Summary of Lecture 1

- "Let there be light" $\rightarrow$ incident light $\rightarrow$ reflectivity $\rightarrow$ reflected light $\rightarrow$ projection $\rightarrow$ sensitivity $\rightarrow$ $f_{\mathrm{C}}(x', y')$
- Sampling: $f_{\mathrm{C}}(x', y') \rightarrow f_{\mathrm{C}}(i, j)$.
- Quantization: $f_{\mathrm{C}}(i, j) \rightarrow \hat{f}_{\mathrm{C}}(i, j) \in \{0, \ldots, 255\}$.
- As a matrix: $\hat{f}_{\mathrm{C}}(i, j) \rightarrow \hat{f}_{\mathrm{gray}}(i, j) \rightarrow A(i, j)$.

Images as Matrices

- An image matrix $(N \times M)$:

$$\mathbf{A} = \underbrace{\begin{bmatrix} A(0,0) & A(0,1) & A(0,2) & \ldots & A(0,M-1) \\ A(1,0) & A(1,1) & A(1,2) & \ldots & A(1,M-1) \\ \vdots & & & & \\ A(N-1,0) & A(N-1,1) & A(N-1,2) & \ldots & A(N-1,M-1) \end{bmatrix}}_{}\left.\vphantom{\begin{bmatrix}1\\1\\1\\1\end{bmatrix}}\right\} N \text{ rows}$$

- $A(i,j) \in \{0,1,\ldots,255\}$.
- $A(i,j)$:
  - "Matrix case:" The matrix element (i,j) with value $A(i,j)$.
  - "Image case:" The pixel (i,j) with value $A(i,j)$.
  - Will use both terminologies.

Simple Processing - Transpose

- The transpose image $\mathbf{B}$ $(M \times N)$ of $\mathbf{A}$ $(N \times M)$ can be obtained as $B(j,i) = A(i,j)$ $(i = 0, \ldots, N-1,\ \ j = 0, \ldots, M-1)$.

$\mathbf{A}$

$\mathbf{B}$

```
>> for i = 1 : 512
     for j = 1 : 512
       B(j,i) = A(i,j);
     end
   end
```

OR

```
>> B = A';
```

Simple Processing - Flip Vertical

- The vertical flipped image $\mathbf{B}$ $(N \times M)$ of $\mathbf{A}$ $(N \times M)$ can be obtained as $B(i, M-1-j) = A(i,j)$ $(i = 0, \ldots, N-1,\ j = 0, \ldots, M-1)$.

A

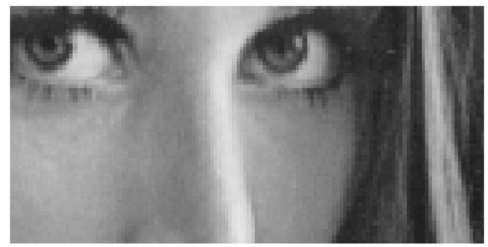
B

$$
\begin{array}{ll}
>> & \text{clear } B; \\
>> & \text{for } i = 1:512 \\
 & \quad \text{for } j = 1:512 \\
 & \qquad B(i, 512+1-j) = A(i,j); \\
 & \qquad \text{end} \\
 & \quad \text{end}
\end{array}
$$

Simple Processing - Cropping

- The cropped image $\mathbf{B}$ $(N_1 \times N_2)$ of $\mathbf{A}$ $(N \times M)$, starting from (n_1, n_2), can be obtained as $B(k,l) = A(n_1 + k, n_2 + l)$ $(k = 0, \ldots, N_1 - 1,\ \ l = 0, \ldots, N_2 - 1)$.

A



B

$$
\begin{array}{ll}
>> & \text{clear } B; \\
>> & \text{for } k = 0 : 64 - 1 \\
 & \quad \text{for } l = 0 : 128 - 1 \\
 & \quad\quad B(k+1, l+1) = A(255 + k + 1, 255 + l + 1); \quad \text{\% n1=n2=255 N1=64,N2=128} \\
 & \quad\quad \text{end} \\
 & \quad \text{end}
\end{array}
$$

Cropping as a Matlab Function

function $[B]$ =mycrop$(A, n1, n2, N1, N2)$
% mycrop.m
% $[B]$ =mycrop$(A, n1, n2, N1, N2)$
% crops the image A from location $n1, n2$
% with size $N1, N2$

for $k = 0 : N1 - 1$
    for $l = 0 : N2 - 1$
      $B(k+1, l+1) = A(n1+k+1, n2+l+1);$
      end
    end

