EE421/521 Image Processing

Lecture 2
IMAGE ENHANCEMENT

1

Introduction

2

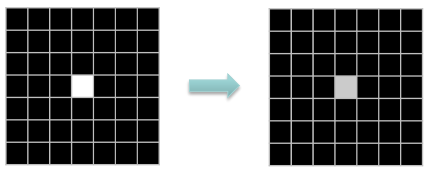
Image Processing Operations

- Spatial domain operations
  - Point operations
  - Neighborhood operations
  - Operations combining multiple images
- Transform domain operations

Point Operations

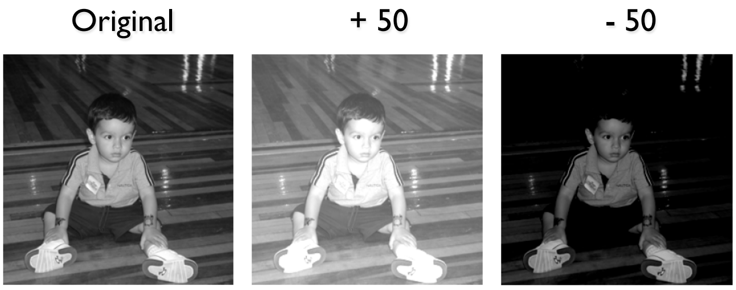
- Transformation depends on the value of the pixel only, regardless of the values of its neighbors.

$$r(x,y) = T\left[s(x,y)\right]$$



Example: Adding A Constant ➔ Brightness Change

- Image has more brightness (larger intensity values) if the additive constant is positive.



By Oge Marques

Example: Multiplying with a Constant ➔ Contrast Change

- Image has more contrast (larger range of intensity values) if the multiplicative constant is larger than one.

By Oge Marques

Handling Overflow and Underflow

- Handling overflow:
  - Truncate the results to the maximum number allowed
  - Scale(normalize) the results to the maximum range:

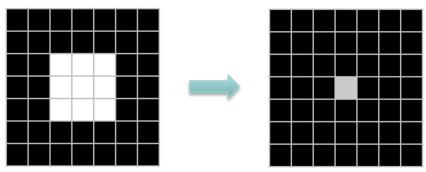
$$g = \frac{L_{max}}{f_{max} - f_{min}}(f - f_{min})$$

- Handling underflow:
  - Truncate negative values to zero
  - Use the absolute value of the difference
  - Add a positive constant to the difference

Neighborhood Operations

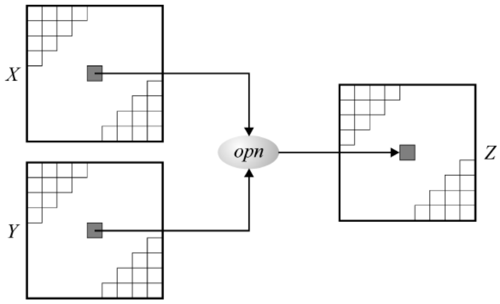
- Transformation depends on the value of the pixel as well as the values of its neighbors.

$$r(x,y) = T\left[s(x,y) : (x,y) \in W\right]$$



Operations Combining Multiple Images

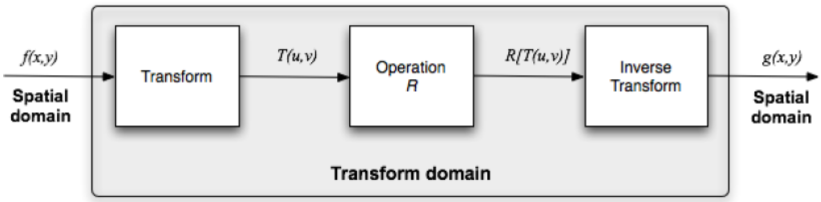
- Two or more images combined pixel-by-pixel, using an arithmetic or logical operator, resulting in a third image



By Oge Marques

Transform Domain Operations

- Operations carried out in a transform (e.g., Fourier transform) domain.

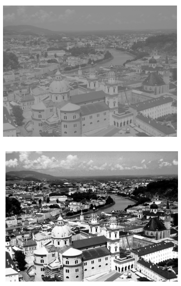


By Oge Marques

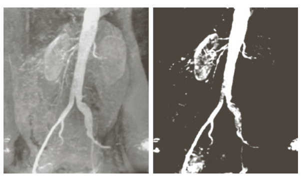
Image Enhancement

Image Enhancement Goals

- to improve the subjective quality of an image for human viewing



- to modify the image in such a way as to make it more suitable to further analysis by a human or a computer



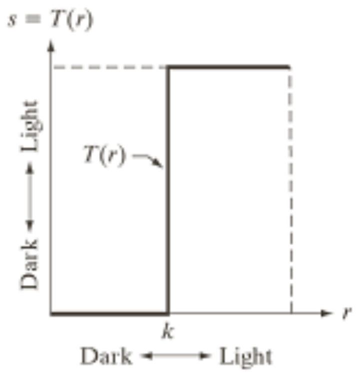
Main Techniques

- Intensity transformations
  - Thresholding
  - Intensity-level slicing
  - Contrast stretching
  - Gamma correction
  - Histogram equalization
- Spatial filtering
  - Unsharp masking

13

Thresholding

- r: input intensity value
- T(r): output intensity value
- k: intensity threshold



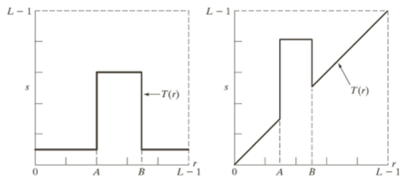
14

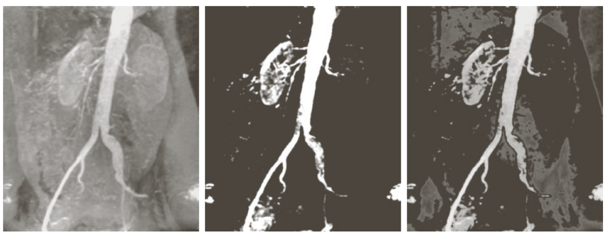
Thresholding

15

Intensity-Level Slicing

A range of intensity levels is highlighted in the output image, while all other values are suppressed or remain untouched.



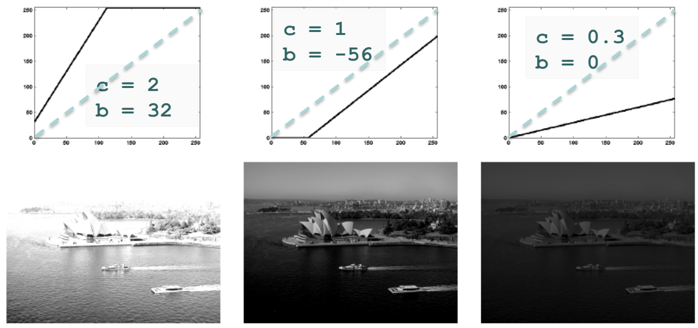


16

Linear Transformations

$s = c \cdot r + b$

c: controls **contrast**, slope
b: impacts overall **brightness**, y-intercept.



