Lecture 11

Discrete Intensity Values

PMF :

$$p_r(r_k) = \frac{n_k}{MN} \qquad k = 0, 1, 2, \ldots, L-1$$

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L-1$$

Continuous Intensity Values:

$$s = T(r) = (L-1)\int_0^r p_r(\omega)d\omega$$

PDF

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

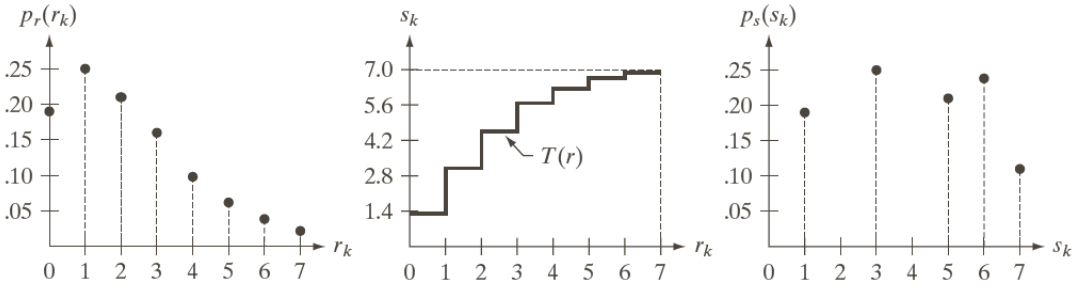
TABLE 3.1
Intensity distribution and histogram values for a 3-bit, 64 × 64 digital image.

M = N = 64

>> 790/(64*64)

ans =

0.1929



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

http://www.ece.uah.edu/~dwpan/course/ee604/code/ch3/hist_eq.m

```
function J = hist_eq(I)
% I: Input Image
% J: Output Image

L = 256;
K = (L-1)/ numel(I);
      [counter, bin] = imhist(I);
transf = bin;
sum = 0;
for i = 1: length(bin)
   sum = sum + counter(i);
   transf(i) = uint8(round(K * sum));
end
J = uint8(transf(I+1));          % accounting for pixel value starting from 0
end
```

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L-1$$

MN = numel(I)

Rounding

$s_0 = 1.33 \rightarrow 1$ $\quad s_4 = 6.23 \rightarrow 6$

$s_1 = 3.08 \rightarrow 3$ $\quad s_5 = 6.65 \rightarrow 7$

$s_2 = 4.55 \rightarrow 5$ $\quad s_6 = 6.86 \rightarrow 7$

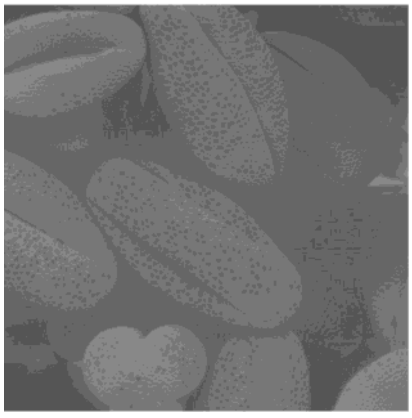
$s_3 = 5.67 \rightarrow 6$ $\quad s_7 = 7.00 \rightarrow 7$

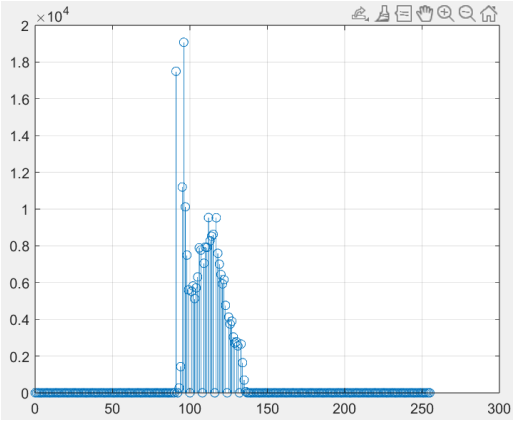
```
>> round(1.33)
ans =
    1

>> round(4.55)
ans =
    5
```