---

function $[B]$ =mycrop$(A, n1, n2, N1, N2)$
% mycrop.m
% $[B]$ =mycrop$(A, n1, n2, N1, N2)$
% crops the image A from location $n1, n2$
% with size $N1, N2$

$B(1 : N1, 1 : N2) = A(n1+1 : n1+N1, n2+1 : n2+N2);$

>> help mycrop
>> B =mycrop$(A, 255, 255, 64, 128);$

Simple Image Statistics - Sample Mean and Sample Variance

- The sample mean (m_A) of an image $\mathbf{A}$ ($N \times M$):

$$m_A = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A(i,j)}{NM} \tag{1}$$

- The sample variance (σ_A^2) of $\mathbf{A}$:

$$\sigma_A^2 = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (A(i,j) - m_A)^2}{NM} \tag{2}$$

- The sample standard deviation, $\sigma_A = \sqrt{\sigma_A^2}$.

Simple Image Statistics - Histogram

Let S be a set and define $\#S$ to be the cardinality of this set, i.e., $\#S$ is the number of elements of S.

- The histogram $h_A(l)$ $(l = 0, \ldots, 255)$ of the image $\mathbf{A}$ is defined as:

$$h_A(l) \;=\; \#\{(i,j) \mid A(i,j) = l,\; i = 0, \ldots, N-1,\; j = 0, \ldots, M-1\} \tag{3}$$

- Note that:

$$\sum_{l=0}^{255} h_A(l) = \text{Number of pixels in } \mathbf{A} \tag{4}$$

Calculating the Histogram

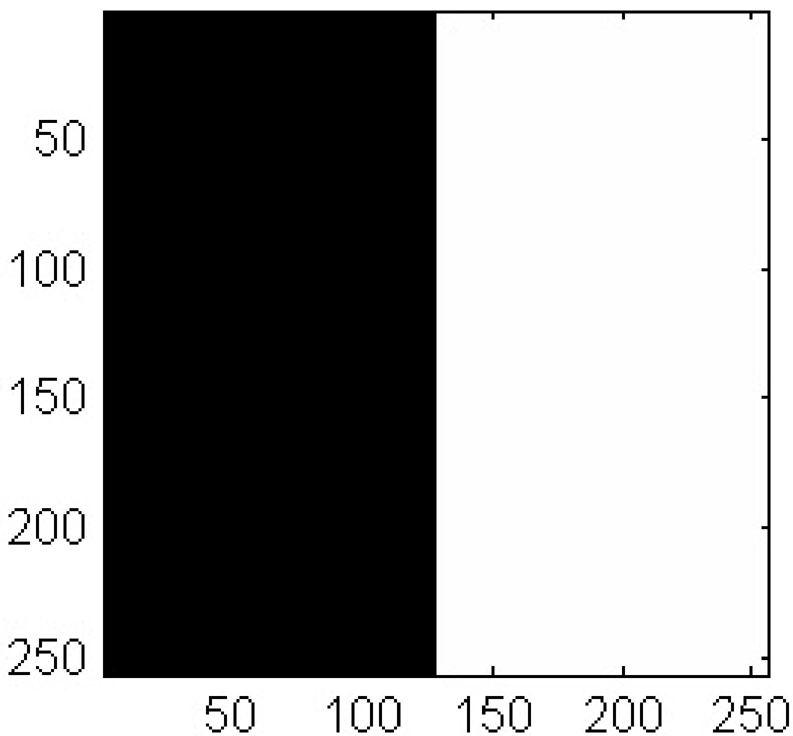
```
>> h=zeros(256,1);
>> for l = 0 : 255
     for i = 1 : N
         for j = 1 : M
            if (A(i, j) == l)
               h(l + 1) = h(l + 1) + 1;
               end
            end
         end
     end
>> bar(0:255,h);
```