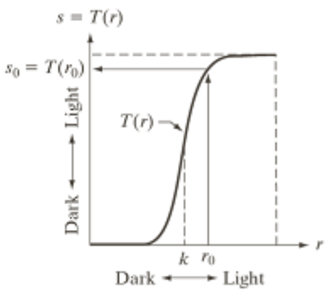
Overall **brigthening**, many saturated pixels

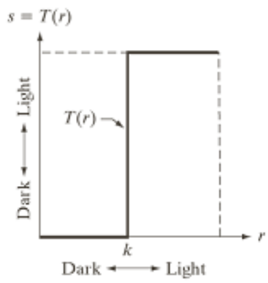
Overall darkening

Significant **contrast** reduction
Overall darkening

Contrast Stretching

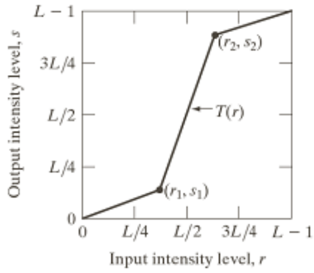
- Small pixel values are compressed toward darker values while large pixel values are pushed toward brighter pixel values.
- Extreme case: binary thresholding





18

Contrast Stretching using Piecewise-Linear Transformations

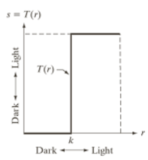


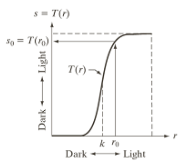
Thresholding vs. Contrast Stretching

Original

Thresholded

Contrast Stretched

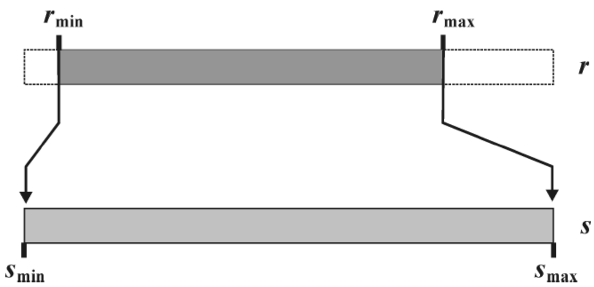




20

Auto-Contrast Adjustment

$$s = \frac{L-1}{r_{max} - r_{min}} \cdot (r - r_{min})$$



Auto-Contrast Example

```
I = imread('salzburg_before.png');
f = double(I);
fmin = min(min(f))
fmax = max(max(f))
g = uint8(255*(f-fmin)/(fmax-fmin));
imshow(g)
```

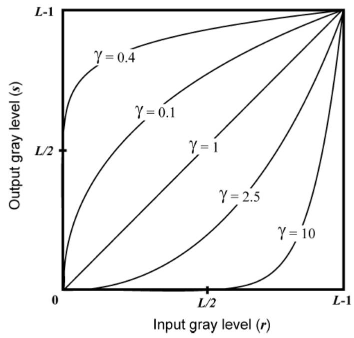
Original image

Adjusted image

Power-Law (Gamma) Transformation

$$s = c \cdot r^{\gamma}$$

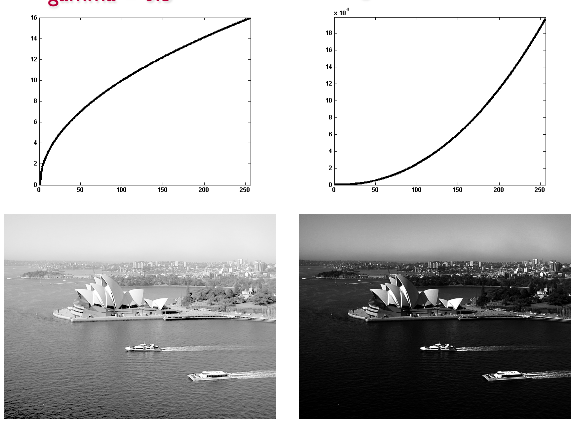
- For gamma < 1, the output image is brighter than the input image
- For gamma > 1, the output image is darker than the input image



If *r* is between 0 and 255, choose *c* such that *s* is also between 0 and 255. Alternatively, use normalized *r* and *s,* i.e., 0<=*r*,*c*,<1, in which case *c*=1.

Gamma Transformation

gamma = 0.5 gamma = 2.2



Gamma Correction

Scanner, printers, and display devices have a power-law response to input intensity:

$$s = r^{\gamma}$$

In particular for a CRT:

$$s = r^{2.5}$$

Gamma correction (preprocessing):

$$s = r^{0.4}$$

25

Gamma Correction

original image

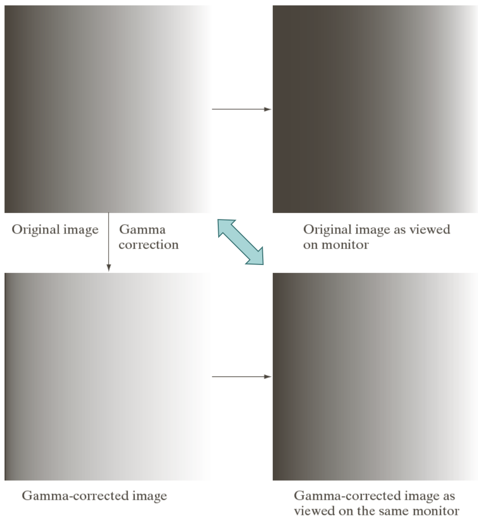
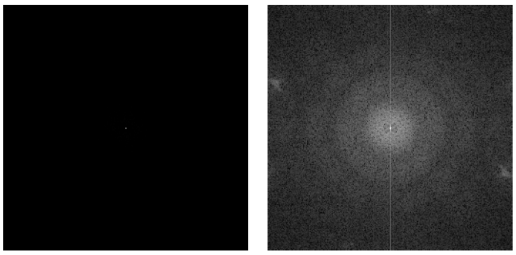


Image displayed on the CRT monitor looks the same as the original image

Log Transformation



(a) (b)

$$s = c \cdot \log(1 + r)$$

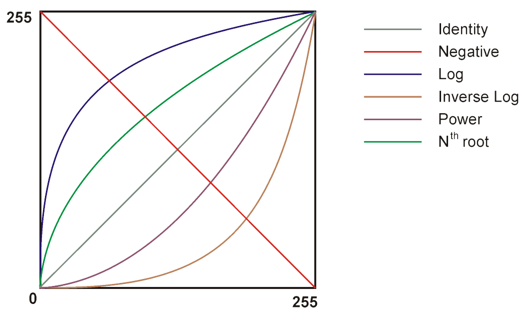
a) Dynamic range is [0, 28591].

b) After log transform, the dynamic range becomes [0, 10.26], which is linearly scaled to [0,255]. Details become noticeable.

