random variable χ, i.e., $Y = g(\chi)$.
- We wish to make Y a uniform random variable (a random variable having the uniform probability density function) irrespective of the density of χ.
- Our main assumption will be:
  - Assume $F_\chi(x)$ is a continuous and strictly increasing function (compared to the general case of non-decreasing as in Equation 4)
  - Note that such an $F_\chi(x)$ is one-to-one which will allow us to use its inverse $F_\chi^{-1}(x)$.

Main Derivation - contd.

- Let $Y = F_\chi(\chi)$, i.e., $g(\chi) = F_\chi(\chi)$. Note that $Y \in [0,1]$ and $f_Y(y) = 0$ if $y \notin [0,1]$.
- Let us derive $F_Y(y)$ for $y \in [0,1]$:

$$\begin{aligned} F_Y(y) &= \text{Probability}(Y \leq y) \\ &= \text{Probability}(F_\chi(\chi) \leq y) \\ &= \text{Probability}(\chi \leq F_\chi^{-1}(y)) \\ &= F_\chi(F_\chi^{-1}(y)) \\ &= y \end{aligned}$$

  where the next to last step follows from Equation 2.
- Using Equation 3 and $f_Y(y) = 0$ if $y \notin [0,1]$, we have

$$f_Y(y) = \begin{cases} 0 & y < 0, \; y > 1 \\ 1 & y \in [0,1] \end{cases} \tag{10}$$

  i.e., Y is a uniform random variable with $a = 0$ and $b = 1$.

Discrete Amplitude Random Variables

- Let Θ be a discrete amplitude random variable.
  $\Theta = x_i$ for some i, $\ldots, -1, 0, 1, \ldots$.
  x_i are a sequence of possible values for Θ.

$p_\Theta(x_i)$: the probability mass function of Θ,
$F_\Theta(x_i)$: the probability distribution function of Θ.

$$p_\Theta(x_i) = \text{Probability}(\Theta = x_i) \tag{11}$$

$$F_\Theta(x_i) = \text{Probability}(\Theta \leq x_i) \tag{12}$$

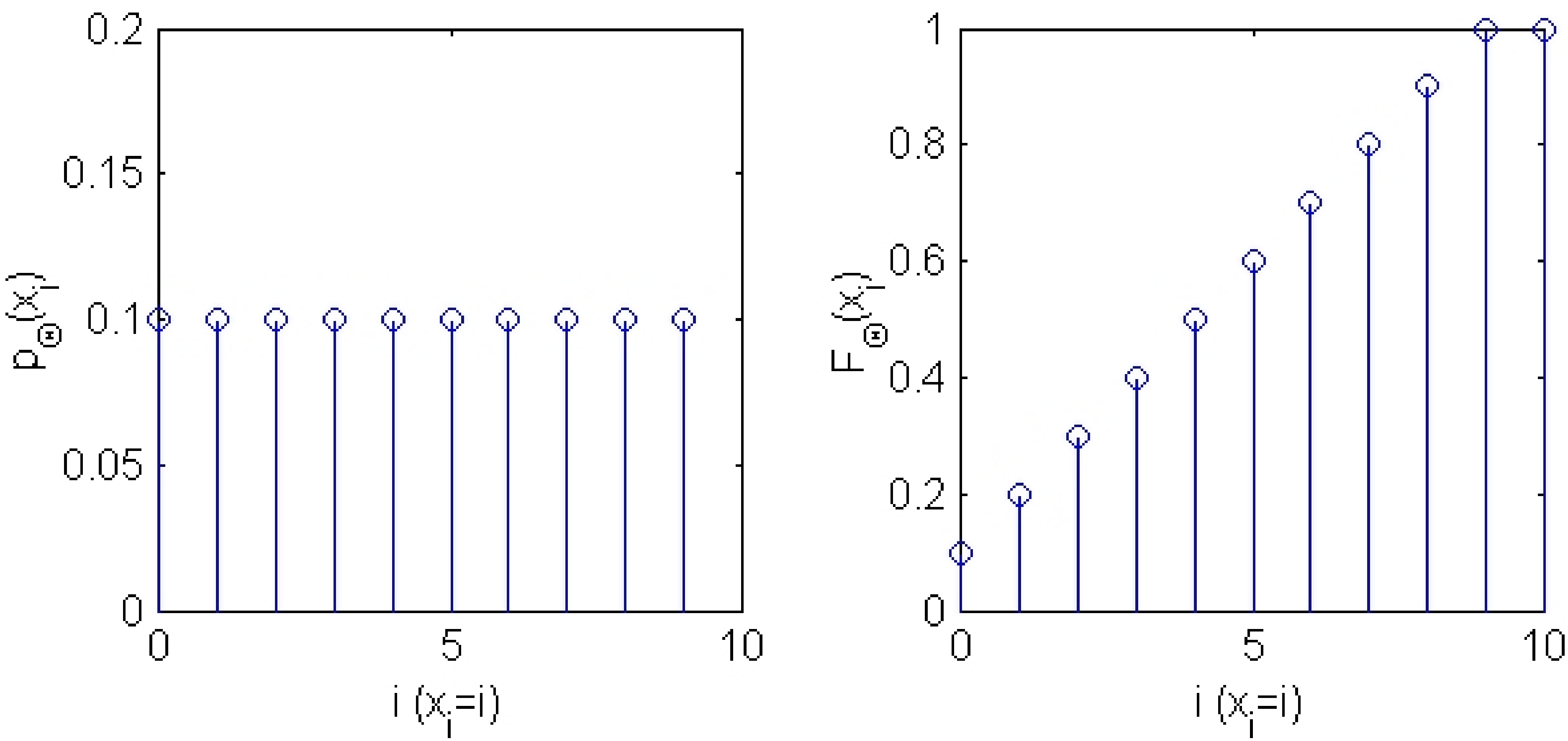
- Properties:

$$F_\Theta(x_i) = \sum_{j=-\infty}^{j=i} p_\Theta(x_j) \tag{13}$$

$$p_\Theta(x_i) = F_\Theta(x_i) - F_\Theta(x_{i-1}) \geq 0 \tag{14}$$

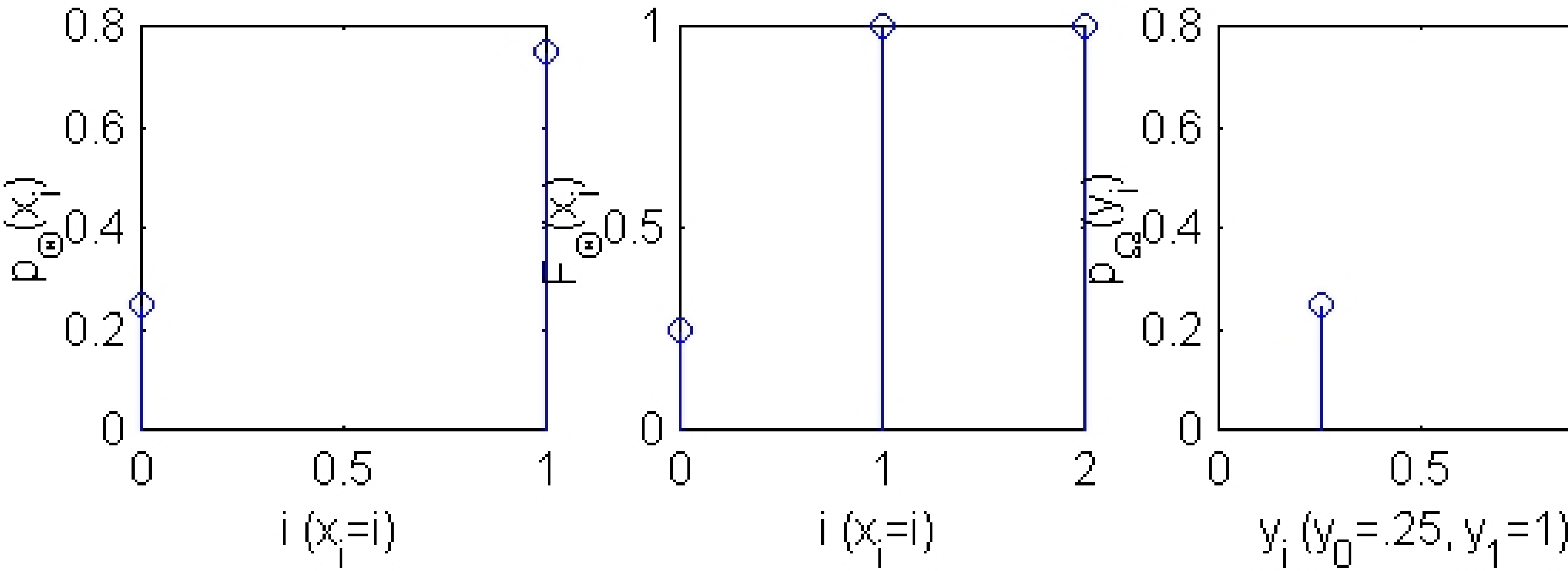
$$\sum_{j=-\infty}^{j=+\infty} p_\Theta(x_j) = 1 \tag{15}$$

Example



The probability mass and distribution functions for a uniform, discrete amplitude random variable.