OR

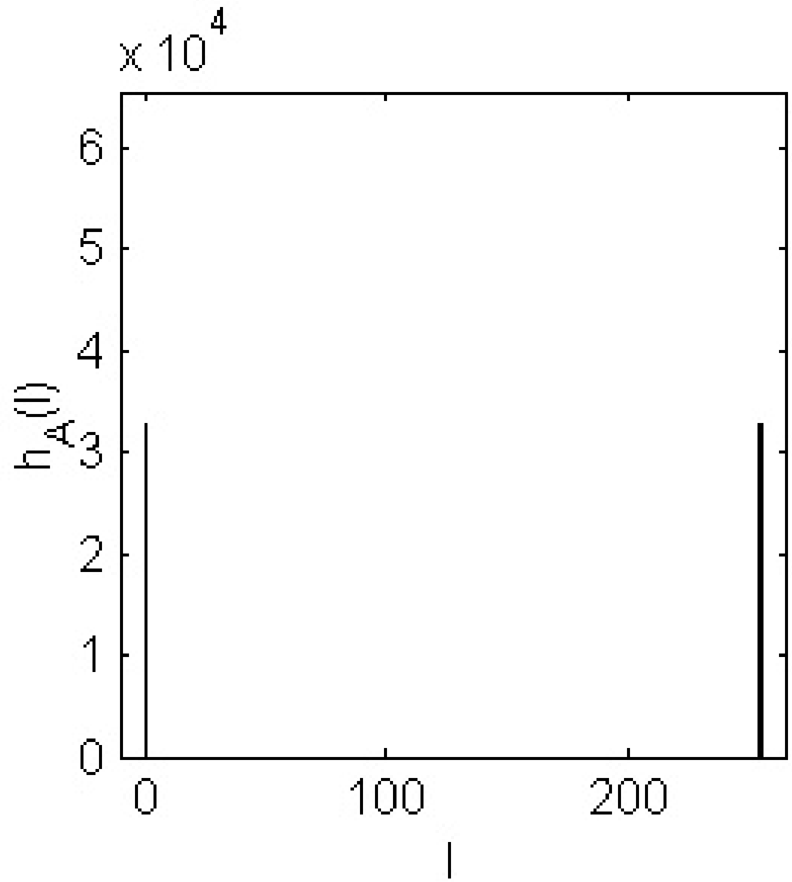
```
>> h=zeros(256,1);
>> for l = 0 : 255
      h(l + 1)=sum(sum(A == l));
      end
>> bar(0:255,h);
```

?