Summary of Intensity Transformations

- Linear (inc. negative)
- Piecewise linear (e.g., gray level slicing)
- Nonlinear (e.g., gamma correction, Nth root, log, power, inverse log)

$$s = T[r]$$

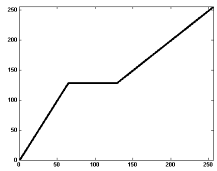


Using Look-Up Tables (LUTs)

For monochrome images with 256 gray levels, the LUT will be a 1D array of length 256.

Example:

$$s = \begin{cases} 2 \cdot f & \text{for } 0 < r \leq 64 \\ 128 & \text{for } 64 < r < 128 \\ f & \text{for } r \geq 128 \end{cases}$$




```
% Building the LUT
LUT = uint8(zeros([1 256]));
LUT(1:65) = 2*(0:64);
LUT(66:129) = 128;
LUT(130:256) = (130:256)-1;
% Applying the LUT
B = intlut(A, LUT)
```

Nonlinear Transformation Using LUTs

```
% Without a LUT
I = imread('klcc_gray.png');
tic
I2 = double(I);
J = 5*sqrt(I2);
O = uint8(J);


% With a LUT
I = imread('klcc_gray.png');
tic
LUT = double(zeros([1 256]));
LUT(1:256) = 5 * sqrt(0:255);
LUT_int8 = uint8(LUT);
O = intlut(I, LUT_int8);
```

Image Histogram

31

Questions

- What is the histogram of an image?
- How can the histogram be computed?
- What is histogram equalization?
- How can the image be modified to match a given histogram?

What is a histogram?

- The histogram of a monochrome image is a representation of the **frequency of occurrence of each intensity level** in the image.
- The data structure that stores the frequency values is a 1D array of numerical values, *h(r)*, whose individual elements store the number (or percentage) of image pixels that correspond to each possible intensity level *r*.

Histogram of an Image

$$h(r) = n$$

r: image intensity value (0,1,…,255)

n: total number (0,1,…,MN-1) of pixels in the image having the intensiy value r

$$\sum_{i,j} h(r) = MN$$

$$f(r) = \frac{1}{MN} h(r)$$

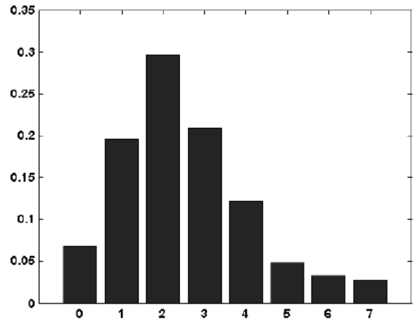
Probability density function (pdf) of image intensities

34

Histogram Example

- Histogram for a hypothetical image containing 128×128 pixels and 8 intensity levels.

Gray level (r_k)	n_k	$p(r_k)$
0	1120	0.068
1	3214	0.196
2	4850	0.296
3	3425	0.209
4	1995	0.122
5	784	0.048
6	541	0.033
7	455	0.028
Total	16384	1.000



Computing Image Histograms

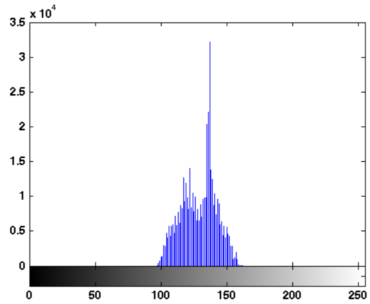
- An array of B elements is created, where B is the number of histogram bins (or buckets). B <= K, where K is the number of gray levels in the image. It is common to choose B=K.
- The histrogram (number of pixels in each bin) is calculated as:

$$h(j) = card\{(x,y) | r_j \leq f(x,y) < r_{j+1}\}, \quad \text{for } 0 \leq j \leq B$$

$$r_j = j.\frac{K}{B}$$

Example: Low Contrast Image

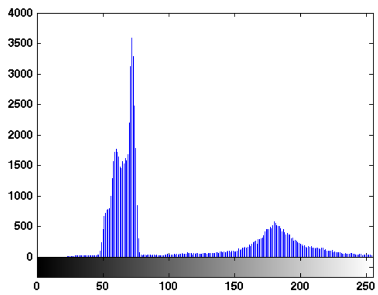
- In MATLAB: **`imhist`**



Pixels are grouped around intermediate gray-level values, indicating an image with low contrast

By Oge Marques

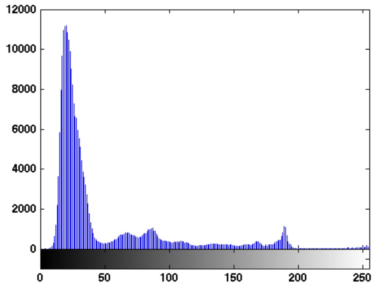
Example: Bimodal Image



- Two distinct hills.
- Image has high contrast since the two modes are well seperated from each other.

By Oge Marques

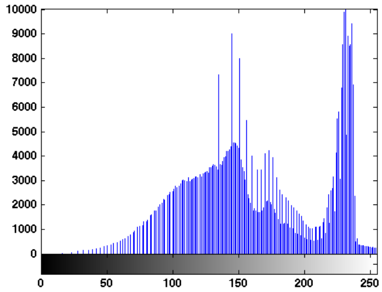
Example: Dark Image



- Histogram is concentrated in lower gray levels, which corresponds to a moslty dark image.

By Oge Marques

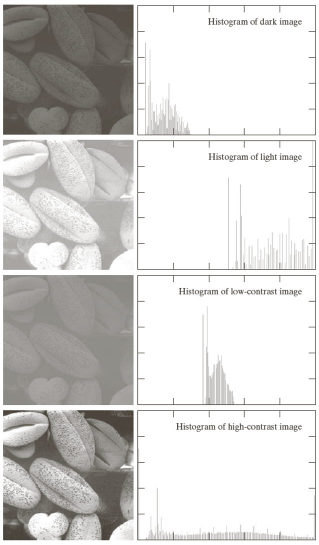
Example: Bright Image



- Histogram is grouped close to the higher gray-levels, which corresponds to a bright image.

By Oge Marques

Histogram Comparison



Dark

Light

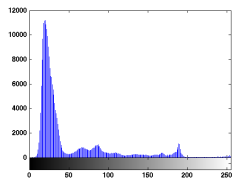
Low contrast

High contrast

41

Utilization of Histograms

- Contemporary digital cameras have an optional real-time histogram overlay in their viewfinder.
- This information prevents taking underexposed or overexposed pictures.



Example:
Too dark ➔ increase exposure

Summary

- Histograms provide a **statistical** representation of the intensity distribution in an image.
- Histograms can be used to evaluate image attributes such as minimum, average, and maximum intensity values, overall contrast and average brightness, and dominance of bright or dark pixels.
- Histograms do **not** contain any information about the spatial distribution of the pixels.
- Histograms can be **modified** to enhance the appearence of an image.

