# Enhancement vs Recovery

- **ENHANCEMENT**: The manipulation or transformation of an image, with the aim of increasing its usefulness or visual appearance.
- **RESTORATION/RECOVERY**: The modeling of a degradation the image is subjected to (which resulted in loss of information), and the removal of this degradation (or recovery of the information lost), based on an optimality criterion
- **Examples:**
  - Modification of intensity values so as to increase contrast enhancement
  - Deconvolution restoration
  - Inpainting enhancement or recovery
  - Smoothing or removal of noise enhancement or restoration

# Image Enhancement

- No general “theory” behind enhancement
  - Criteria for enhancement are often **subjective** or too **complex** to be easily converted to useful objective measures
- The algorithms tend to be qualitative & ad hoc and they are application-dependent
- Application-specific evaluation of its effectiveness

# Enhancement Topics

- Point-wise Intensity Transformations
  - Log, Power-law, Piecewise linear
- Histogram Processing
- Spatial Filtering (LSI vs non-linear)
  - Smoothing
  - Sharpening
  - Homomorphic Filtering
- Pseudo-Coloring
- Video Enhancement

# Point-Wise Intensity Transformations

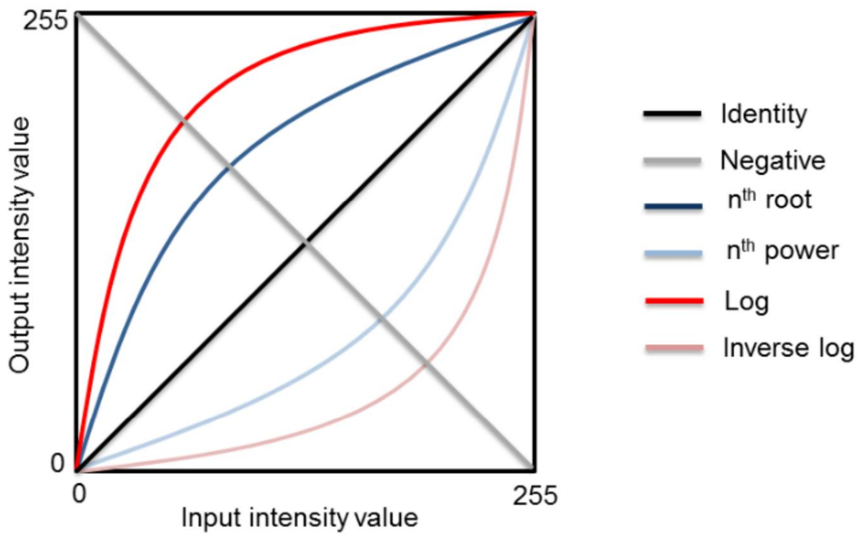
- Expressed by

$$y(n_1, n_2) = \mathbf{T}_{\text{point}}[x(n_1, n_2)]$$

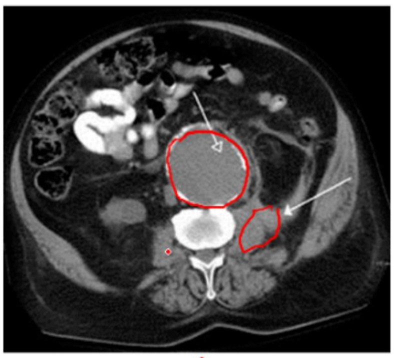
$$\mathbf{y}(n_1, n_2) = \mathbf{T}_{\text{point}}[\mathbf{x}(n_1, n_2)]$$

$$\mathbf{y}(\mathbf{n}) = \mathbf{T}_{\text{point}}[\mathbf{x}(\mathbf{n})]$$

# Intensity Transformations




# Negative Transformation



CT scan

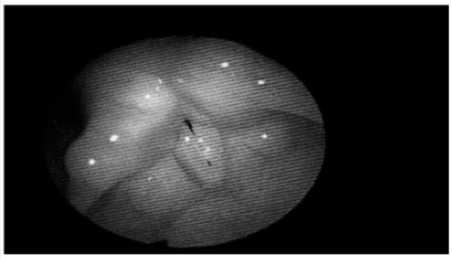
Image source: http://en.wikipedia.org/wiki/File:RupturedAAA.png

# Log Transformation

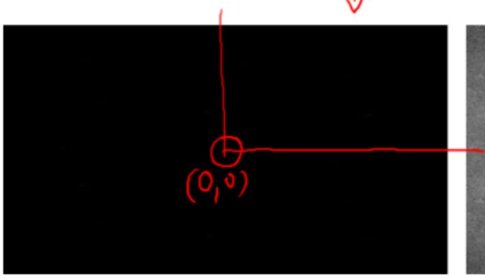
- Log-Transform (expand low – compress high):

$$y(\mathbf{n}) = c \cdot log(x(\mathbf{n}) + 1)$$

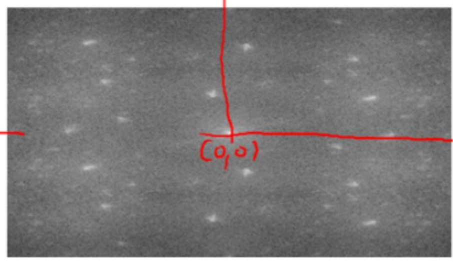
- Used to display Fourier Spectra



*Original*




*Magnitude of Fourier spectrum*



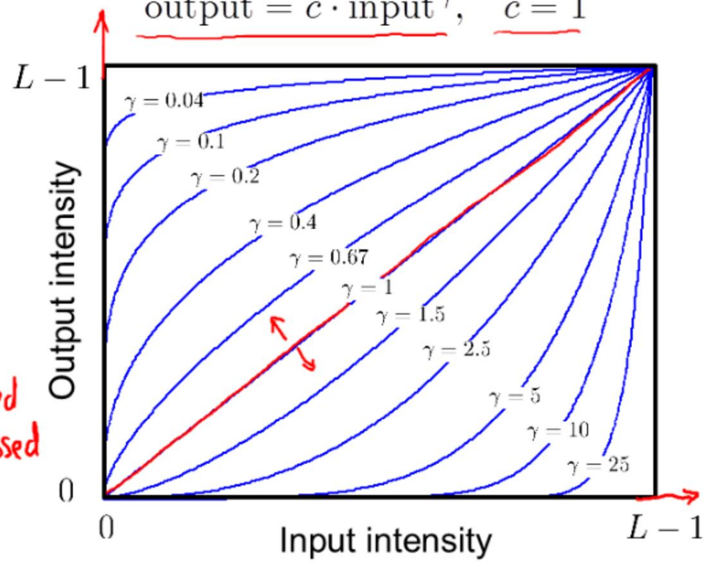

*Log-transform of Fourier spectrum magnitude*

# Power-Law Transformation

$$\text{output} = c \cdot \text{input}^{\gamma}, \quad c = 1$$

$\gamma < 1$

Small values → expanded
high values → compressed



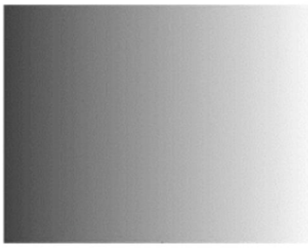

Intensity = voltage$^{\gamma}$

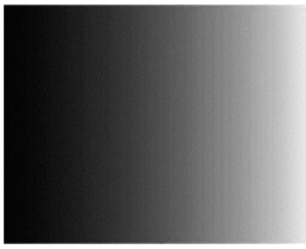
gamma correction

$2^{k} = L$ ($k$: bits per pixel)