Derivation for Discrete Amplitude R.V.s



- Let $\Omega = F_\Theta(\Theta)$. $\Omega = y_i = F_\Theta(x_i)$ for some i, $\ldots, -1, 0, 1, \ldots$.
- Our earlier derivation for continuous amplitude random variables does not "work" for discrete amplitude random variables.
- In general Ω is not a uniform random variable.

Histogram as a Probability Mass Function

- For a given image $\mathbf{A}$, consider the image pixels as the realizations of a discrete amplitude random variable "A".
  - For example suppose we toss a coin (Heads=255 and Tails=0) $N \times M$ times and record the results as an N by M image matrix.
- Define the sample probability mass function $p_A(l)$ as the probability of a randomly chosen pixel having the value l.

$$p_A(l) = \frac{h_A(l)}{NM} \tag{16}$$

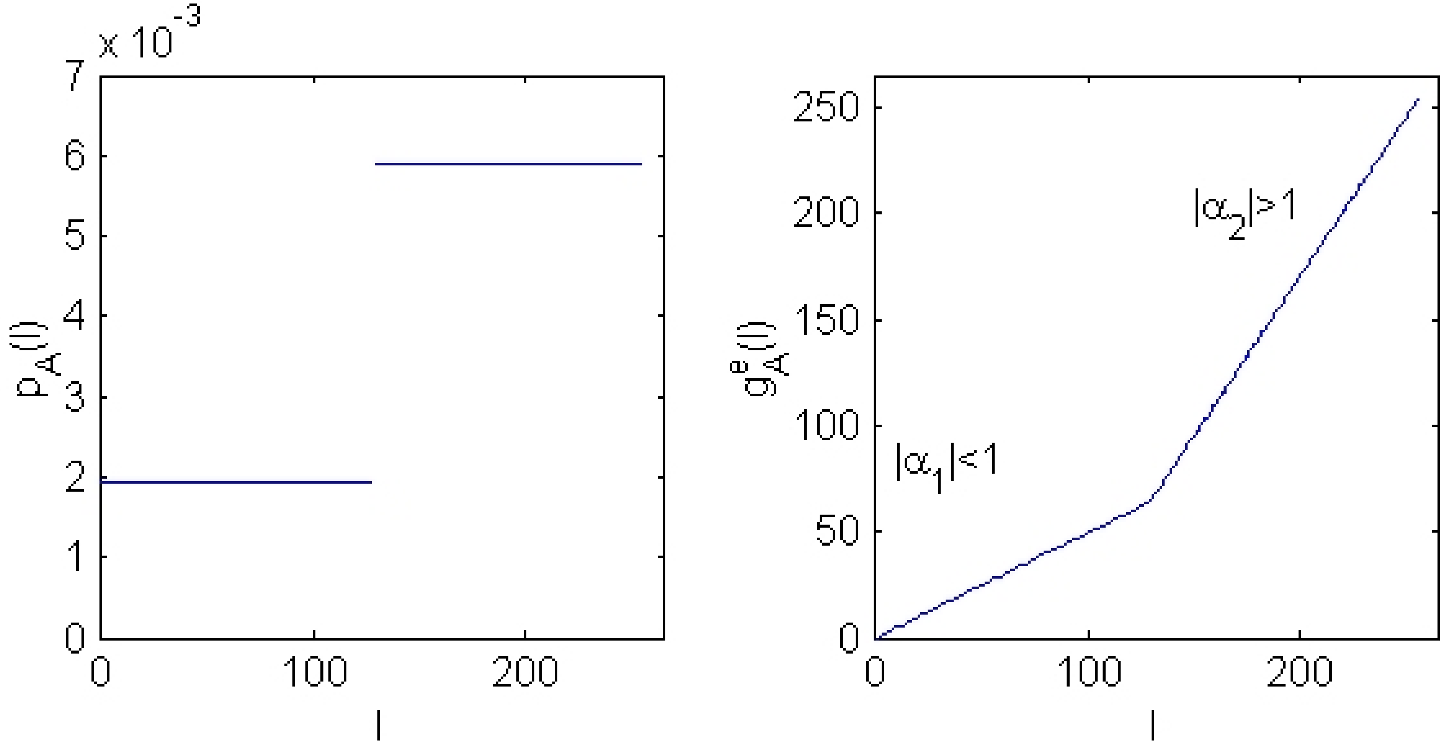
- Note that the sample mean and variance we talked about in Lecture 2 can be calculated as:

$$\begin{aligned} m_A &= \sum_{l=0}^{255} l p_A(l) \\ \sigma_A^2 &= \sum_{l=0}^{255} (l - m_A)^2 p_A(l) \end{aligned}$$

Histogram Equalizing Point Function

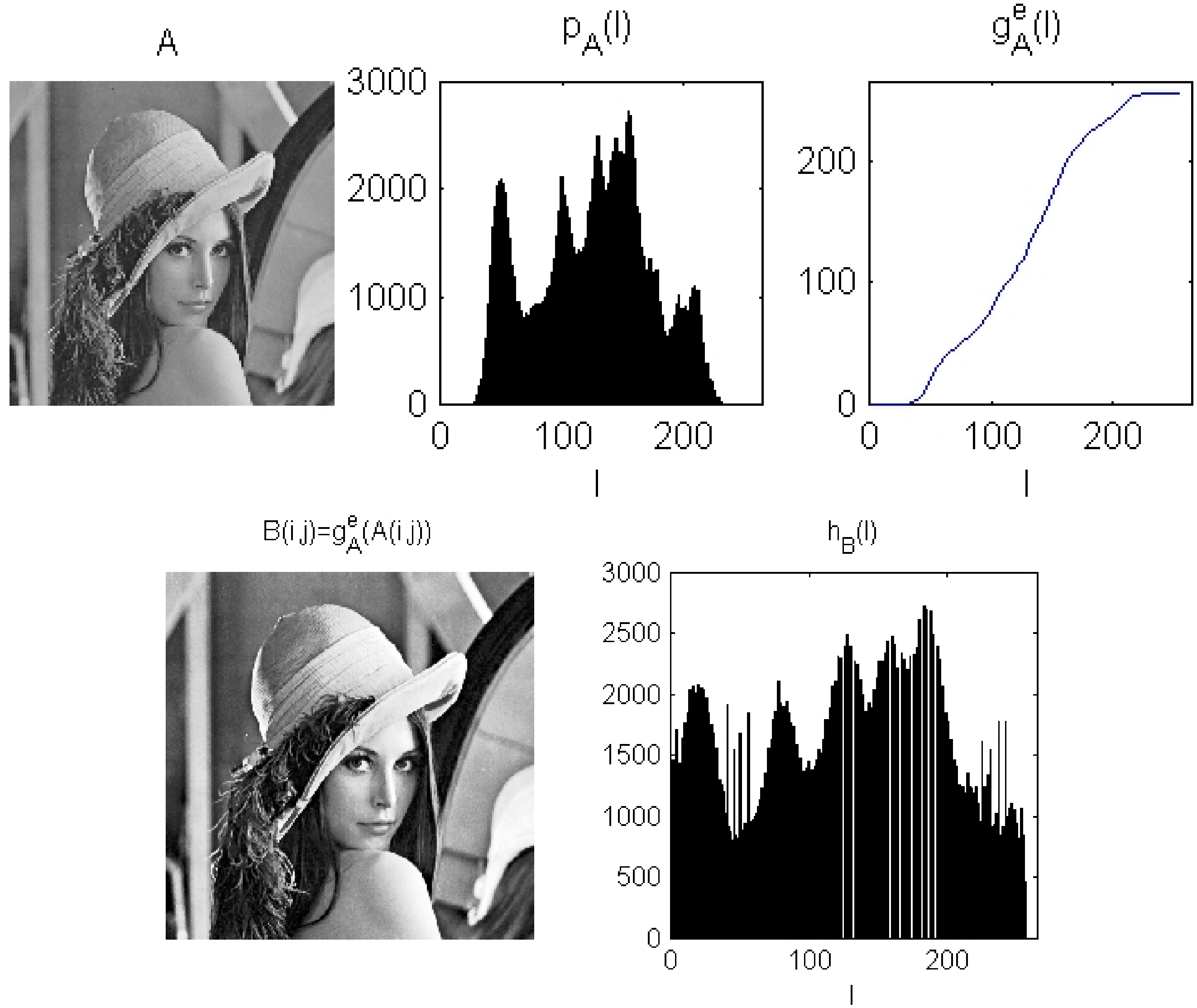
- Let $g_1(l) = \Sigma_{k=0}^{l} p_A(k)$. Note that $g_1(l) \in [0, 1]$.
- $g_A^e(l) = \mathrm{round}(255 g_1(l))$ is the histogram equalizing point function for the image $\mathbf{A}$.
- Image $\mathbf{A}$ $\Rightarrow$ "equalize image" $\Rightarrow B(i,j) = g_A^e(A(i,j))$.
- As we have seen, in general $p_B(l)$ will not be a uniform probability mass function but hopefully it will be close.
- In matlab $>>$ help filter to construct $g_A^e(A(i,j))$ fast.
- Assuming you gAe is an array that contains the computed $g_A^e(l)$, you can use $>> B = gAe(A+1);$ to obtain the equalized image.