Example I

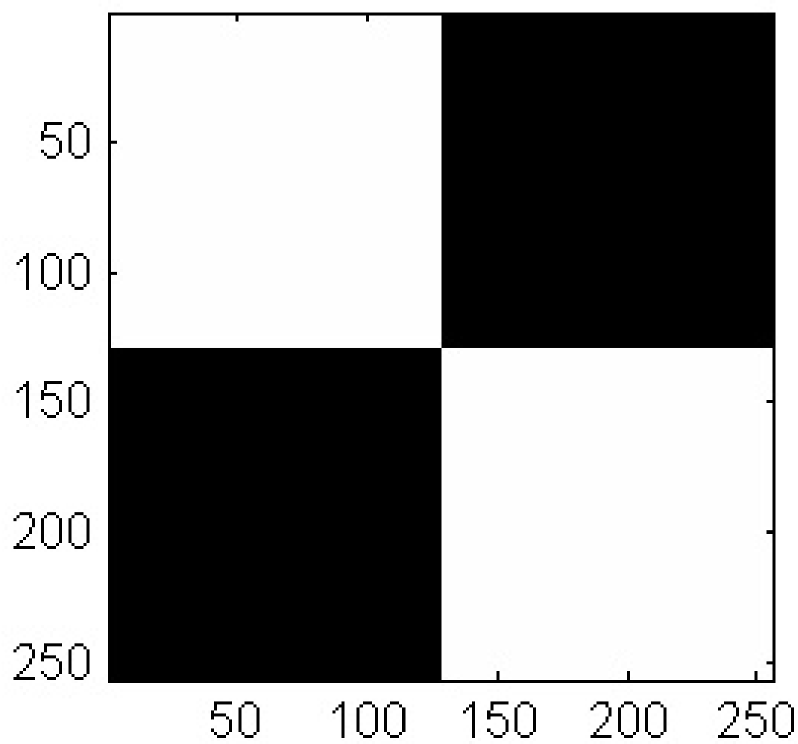


A

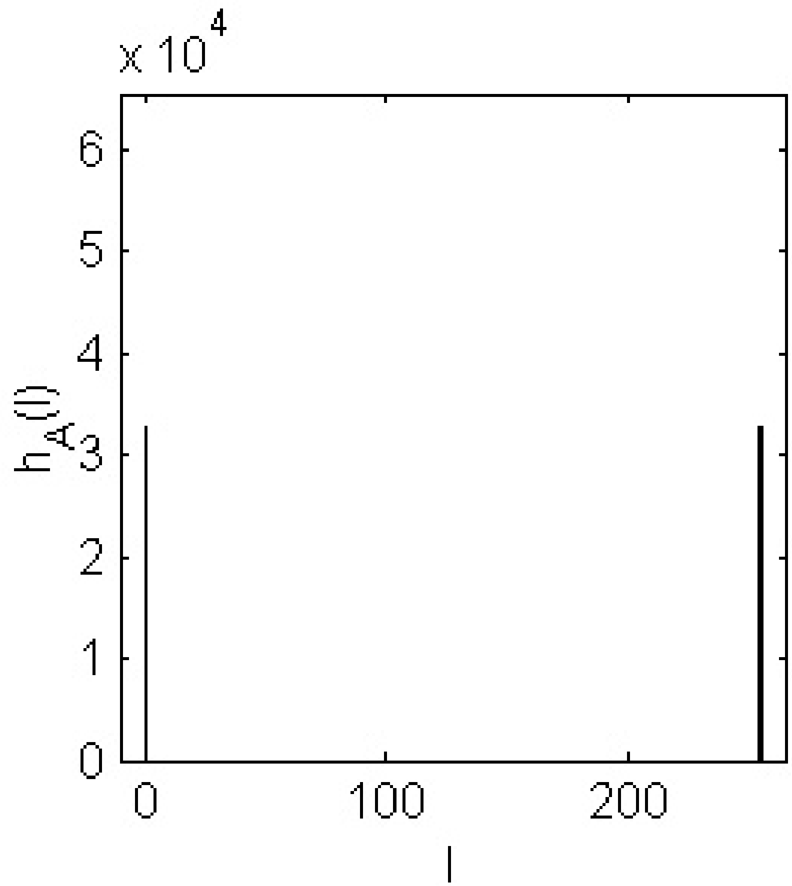


$h_A(l)$

Example II



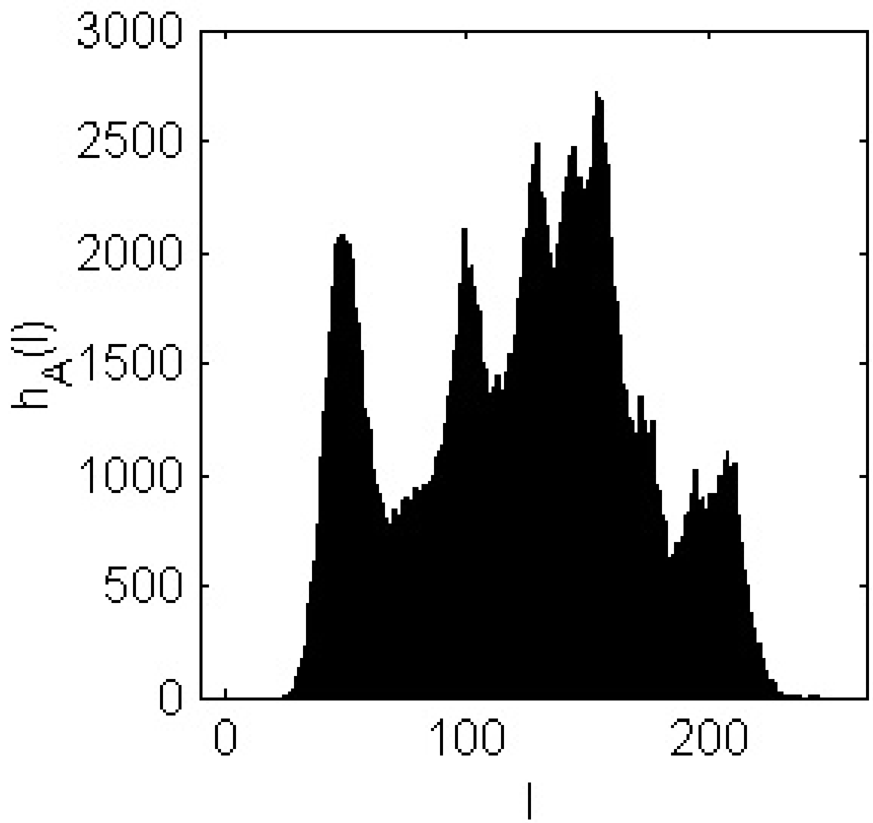
A



$h_A(l)$

Example III

A



$h_A(l)$

Point Processing

- We will now utilize a "function" $g(l)$ $(l = 0, \ldots, 255)$ to generate a new image $\mathbf{B}$ from a given image $\mathbf{A}$ via:

$$B(i,j) = g(A(i,j)), \quad i = 0, \ldots, N-1, \; j = 0, \ldots, M-1 \tag{5}$$

- The function $g(l)$ operates on each image pixel or each image point independently.
- In general the resulting image $g(A(i,j))$ may not be an image matrix, i.e., it may be the case that $g(A(i,j)) \notin \{0, \ldots, 255\}$ for some (i,j).
- Thus we will have to make sure we obtain an image $\mathbf{B}$ that is an image matrix.
- The histograms $h_A(l)$ and $h_B(l)$ will play important roles.

Identity point function

- Let $g(l) = l \ (l = 0, \ldots, 255)$.

$$\begin{aligned} B(i,j) &= g(A(i,j)), \quad i = 0, \ldots, N-1, \ j = 0, \ldots, M-1 \\ &= A(i,j) \end{aligned}$$

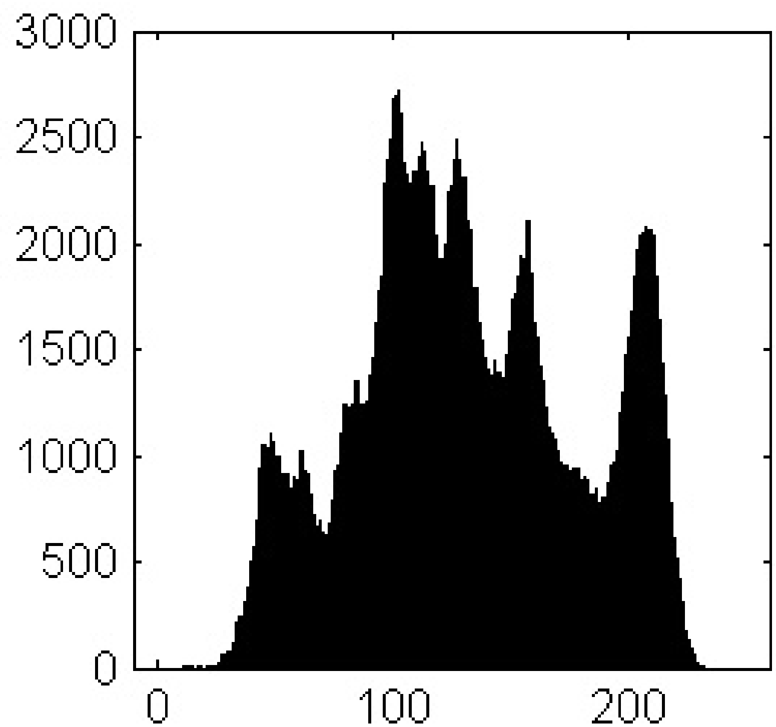
- In this case $g(A(i,j)) \in \{0, \ldots, 255\}$ so no further processing is necessary to ensure $\mathbf{B}$ is an image matrix.
- Note also that $\mathbf{B} = \mathbf{A}$ and hence $h_B(l) = h_A(l)$.

Digital Negative

- Let $g(l) = 255 - l \ (l = 0, \ldots, 255)$. ?