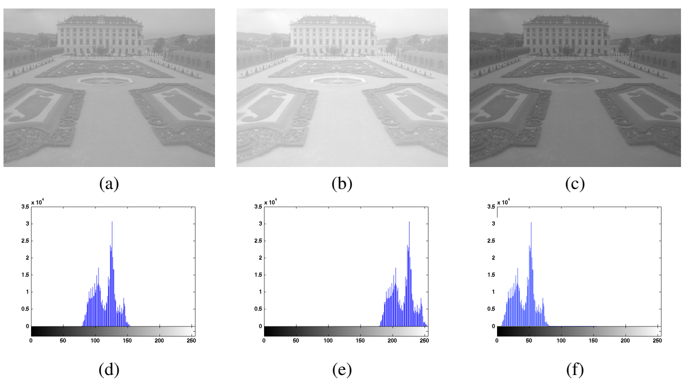
Histogram Modification Techniques

Histogram Modification Techniques

- Histogram sliding
- Histogram stretching
- Histogram shrinking
- Histogram equalization
- Histogram mathing
- Adaptive histogram equalization

Histogram Sliding

- Same as adding or subtracting a constant value (brightness change)

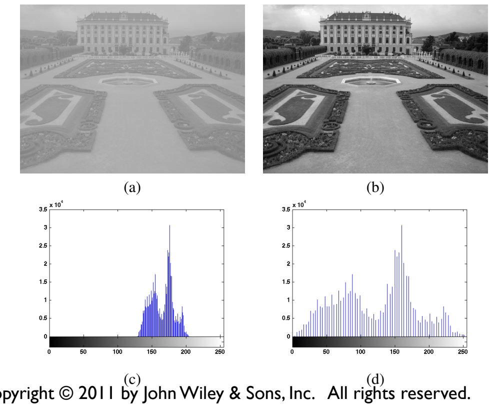


By Oge Marques

Histogram Stretching

- Expand the part of the histogram such that its nonzero intensity range occupies the full dynamic gray scale. (same as autocontrast)

$$s = \frac{r - r_{min}}{r_{max} - r_{min}} \cdot (L - 1)$$

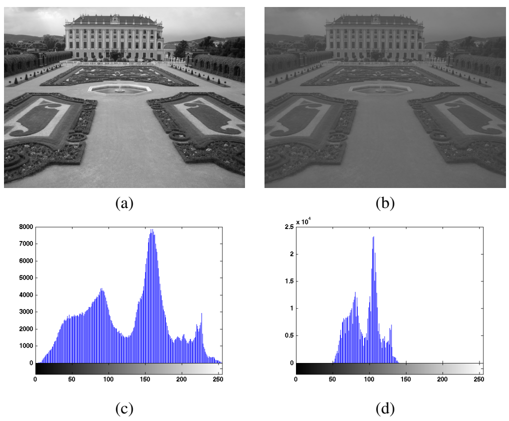


By Oge Marques

Histogram Shrinking

- Compress the dynamic range

$$s = \left[\frac{s_{max} - s_{min}}{r_{max} - r_{min}}\right](r - r_{min}) + s_{min}$$



By Oge Marques

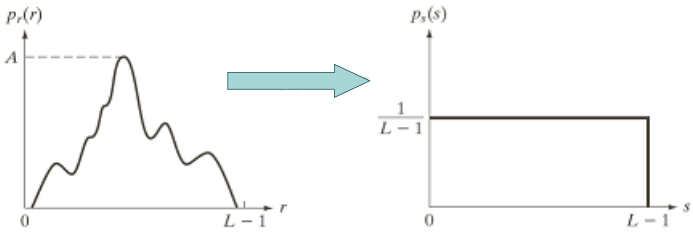
Histogram Equalization

Histogram Equalization

- Map the pixel intensities in an image so as to obtain a uniform (flat) resulting histogram, in which the percentage of pixels of every intensity level is the same.

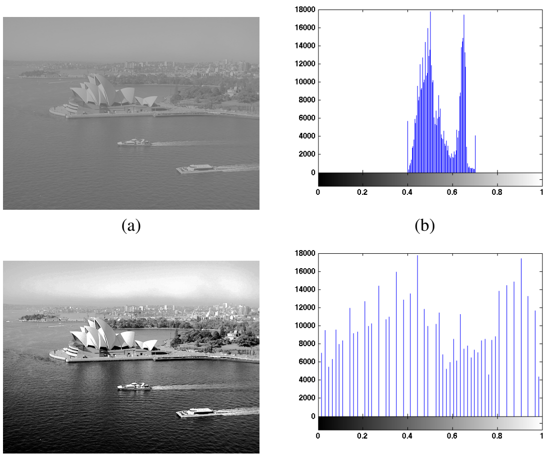
Histogram Equalization

Find an intensity mapping $s=T(r)$ such that the new intensity values have a flat probability density function (pdf):



51

Histogram equalization



(a) (b)

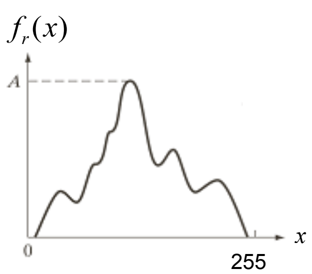
(c) (d)

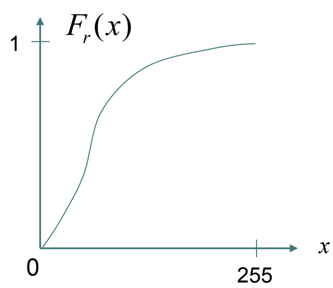
By Oge Marques

MATLAB: **`histeq`**

```
I = imread('sydney_low_contrast.png');
I = rgb2gray(I);
J = histeq(I);
```


pdf vs. cdf

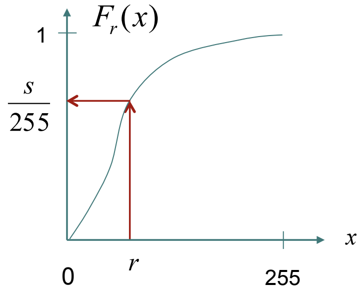




54

Finding s Given r

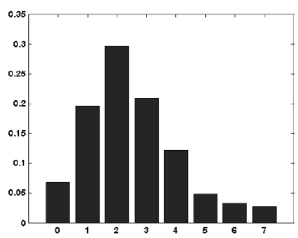
1. Calculate the histogram $h(r)$ of the image
2. Evaluate the cdf $F_r(x)$
3. Use the cdf to map r to s



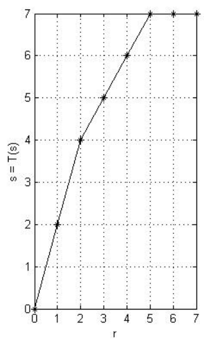
$$T(x) = 255 \cdot F_r(x)$$

55

Example: Histogram Equalization



Gray level (r_k)	n_k	$p(r_k)$
0	1120	0.068
1	3214	0.196
2	4850	0.296
3	3425	0.209
4	1995	0.122
5	784	0.048
6	541	0.033
7	455	0.028
Total	16384	1.000



$$s_i = T(r_i) = \text{nearest integer}\left[7 \times \sum_{j=0}^{i} p(r_j)\right]$$