Stretching and Compression



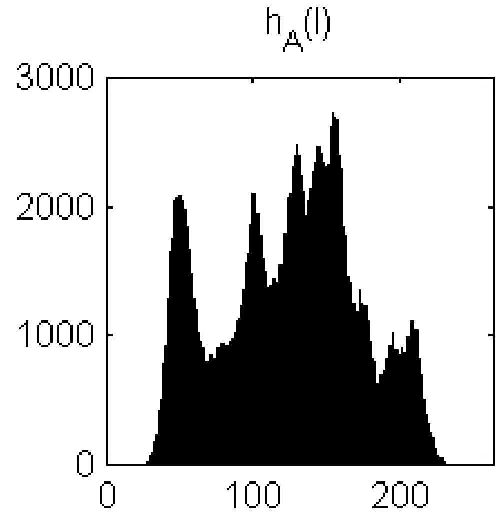
- $g_A^e(l)$ stretches the range of pixel values that occur frequently in $\mathbf{A}$.
- $g_A^e(l)$ compresses the range of pixel values that occur infrequently in $\mathbf{A}$.

Example

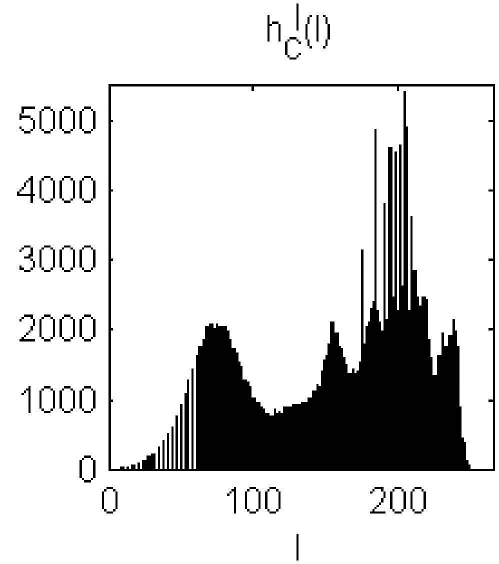

A
p_A(l)
3000
2000
1000
0
0
100
200
l
g^e_A(l)
200
100
0
0
100
200
l
B(i,j)=g^e_A(A(i,j))
h_B(l)
3000
2500
2000
1500
1000
500
0
0
100
200
l

Comparison / “Undoing”