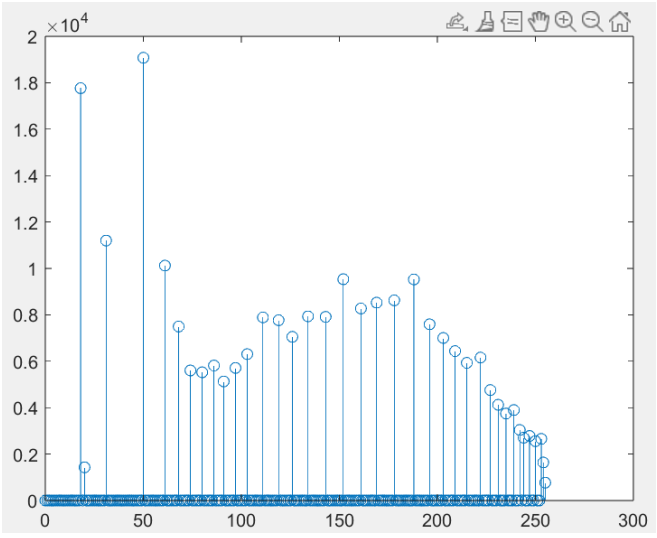
```
>> I = imread('Fig0320(2)
(2nd_from_top).tif');
>> imshow(I)

>> [counter, bin] = imhist(I);
>> figure; stem(bin, counter)
>> grid
```



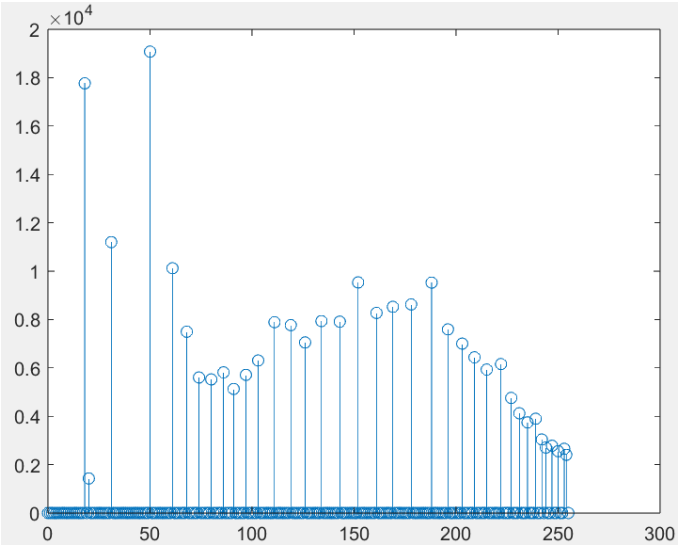


```
>> J = hist_eq(I);
>> figure; imshow(J)
>> [counter, bin] = imhist(J);
>> figure; stem(bin, counter)
```



Continue using histogram equalization on image J:

```
>> K = hist_eq(J);
>> figure; imshow(K)
>> [counter, bin] = imhist(K);
>> figure; stem(bin, counter)
```



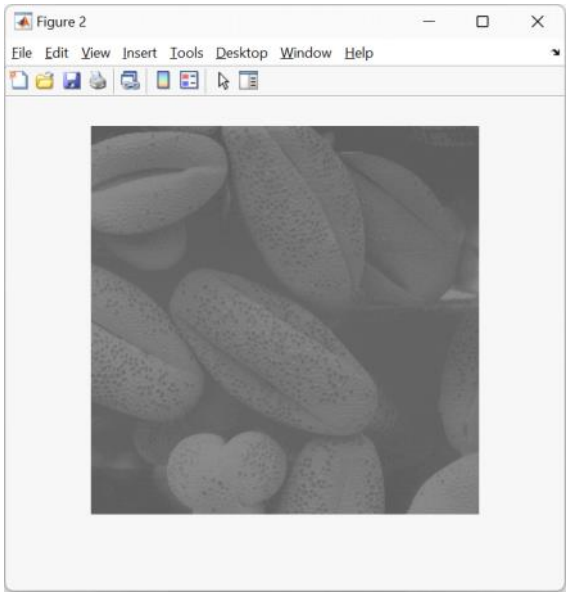
As a comparison, *imadjust* function in Matlab