Example (continued)

- Find the cdf

Gray level (r_k)	n_k	$p(r_k)$
0	1120	0.068
1	3214	0.196
2	4850	0.296
3	3425	0.209
4	1995	0.122
5	784	0.048
6	541	0.033
7	455	0.028
Total	16384	1.000

$$s_0 = T(r_0) = \sum_{j=0}^{0} p(r_j) = p(r_0) = 0.068$$

$$s_1 = T(r_1) = \sum_{j=0}^{1} p(r_j) = p(r_0) + p(r_1) = 0.068 + 0.196 = 0.264 \to 2$$

$$s_2 = \sum_{j=0}^{2} p(r_j) = 0.560 \times 7 = 3.92 \cong 4,$$

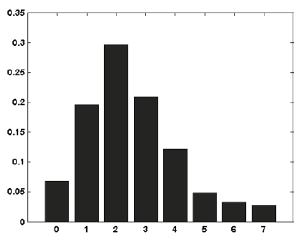
$$s_3 = 0.769 \times 7 = 5.383 \cong 5, s_4 = 0.891 \times 7 = 6.237 \cong 6,$$

$$s_5 = 0.939 \to 7, s_6 = 0.972 \to 7, s_7 = 1 \to 7$$

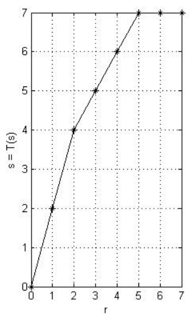
- Since the image was quantized with eight gray levels, multiply each s_k with 7 and round to the nearest integer:

$$s_0 \cong 0, s_1 \cong 2, s_2 \cong 4, s_3 \cong 5, s_4 \cong 6, s_5 \cong 7, s_6 \cong 7, s_7 \cong 7$$

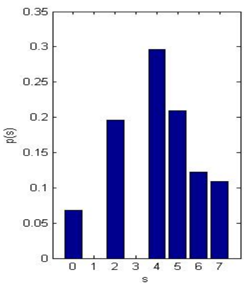
Example (Continued)



Original histogram



Transformation function



Equalized histogram

The equalized histogram is not perfectly flat, but it is the best possible result for this particular image.

Derivation

Let *r* and *s* denote original and transformed pixel values.
For an 8-bit image, we have $0 \le r,\ s \le 255$.

Find the transformation function $s=T(r)$ such that

$$f_s(x) = \text{constant} \quad \Rightarrow \quad F_s(x) = \frac{x}{255}$$

where f() denotes the probability density function (pdf) and
F denotes the cumulative distribution function (cdf).

$$F_s(x) = P(s \le x) = P(T(r) \le x) = P(r \le T^{-1}(x)) = F_r(T^{-1}(x)) = \frac{x}{255}$$

59

T(·)?

$$F_s(x) = F_r(T^{-1}(x)) = \frac{x}{255}$$

$$F_r^{-1}\left(\frac{x}{255}\right) = T^{-1}(x) = y$$

$$F_r(y) = \frac{x}{255} \qquad\qquad x = T(y)$$

$$T(y) = 255 \times F_r(y)$$

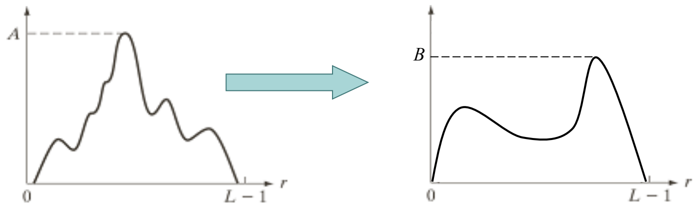
60

Histogram Matching

61

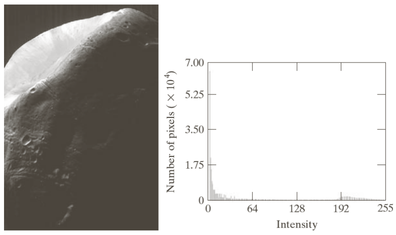
Histogram Matching

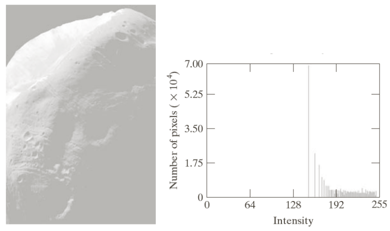
Find an intensity mapping $s = T(r)$ such that the new intensity values have an arbitrarily different probability density function:



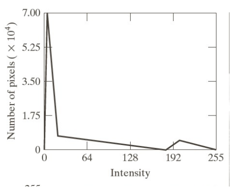
62

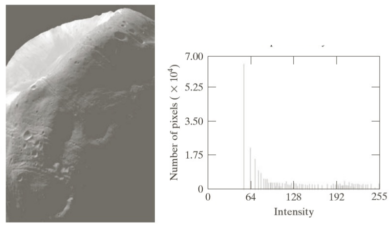
Histogram Equalization vs. Histogram Matching





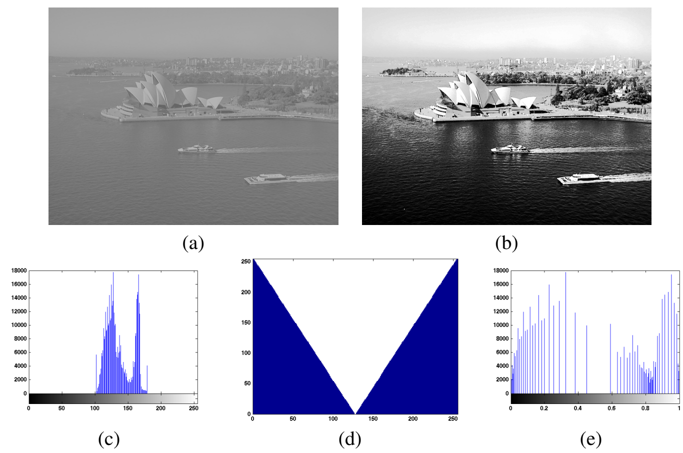
Specified histogram





63

Histogtram Specification



By Oge Marques

MATLAB: **histeq**

```
I = imread('sydney_low_contrast.png');
Id = im2double(I);
figure, imhist(Id), ylim('auto'), ...
       title ('Original histogram');

des_hist = uint8(zeros(1,256));
des_hist(1:128) = linspace(256,0,128);
des_hist(129:end) = linspace(0,256,128);
x_axis = 0:255;
figure, bar(x_axis, des_hist), axis tight, ...
       title('Desired histogram');

hgram = im2double(des_hist);
Jd = histeq(Id,hgram);
figure, imhist(Jd), ylim('auto'), ...
       title ('Resulting histogram');
```


T(·)?