A

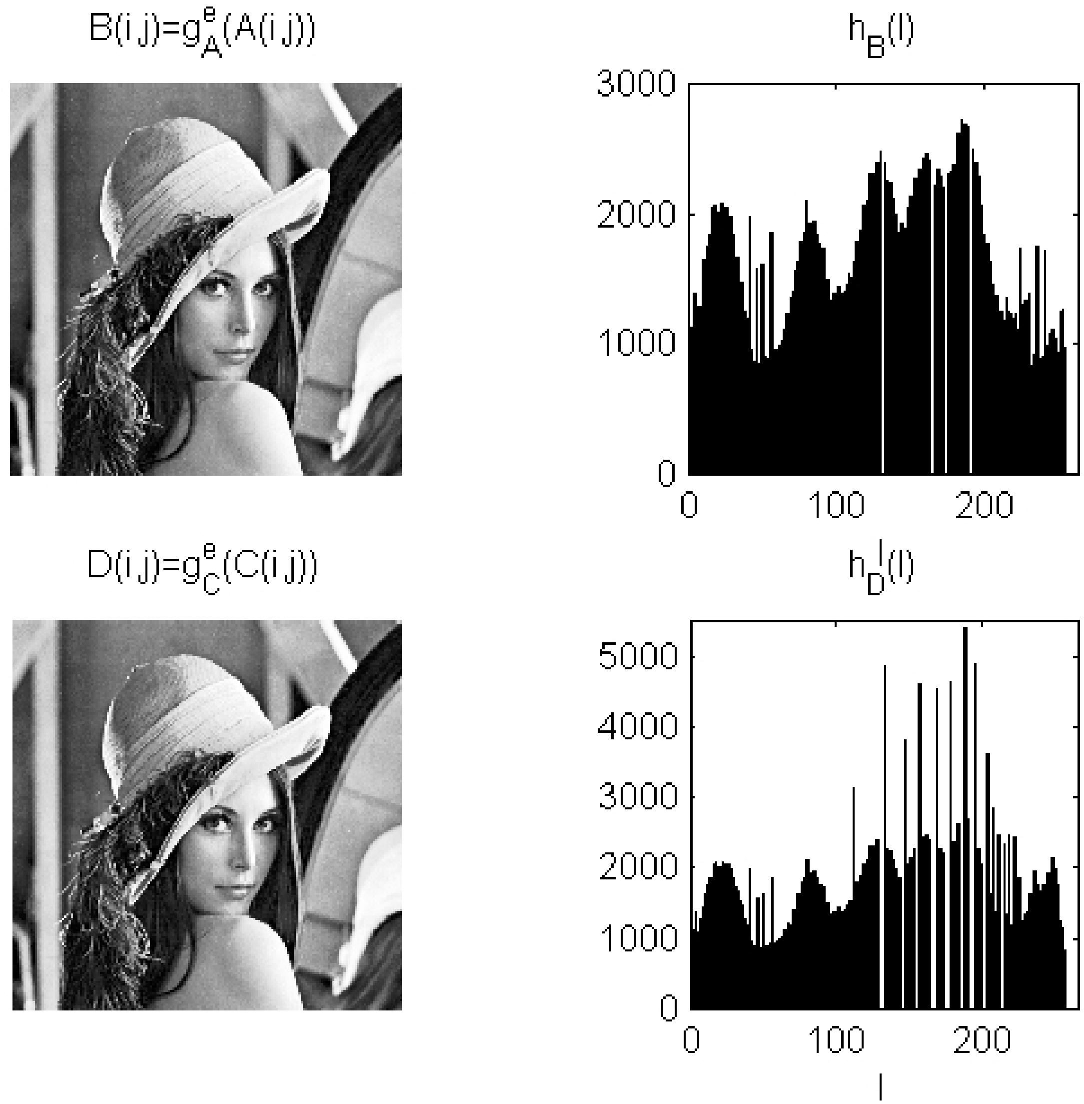


$C(i,j)=g_s^T(T(i,j))$ $(T(i,j)=\log_{10}(A(i,j)+1))$



Instead of comparing **A** and **C**, compare their equalized versions.

Comparison/ "Undoing" - contd.



Summary

- In this lecture we learnt about simple histogram based image segmentation and its limitations.
- We reviewed continuous and discrete amplitude random variables and used their properties to derive the histogram equalizing point function.
- We also defined the relationship between image histograms and sample probability mass functions of images.
- Finally we looked at some equalization examples and learnt what to expect from a histogram equalizing point function when applied on an image.
- Please read Chapter 7, pages 241-244 in the textbook.

Homework III

1. The histogram of an image $\mathbf{A}$ is $h_A(l) = l, \ (l = 0, \ldots, 255)$. A point function

$$g(l) = \begin{cases} l & 0 \leq l < 128 \\ 255 - l & 128 \leq l \leq 255 \end{cases}$$

   Let $B(i,j) = A(i,j)$. Calculate $h_B(l)$ without a computer. Show all your work.

2. Implement histogram based segmentation on your image. Identify the peaks of your histogram with the "objects" that they correspond to. Show your image, its histogram, the ranges, etc. Show the identified objects. Finally construct the histogram segmented image.

3. Derive the mean and variance for continuous amplitude Gaussian and uniform densities.

4. Equalize your image. Show before and after images and histograms. Is the histogram of the equalized image uniform? Which regions got stretched/compressed? (Be as accurate as possible)

5. Implement the Comparison/"Undoing" example on your image.

References

[1] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.