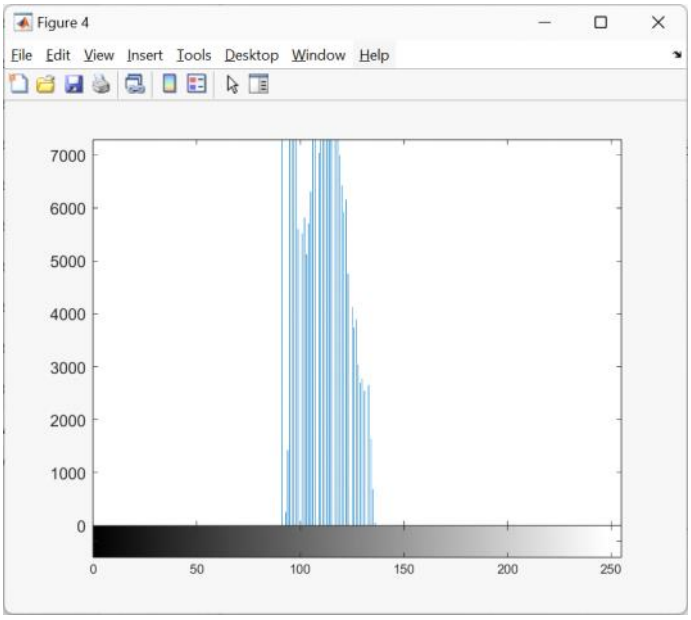
```
>> imtool(I)
>> doc imadjust
```

imadjust

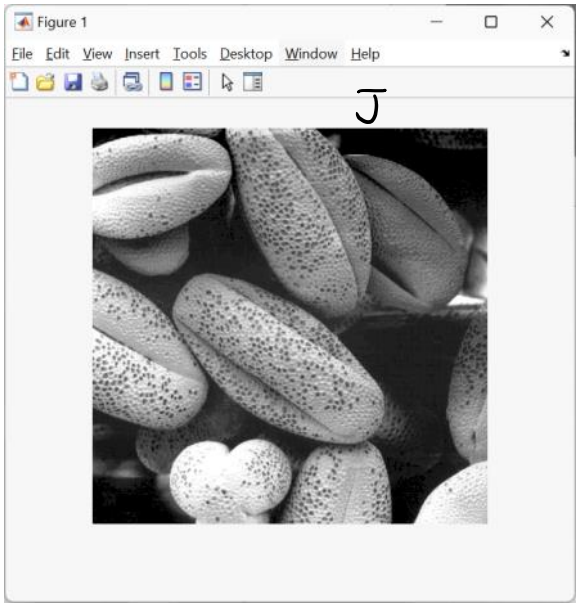
Adjust image intensity values or colormap

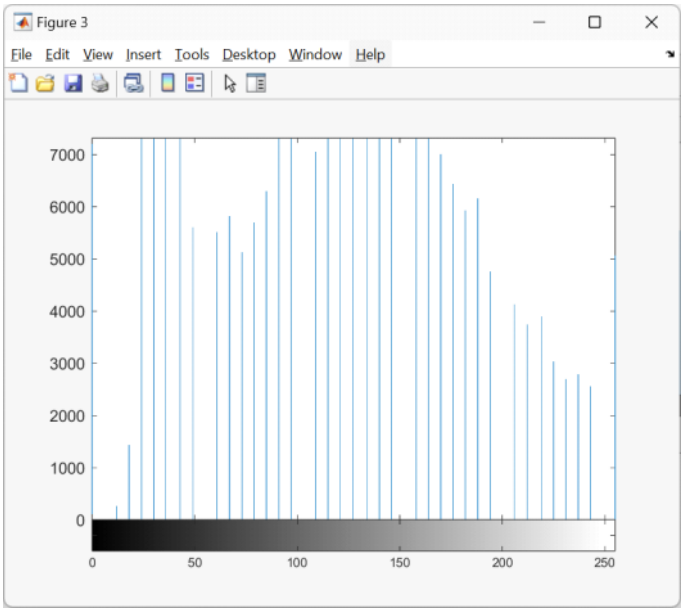
I






```
>> J = imadjust(I);
>> figure; imshow(J)
```





```
>> [counter, bin] = imhist(J);
>> figure; stem(bin, counter)
```