$$F_s(x) = F_r(T^{-1}(x))$$

⇩

$$F_r^{-1}(F_s(x)) = T^{-1}(x)$$

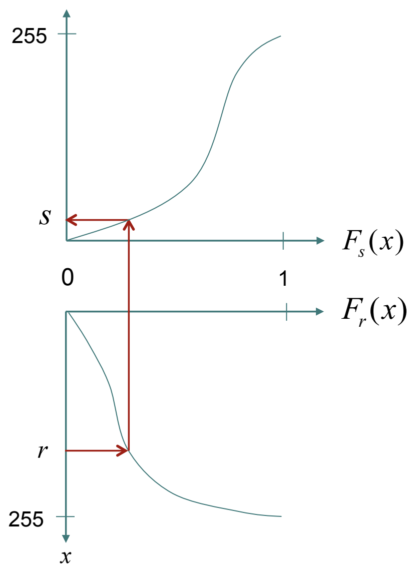
⇩

$$T(\underbrace{F_r^{-1}(F_s(x))}_{r}) = x$$

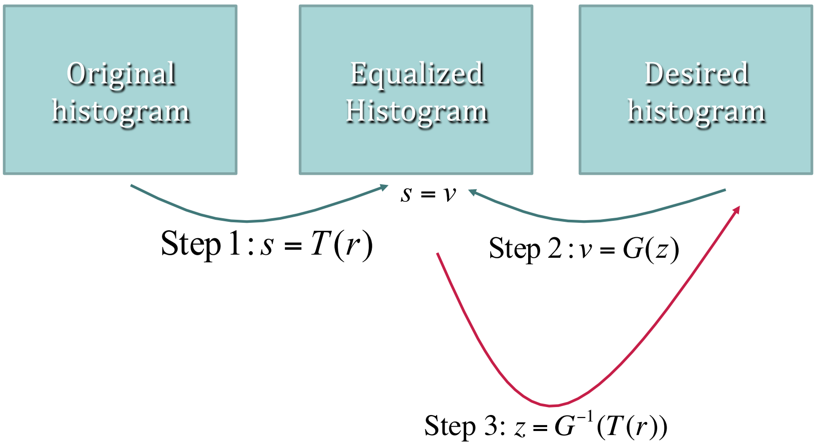
(s)

⇩

$$s = F_s^{-1}(F_r(r))$$



Histogram Specification (Matching)

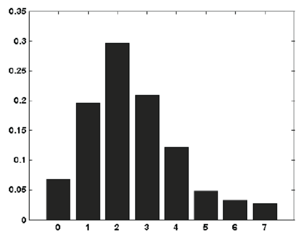


By Oge Marques

Example: Histogram Specification

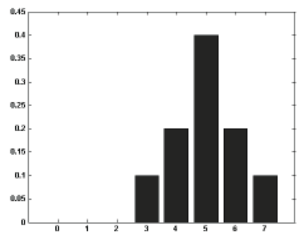
Original histogram

Gray level (r_k)	n_k	$p(r_k)$
0	1120	0.068
1	3214	0.196
2	4850	0.296
3	3425	0.209
4	1995	0.122
5	784	0.048
6	541	0.033
7	455	0.028
Total	16384	1.000



Desired histogram

Gray Level (s_k)	n_k	$p(s_k)$
0	0	0.0
1	0	0.0
2	0	0.0
3	1638	0.1
4	3277	0.2
5	6554	0.4
6	3277	0.2
7	1638	0.1
Total	16384	1.0



By Oge Marques

Example (continued)

- 1. Histogram equalized image has been found before:

$$s_0 \cong 0, s_1 \cong 2, s_2 \cong 4, s_3 \cong 5, s_4 \cong 6, s_5 \cong 7, s_6 \cong 7, s_7 \cong 7$$

- 2. Obtain the cdf of the desired probability mass function:

$$v_k = G(z_k) = \sum_{j=0}^{k} p(z_j)$$

$$v_0 = 0, v_1 = 0, v_2 = 0, v_3 = 0.1, v_4 = 0.3, v_5 = 0.7, v_6 = 0.9, v_7 = 1$$

Example (Continued)

- 3. Apply the inverse transformation function:

$$z = G^{-1}(s)$$

Search for each value of s_k, the closest value of v_k:
E.g., for $s_1 = 2/7 \cong 0.286$, the closest value of v_k is $v_4 = G(z_4) = 0.3$.
In inverse function notation $z_4 = G(0.3)$

$$s_0 \cong 0, s_1 \cong 2, s_2 \cong 4, s_3 \cong 5, s_4 \cong 6, s_5 \cong 7, s_6 \cong 7, s_7 \cong 7$$

$$v_0 = 0, v_1 = 0, v_2 = 0, v_3 = 0.1, v_4 = 0.3, v_5 = 0.7, v_6 = 0.9, v_7 = 1$$

Example (Continued)

- The remaining values of s_k are mapped:

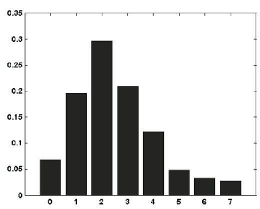
$$s_0 = 0 \rightarrow z_2$$
$$s_1 = 2/7 \cong 0.286 \rightarrow z_4$$
$$s_2 = 4/7 \cong 0.571 \rightarrow z_5$$
$$s_3 = 5/7 \cong 0.714 \rightarrow z_5$$
$$s_4 = 6/7 \cong 0.857 \rightarrow z_6$$
$$s_5 = 1 \rightarrow z_7$$
$$s_6 = 1 \rightarrow z_7$$
$$s_7 = 1 \rightarrow z_7$$

Matched Histogram

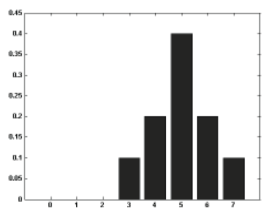
$$p(z_0) = 0.0$$
$$p(z_1) = 0.0$$
$$p(z_2) = 0.0$$
$$p(z_3) = 0.1$$
$$p(z_4) = 0.2$$
$$p(z_5) = 0.4$$
$$p(z_6) = 0.2$$
$$p(z_7) = 0.1$$

Direct Histogram Specification

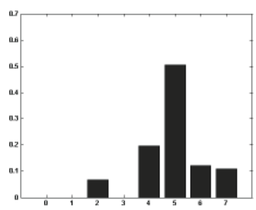
Original



Desired



Result



By Oge Marques

Adaptive (Local) Histogram Equalization

73

Local Histogram Equalization

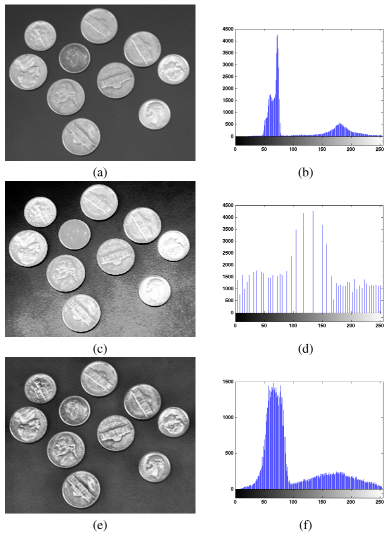
- Use a sliding (rectangular/square) **window**, which moves across the image.
- Compute the histogram in the window and compute the mapping function.
- Map the **center** pixel of the window using the mapping function.
- Much more computationally expensive than global histogram equalization.