# Gamma Correction Example

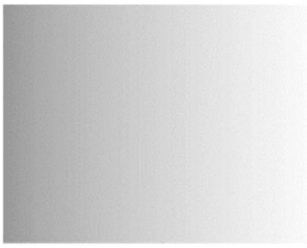
original

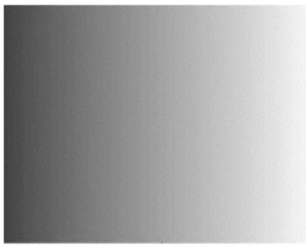




Displayed on a monitor with gamma = 2.5

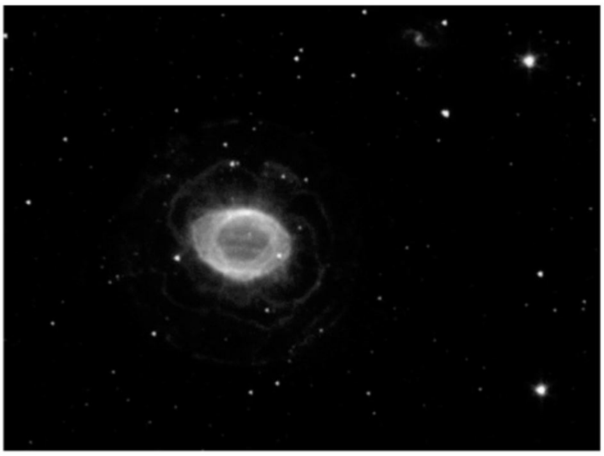
Corrected with gamma = 0.4

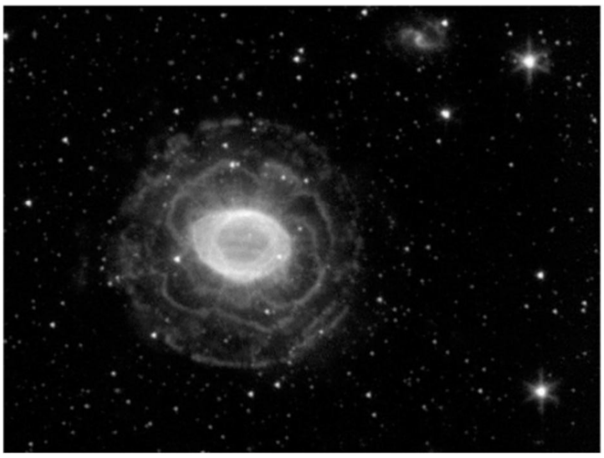




Display of the corrected image

# Power-Law Example

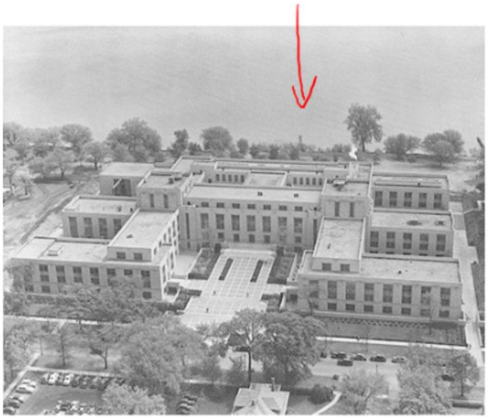


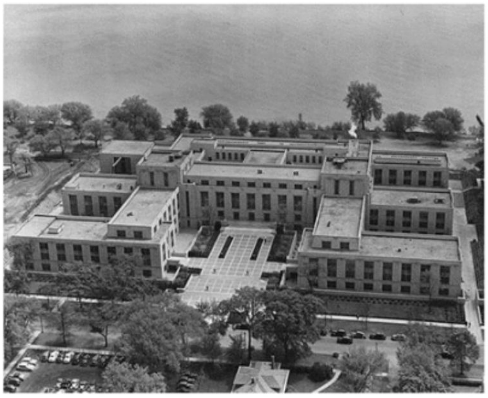


http://www.nasa.gov/multimedia/imagegallery/image_feature_679.html

$\gamma = 0.5$

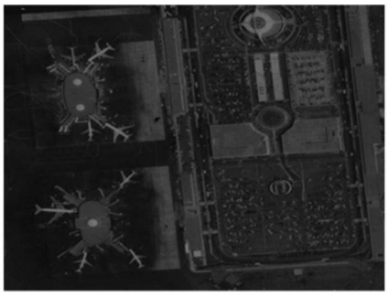
# Power-Law Example

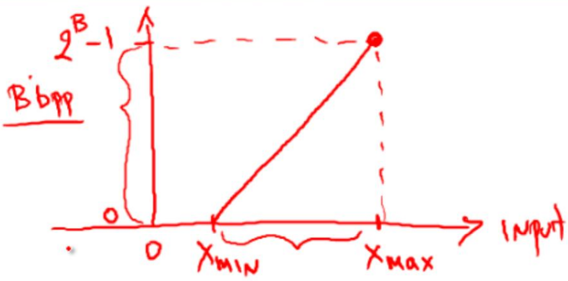




$\gamma = 2$

# Contrast Stretching

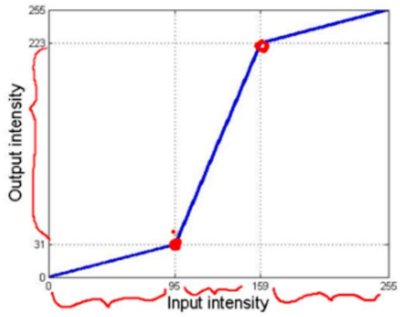




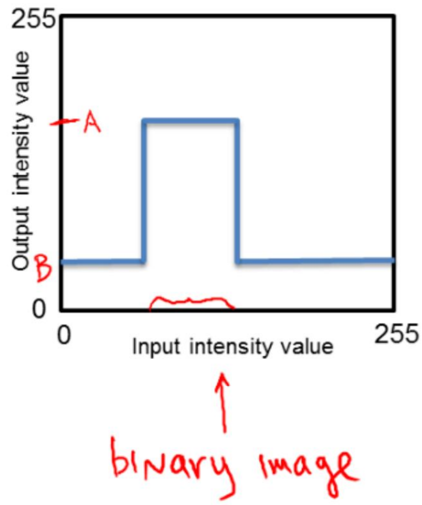

$$y(\mathbf{n}) = \frac{2^B - 1}{x_{max} - x_{min}}(x(\mathbf{n}) - x_{min})$$

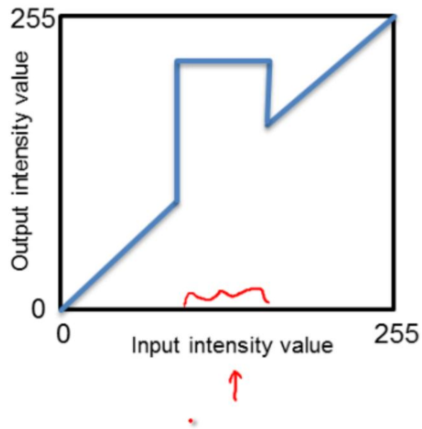
$$x_{max} = \max_{\mathbf{n}} x(\mathbf{n}) \quad x_{min} = \min_{\mathbf{n}} x(\mathbf{n})$$