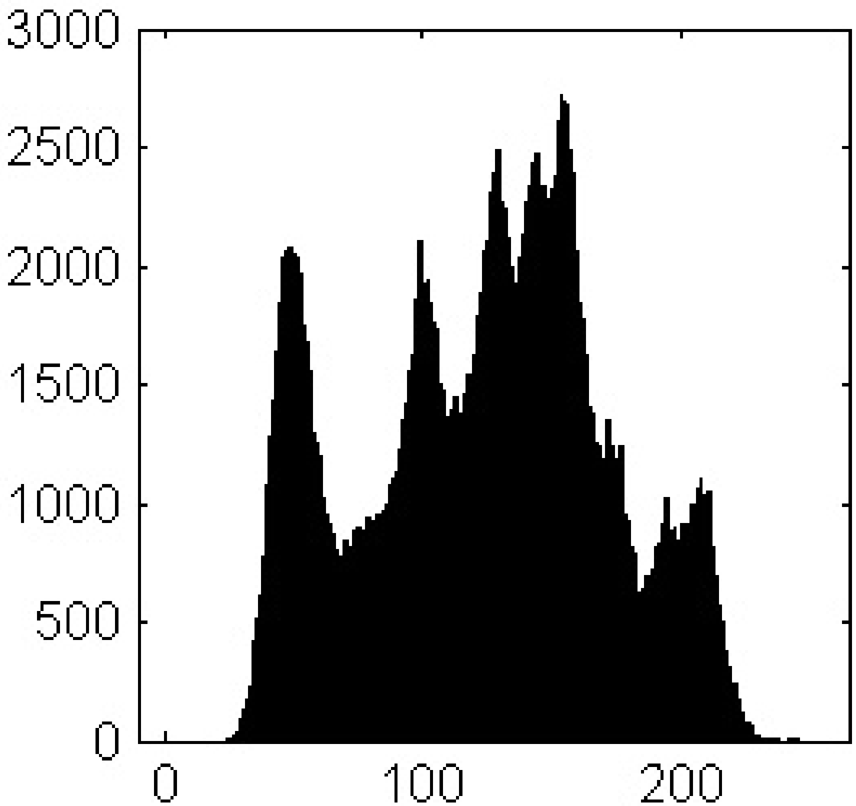
$$\begin{aligned} B(i,j) &= g(A(i,j)), \quad i = 0, \ldots, N-1, \ j = 0, \ldots, M-1 \\ &= 255 - A(i,j) \end{aligned}$$

B

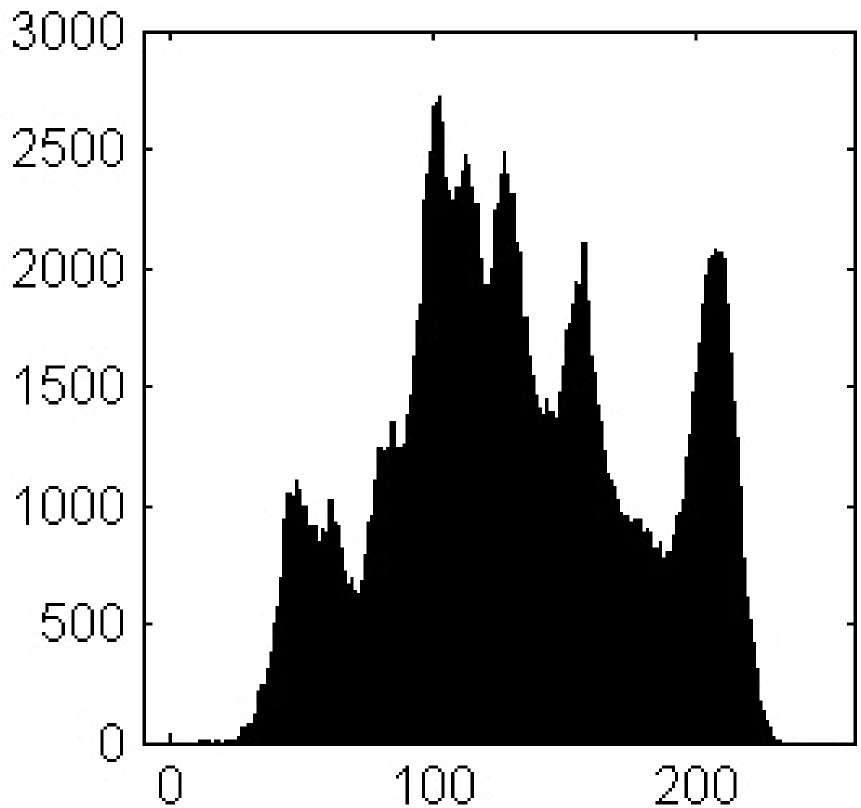


$h_B(l)$

Digital Negative Contd.



$h_A(l)$



$h_B(l)$

- In this case it is easy to see that $h_B(255 - l) = h_A(l)$ or $h_B(g(l)) = h_A(l)$.