Global vs. Local Histogram Equalization

Original image and its histogram

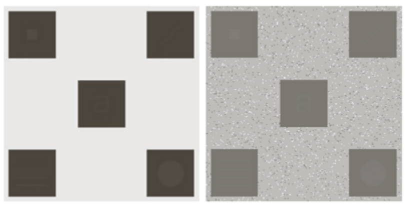
After global histogram equalization

After local histogram equalization. Bimodal nature of the histogram is preserved while still improving the contrast.



By Oge Marques

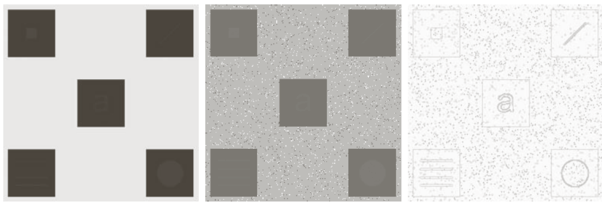
Adaptive (Local) Histogram Equalization



Original Global HE

76

Adaptive (Local) Histogram Equalization



Original   Global HE   Local HE

77

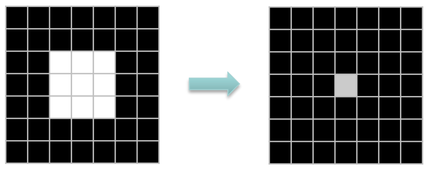
Neighborhood Processing

78

Neighborhood Operations

- Transformation depends on the value of the pixel as well as the values of its neighbors.

$$r(x,y) = T\left[s(x,y) : (x,y) \in W\right]$$



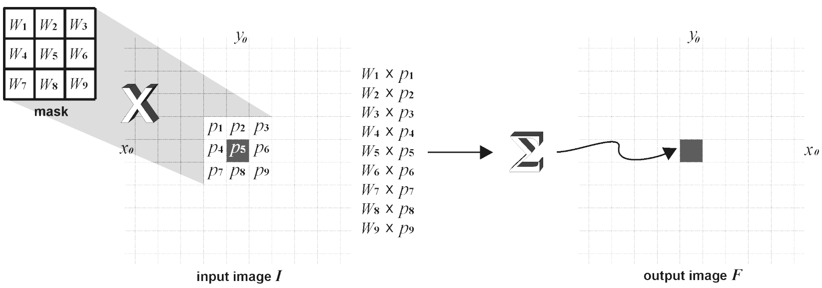
Neighborhood processing

- Main steps:
  - Define a reference point in the input image, $f(x_0, y_0)$.
  - Perform an operation that involves only pixels within a neighborhood around the reference point in the input image.
  - Apply the result of that operation to the pixel of same coordinates in the output image, $g(x_0, y_0)$.
  - Repeat the process for every pixel in the input image.

Convolution

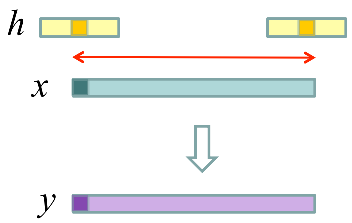
- Convolution is a widely used linear operator that processes an image by computing -- for each pixel -- a weighted sum of the values of that pixel and its neighbors.
  - Depending on the choice of weights a wide variety of image processing operations can be implemented.

Convolution



1-D Convolution

$$y[m] = h[m] * x[m] = \sum_i h[m-i]x[i]$$



83

Example

- Convolve A = {0,1,2,3,2,1,0} and B = {1,3,-1}
  - Mirror array B and align its center (reference) value with the first (leftmost) value of array A.

A		0	1	2	3	2	1	0
B	-1	3	1					
A*B		1						

A		0	1	2	3	2	1	0
B		-1	3	1				
A*B		1	5					

A		0	1	2	3	2	1	0
B			-1	3	1			
A*B		1	5	8				

A		0	1	2	3	2	1	0
B				-1	3	1		
A*B		1	5	8	9			

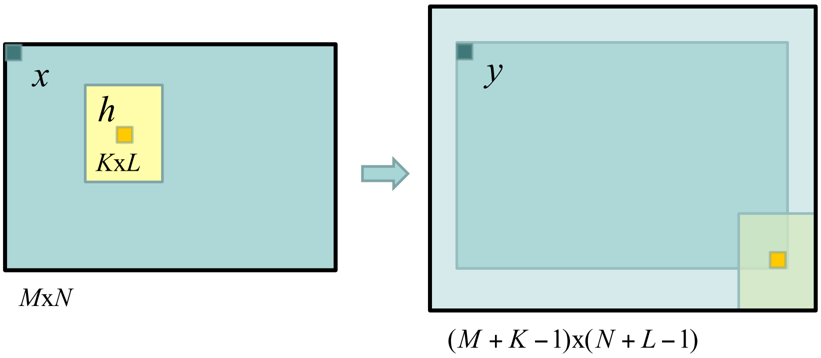
A		0	1	2	3	2	1	0
B					-1	3	1	
A*B		1	5	8	9	4		

A		0	1	2	3	2	1	0	
B							-1	3	1
A*B		1	5	8	9	4	1	-1	

By Oge Marques

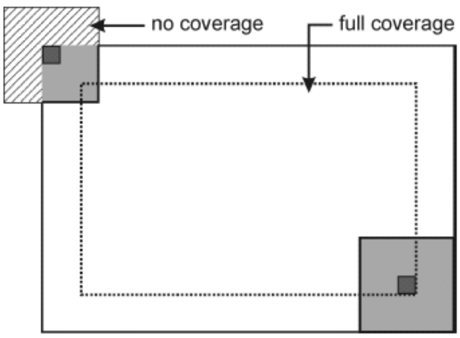
2-D Convolution

$$y[m,n] = \sum_i \sum_j h[m-i, n-j]\,x[i,j]$$



85

Dealing with image borders



By Oge Marques

Dealing with image borders:

1. Keep the pixel values that cannot be reached by the overlapping mask untouched
2. Pad the input image with zeros.
3. Pad with extended (same) values.
4. Pad with mirrored values.
5. Treat the input image as a 2D periodic function whose values repeat themselves in both horizontal and vertical directions.

- In MATLAB: check the **`boundary_options`** parameter for function **`imfilter`**.