# Dynamic Range Expansion

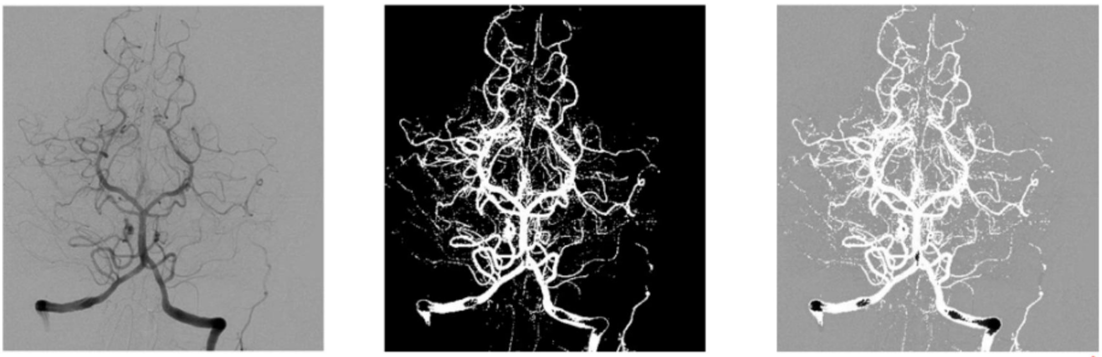



# Intensity-Level Slicing

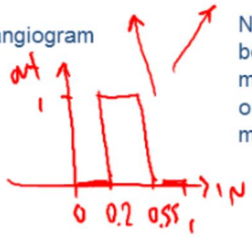


# Intensity-Level Slicing

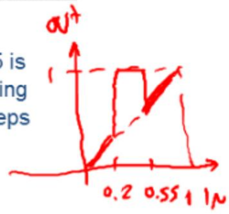


Original angiogram

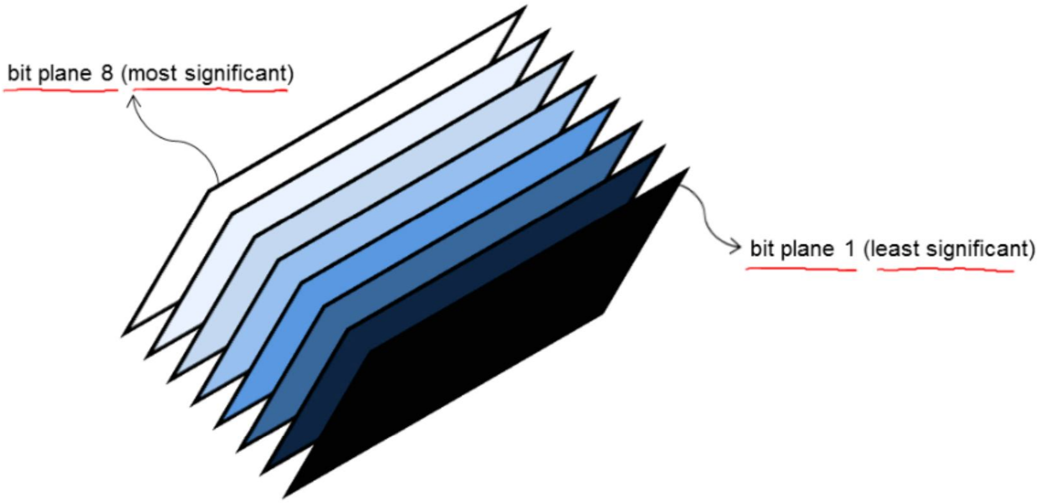



Normalized intensity between 0.2 and 0.55 is mapped to 1, everything outside this range is mapped to 0

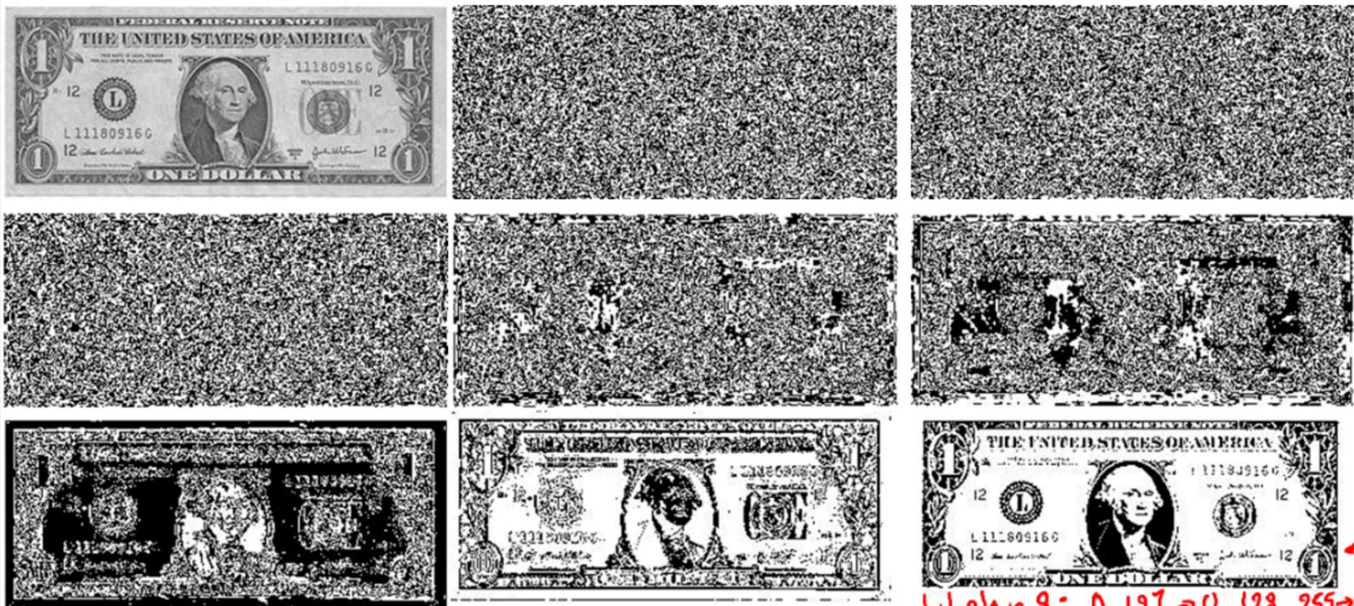
Normalized intensity between 0.2 and 0.55 is mapped to 1, everything outside this range keeps its original value




# Bit-Plane Slicing



# Bit-Plane Slicing Example



bit plane 8 : 0-127→0 128-255→1

# Bit-Plane Slicing Example

reconstruction using bit-planes 8, 7

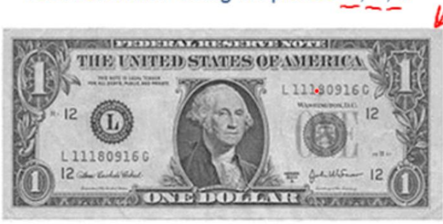
reconstruction using bit-planes 8, 7, 6

reconstruction using bit-planes 8, 7, 6, 5

n-plane : $\times 2^{n-1}$

bit plane 8 : $\times 128$

bit plane 7 : $\times 64$



original

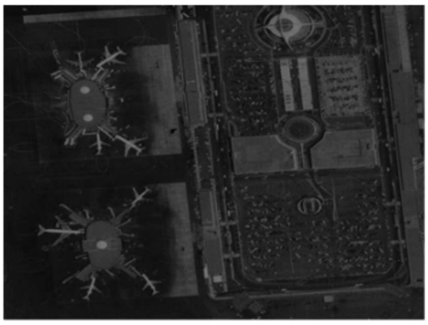
# Enhancement Topics

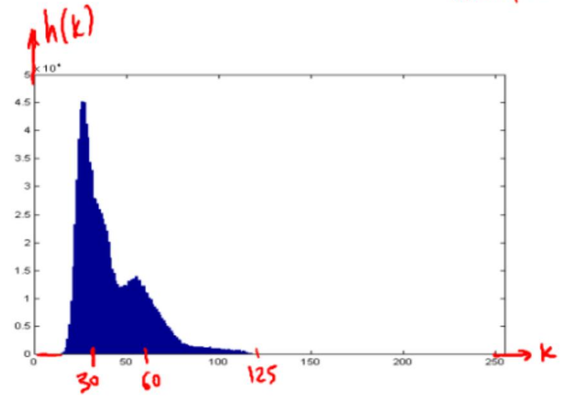
- Point-wise Intensity Transformations
  - Log, Power-law, Piecewise linear
- Histogram Processing
- Spatial Filtering (LSI vs non-linear)
  - Smoothing
  - Sharpening
  - Homomorphic Filtering
- Pseudo-Coloring
- Video Enhancement

# Image Histogram

$h(k) = n_k$, the number of pixels with value $k$, $\quad k = 0, \ldots, N-1$

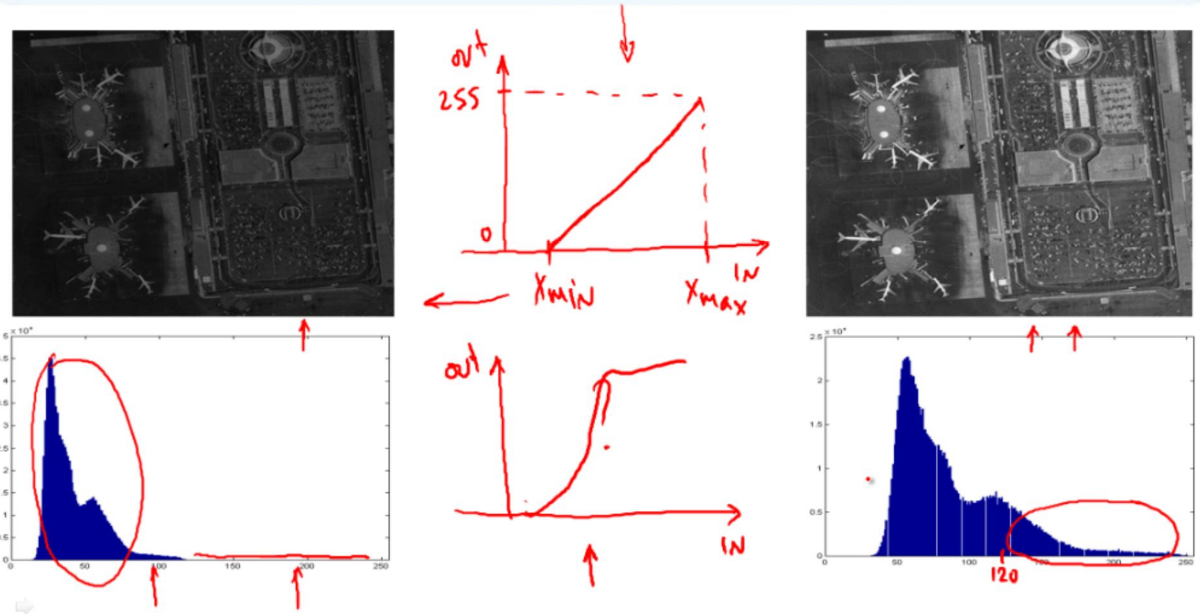
$\frac{h(k)}{\#\text{ of pixels}}$






*Probability density function of intensity values*

# Contrast Stretching




# Histogram Equalization

- Objective: Obtain a **flat** histogram
- Normalized histogram

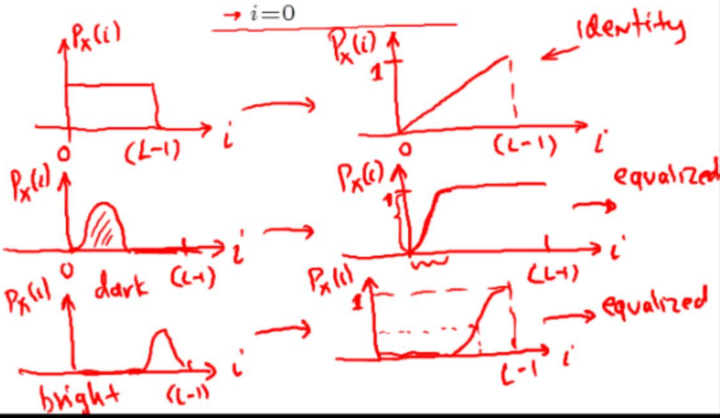
pdf $$p_x(i) = p(x=i) = \frac{n_i}{MN}, \quad 0 \le i \le L-1 \qquad \sum_{i=0}^{L-1} p_x(i) = 1$$

- Procedure:

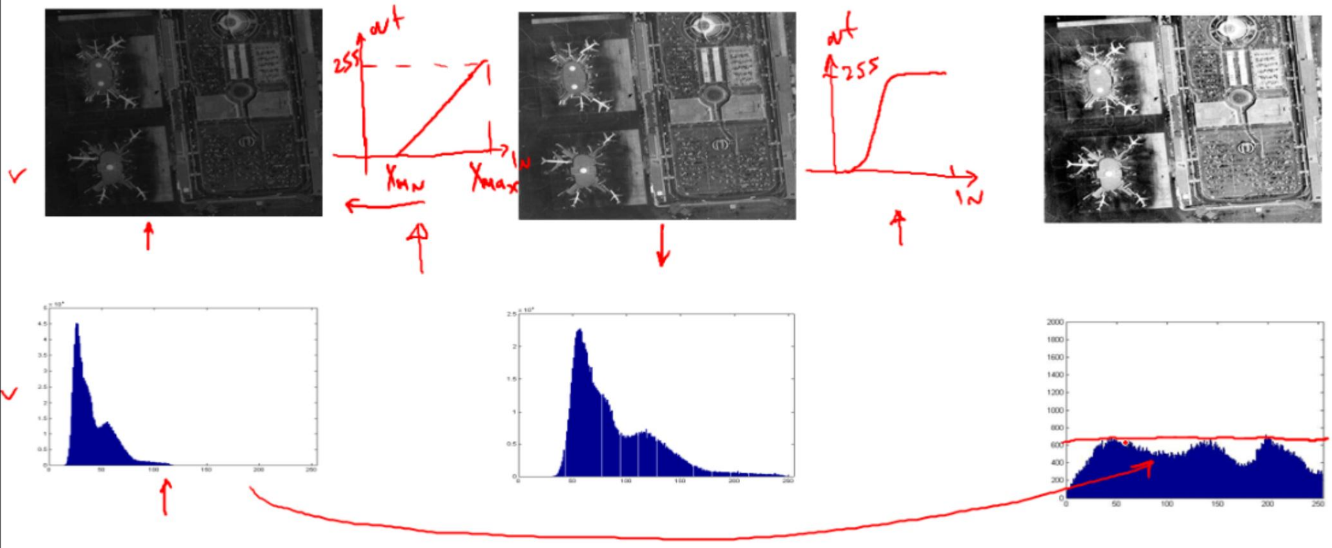
$$P_x(i) = \sum_{j=0}^{i} p_x(j)$$ cdf

$$\tilde{y}(\mathbf{n}) = P_x(x(\mathbf{n}))$$

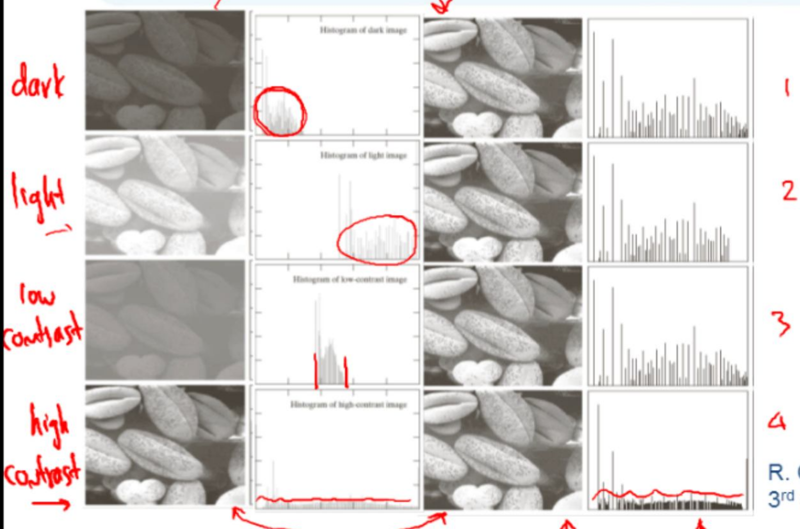
$$y(\mathbf{n}) = (L-1) \cdot \tilde{y}(\mathbf{n})$$



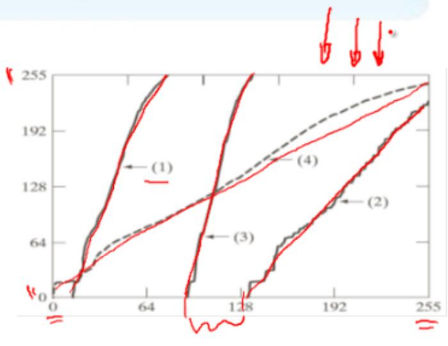
# Contrast Stretching vs Histogram Equalization




# Examples



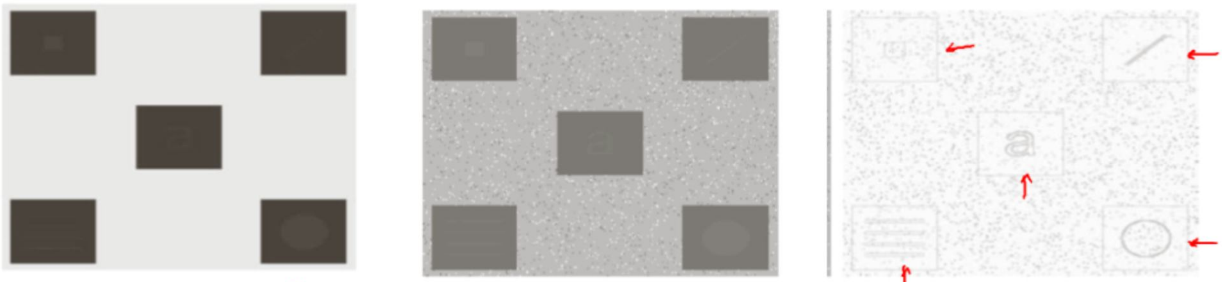

Original | Histogram | Histogram-equalized image | Equalized histogram




*Transformations: 1->top, 4->bottom*

R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, 2008.

# Local Histogram Equalization




*Original image*

*Global histogram equalization*

*Local histogram equalization using a 3x3 neighborhood*

R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed., Prentice Hall, 2008.

# Enhancement Topics

- Point-wise Intensity Transformations
  - Log, Power-law, Piecewise linear
- Histogram Processing
- Spatial Filtering (LSI vs non-linear)
  - Smoothing
  - Sharpening
  - Homomorphic Filtering
- Pseudo-Coloring
- Video Enhancement

# Smoothing Spatial Filters

- **Objective**: Enhance image by reducing noise that may be present.
- Will discuss methods that attempt to reduce additive **broadband** and/or **salt and pepper noise**, also known as impulsive noise.
- Will discuss additional techniques under recovery

# LSI Systems

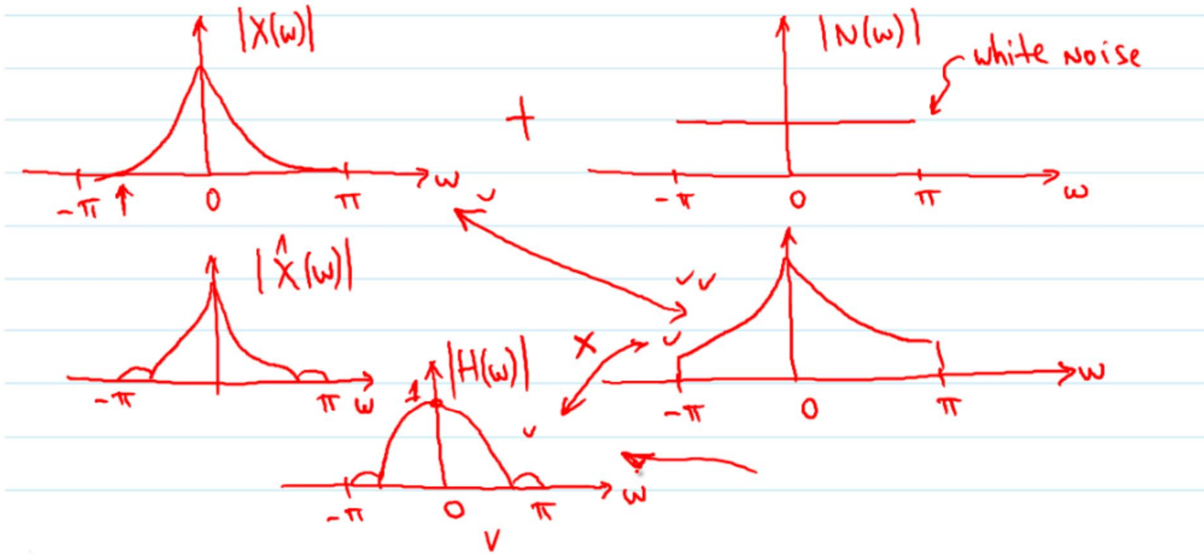
$$y(n_1, n_2) = x(n_1, n_2) + w(n_1, n_2)$$

$$y(n_1, n_2) \longrightarrow \boxed{h(n_1, n_2)} \longrightarrow \hat{x}(n_1, n_2) = y(n_1, n_2) * * h(n_1, n_2)$$