By Oge Marques

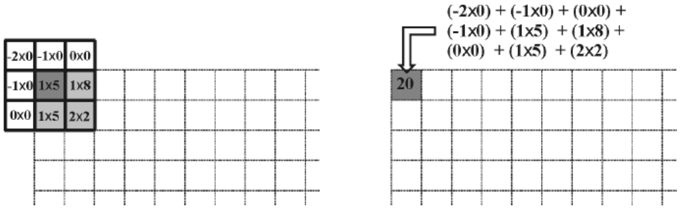
- Convolve A and B:

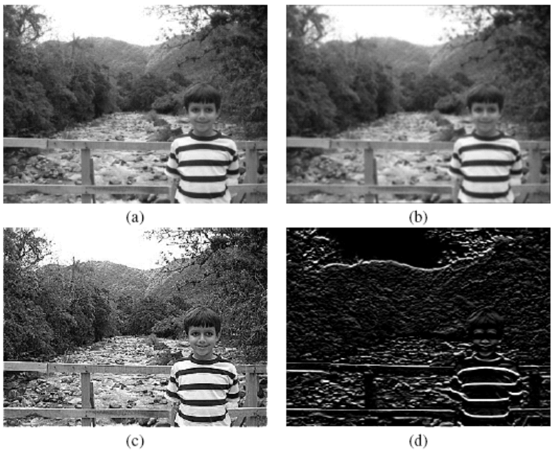
$$A = \begin{bmatrix} 5 & 8 & 3 & 4 & 6 & 2 & 3 & 7 \\ 3 & 2 & 1 & 1 & 9 & 5 & 1 & 0 \\ 0 & 9 & 5 & 3 & 0 & 4 & 8 & 3 \\ 4 & 2 & 7 & 2 & 1 & 9 & 0 & 6 \\ 9 & 7 & 9 & 8 & 0 & 4 & 2 & 4 \\ 5 & 2 & 1 & 8 & 4 & 1 & 0 & 9 \\ 1 & 8 & 5 & 4 & 9 & 2 & 3 & 8 \\ 3 & 7 & 1 & 2 & 3 & 4 & 4 & 6 \end{bmatrix} \quad B = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

- Flipped B: $\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$

- Result A*B:

$$\begin{bmatrix} 20 & 10 & 2 & 26 & 23 & 6 & 9 & 4 \\ 18 & 1 & -8 & 2 & 7 & 3 & 3 & -11 \\ 14 & 22 & 5 & -1 & 9 & -2 & 8 & -1 \\ 29 & 21 & 9 & -9 & 10 & 12 & -9 & -9 \\ 21 & 1 & 16 & -1 & -3 & -4 & 2 & 5 \\ 15 & -9 & -3 & 7 & -6 & 1 & 17 & 9 \\ 21 & 9 & 1 & 6 & -2 & -1 & 23 & 2 \\ 9 & -5 & -25 & -10 & -12 & -15 & -1 & -12 \end{bmatrix}$$





Low-pass filter	High-pass filter	Horizontal edge detection
$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

Separable 2-D Filter

$$y[m,n] = \sum_i \sum_j h[m-i, n-j]x[i,j]$$

separable ⟹ $h[m,n] = h_1[m]h_2[n]$

90

Implementation of a Separable 2-D Filter

$$y[m,n] = \sum_i \sum_j h_1[m-i]h_2[n-j]x[i,j]$$

$$= \sum_j h_2[n-j] \underbrace{\sum_i h_1[m-i]x[i,j]}$$ ← rows

$$= \sum_j h_2[n-j]\bar{x}[m,j]$$ ← columns

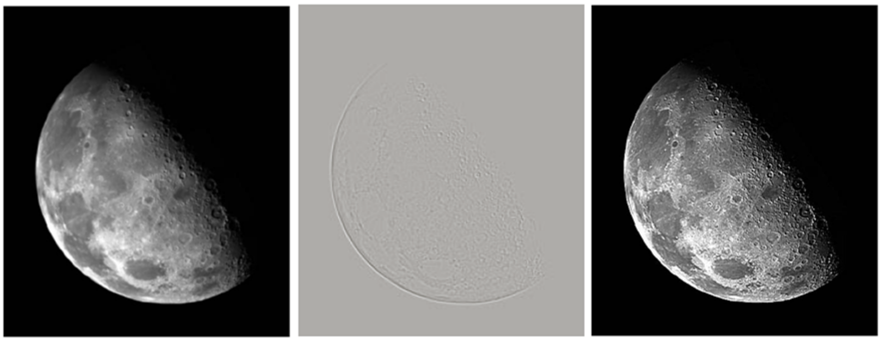
Two 1-D convolution computations: $2N \log N$
Rather than one 2-D convolution: $N^2 \log N^2$

91

Edge Enhancement

92

Image Sharpening with Unsharp Masking

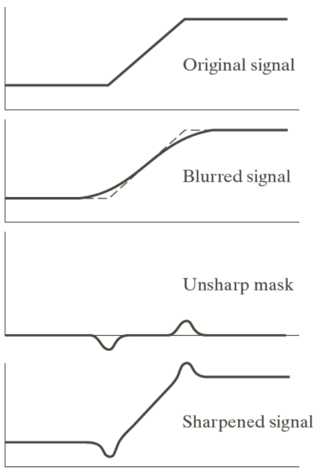


original mask sharpened

Unsharp Masking

original - blurred

original + mask

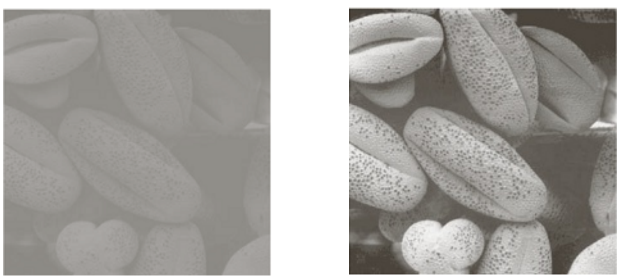


Project 1.2

Image Enhancement

Due 10.10.2013

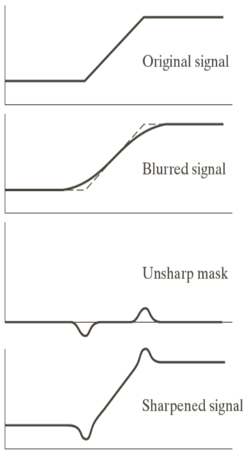
Histogram Equalization



Problem 1: Histrogram Equalization & Matching

1. Pick two images, one with low and the other with high exposure (i.e., a dark and a bright image).
2. Calculate and display the histogram of each image.
3. Apply histogram equalization to each image and display the resulting images and their histograms.
4. Modify the dark image so that its histogram matches that of the bright image. Display the modified image and its histrogram.
5. Comment on the results.

Unsharp Masking



Problem 2: Unsharp Masking

1. Pick an unsharp image.
2. Calculate and display an unsharp mask for the image. You can use a 3x3 mean filter for this purpose.
3. Obtain and display the sharpened image.
4. Compare the original image with the image obtained in Step 3 and comment on any improvements.
5. If using a color image, implement Steps 2 and 3 on all three bands and then combine the sharpened bands.

Next Lecture

- COLOR THEORY