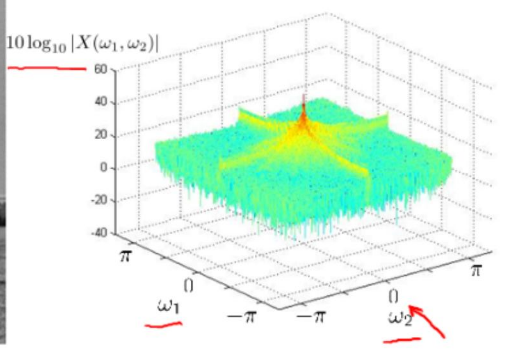
$$Y(k_1, k_2) \longrightarrow \boxed{H(k_1, k_2)} \longrightarrow \hat{X}(k_1, k_2) = Y(k_1, k_2) \cdot H(k_1, k_2)$$

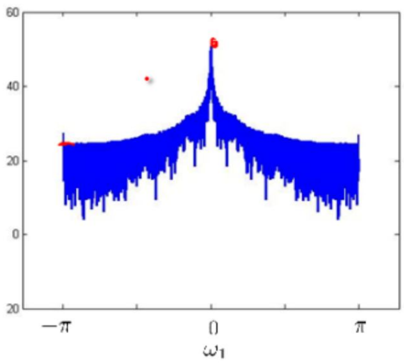
$$\hat{x}(n_1, n_2) = y(n_1, n_2) \circledast \circledast h(n_1 n_2)$$

# Noise Smoothing in the Frequency Domain

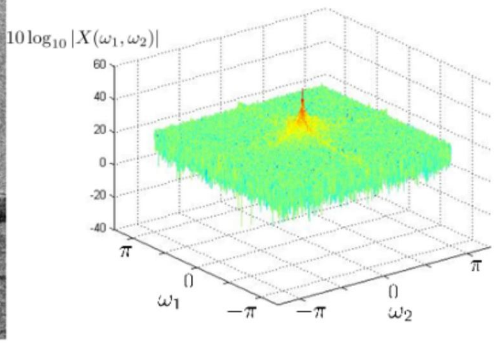


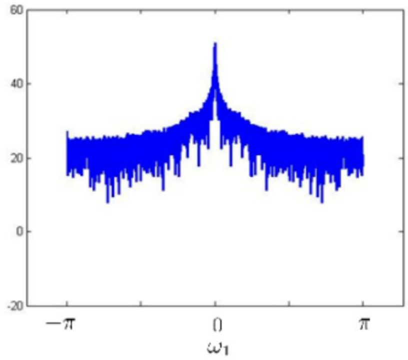

# Original Image



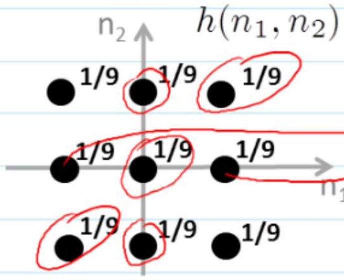
# Noisy Image




# Frequency Response of Flat Filters

3x3



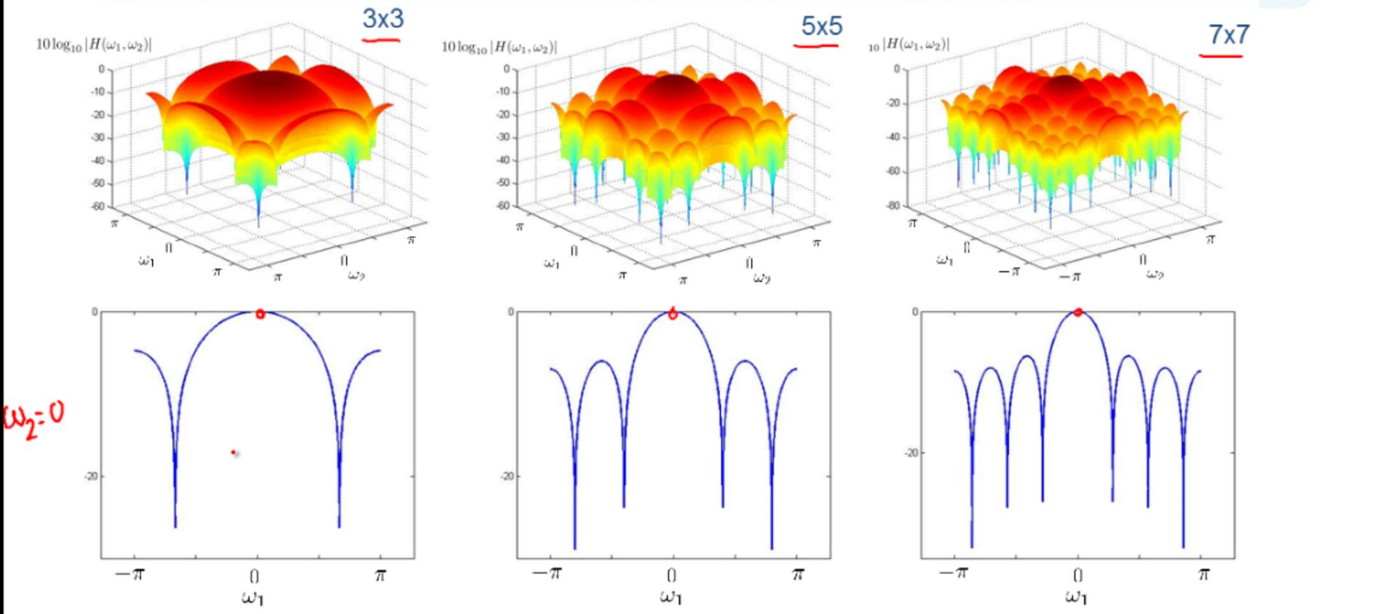

$$H(\omega_1,\omega_2) = \frac{1}{9}$$

$$+\frac{2}{9}cos(\omega_1 n_1) + \frac{2}{9}cos(\omega_2 n_2)$$

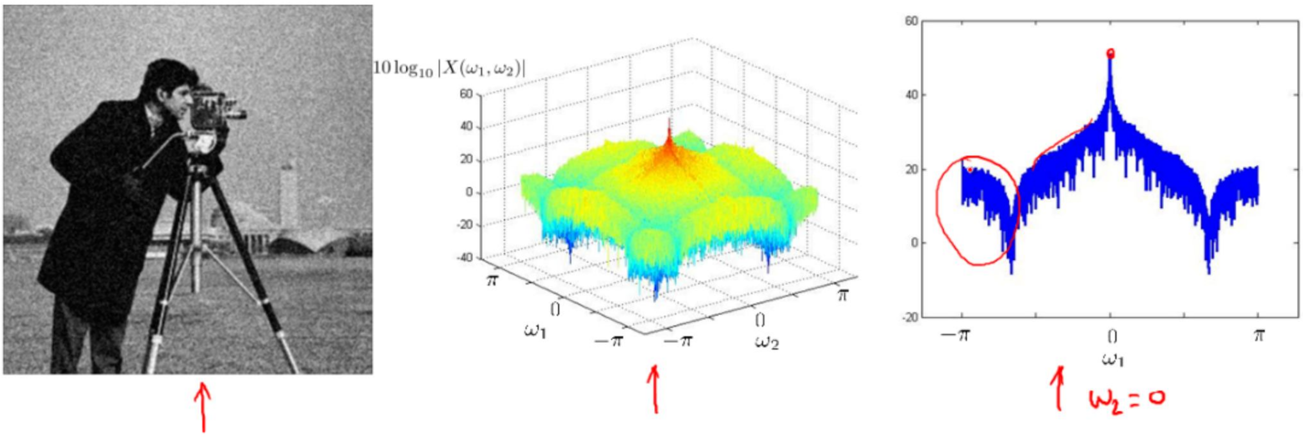
$$+\frac{2}{9}cos(\omega_1 n_1 + \omega_2 n_2) + \frac{2}{9}cos(\omega_1 n_1 - \omega_2 n_2)$$

$H(0,0) = 1$

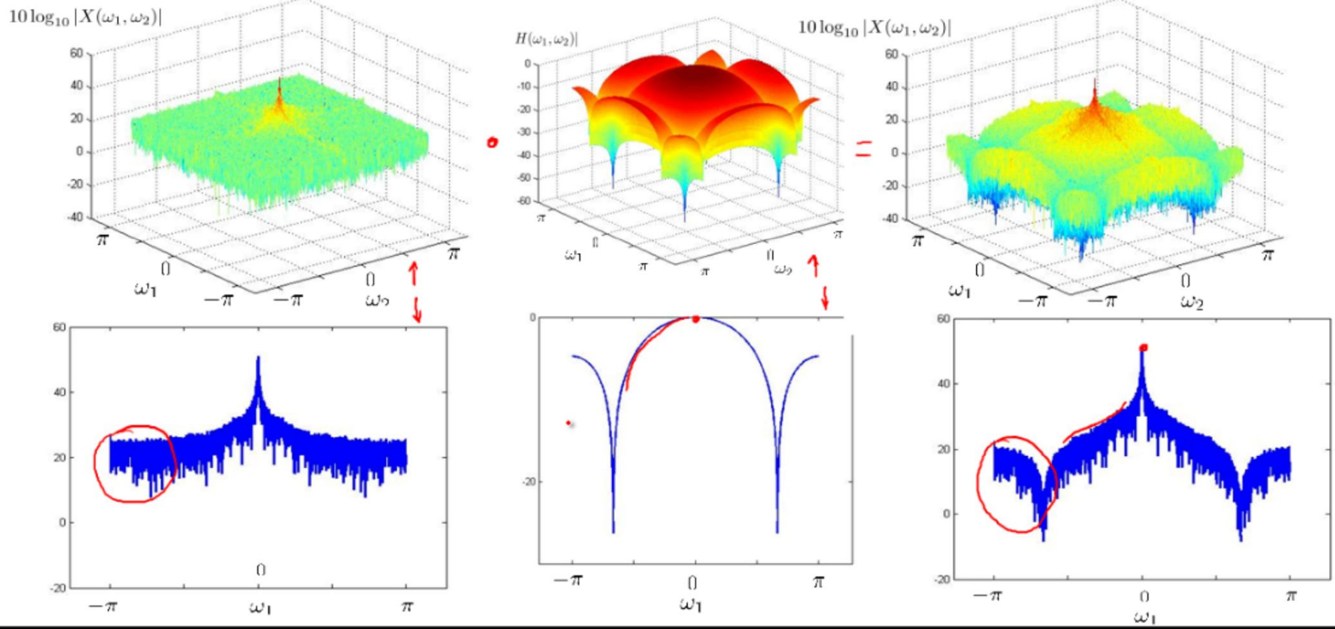
# Frequency Response of Flat Filters



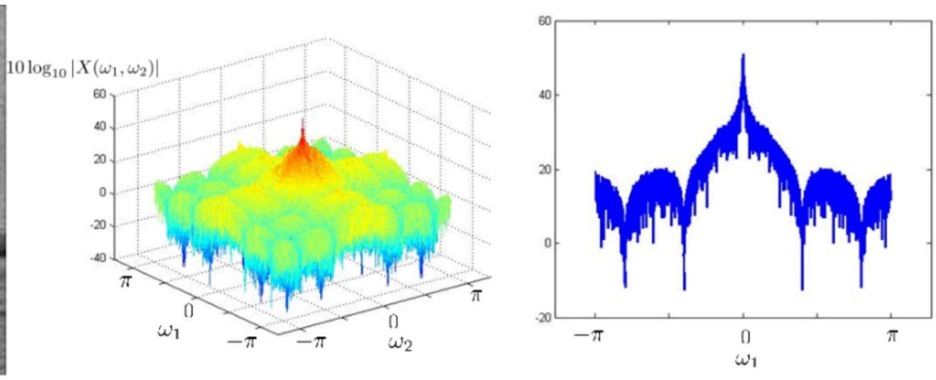

# Filtered by 3x3 Filter



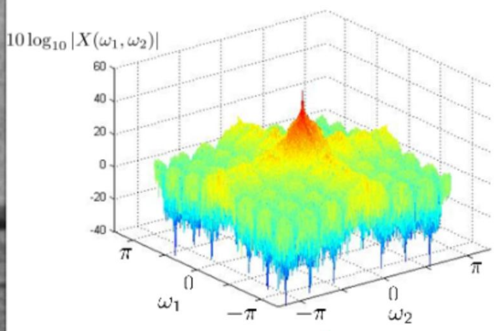
# Frequency Domain Filtering

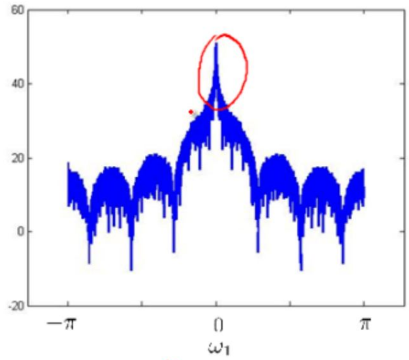



# Filtered by 5x5 Filter

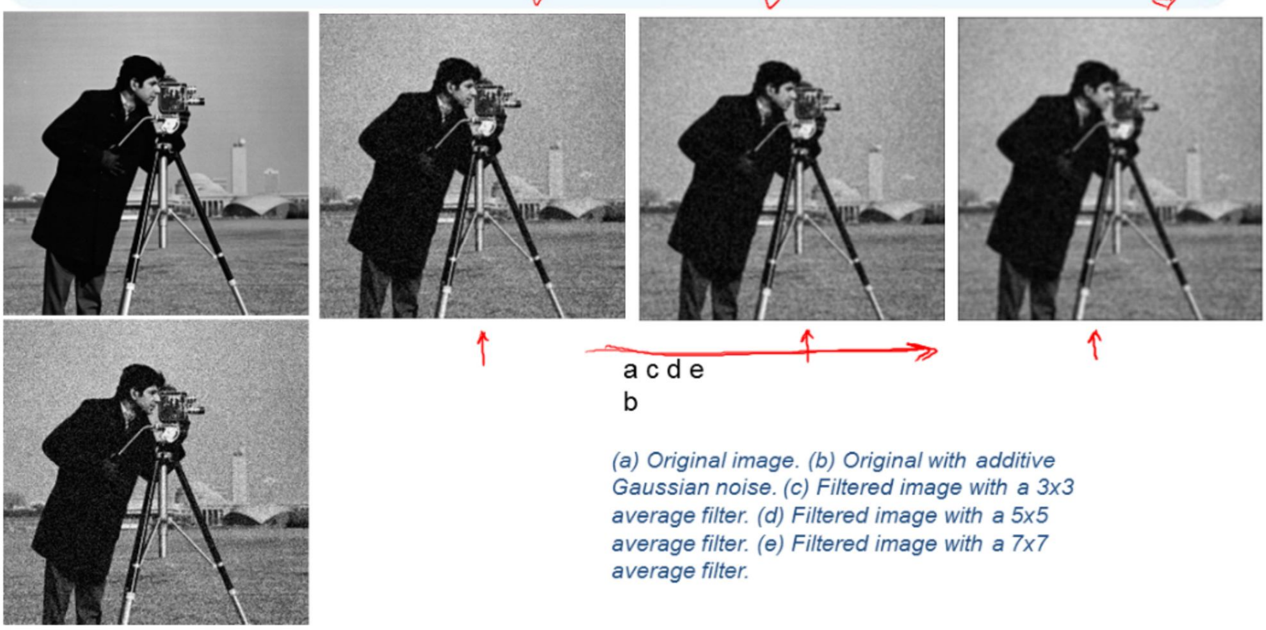


# Filtered by 7x7 Filter




# Noise Smoothing Example



a c d e
b

*(a) Original image. (b) Original with additive Gaussian noise. (c) Filtered image with a 3x3 average filter. (d) Filtered image with a 5x5 average filter. (e) Filtered image with a 7x7 average filter.*

# Spatially Adaptive Noise Smoothing

noise variance

mean, average, flat filter

$$y(n_1, n_2) = (1 - \frac{\sigma_n^2}{\sigma_{x_l}^2}) x(n_1, n_2) + \frac{\sigma_n^2}{\sigma_{x_l}^2} \bar{x}(n_1, n_2)$$

local image variance

$$\sigma_{x_l}^2 = \sum_{(n_1,n_2)\in N}\sum (x(n_1,n_2) - \bar{x}(n_1,n_2))^2$$

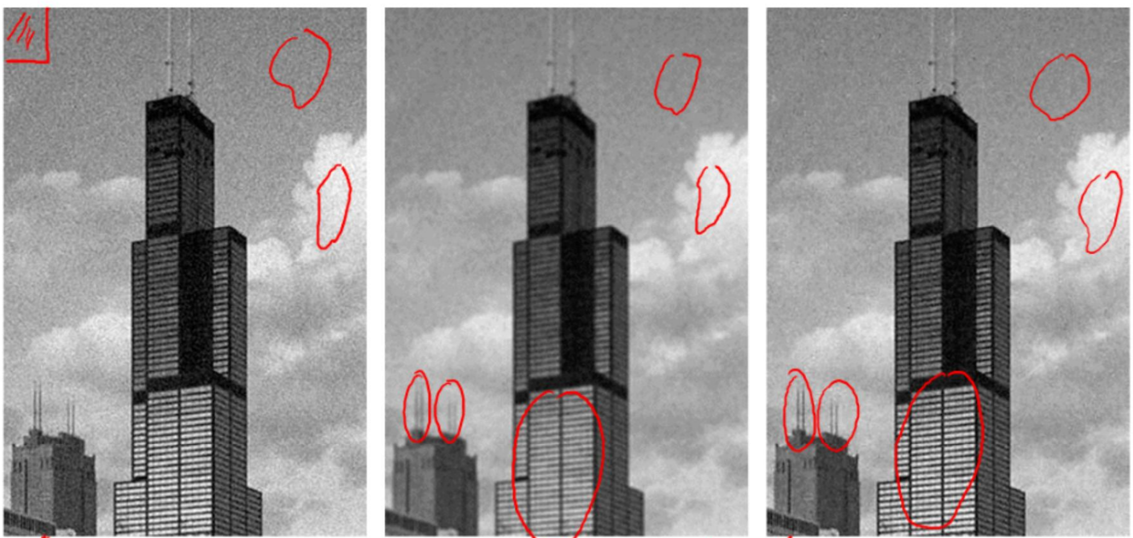
$$\bar{x}(n_1,n_2) = \frac{1}{|N|} \sum_{(n_1,n_2)\in N}\sum x(n_1,n_2)$$ ← flat

$3 \times 3 \quad |N| = 9$

edges: $\sigma_{x_l}^2 >> \sigma_n^2 \to \frac{\sigma_n^2}{\sigma_{x_l}^2} \to 0 \Rightarrow y(n_1,n_2) = x(n_1,n_2)$

flat area: $\sigma_{x_l}^2 < \sigma_n^2 \to \frac{\sigma_n^2}{\sigma_{x_l}^2} \approx 1 \Rightarrow y(n_1,n_2) = \bar{x}(n_1,n_2)$

# Adaptive Spatial Filtering Example




Noisy image

Filtered image by a 3x3 flat filter

Filtered image by a 3x3 spatially adaptive flat filter

# Median Filtering

$$y(n_1, n_2) = \text{median}\{x(n_1 - m_1, n_2 - m_2),\ \ (m_1, m_2) \in \mathcal{N}\}$$

**Median** obtained by sorting all pixels in the analysis window in increasing or decreasing order of amplitudes and picking the middle value if the number of pixels is **odd**, or the average of the two values in middle if the number of pixels is **even** .

$x(n_1,n_2) = a \quad P_a \quad$ bipolar → salt-&-pepper
$x(n_1,n_2) = b \quad P_b \quad$ unipolar → salt, → pepper

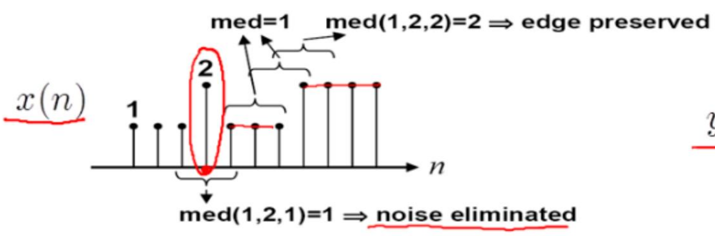
Properties of median filters

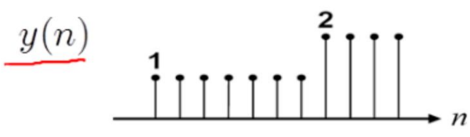
1. **Non-linear** data-drop-out / spike
2. **Good** for reducing **impulsive** (salt and pepper) noise and removing isolated lines and pixels while preserving spatial resolution (reduced blurring) and edges.
3. **Poor** performance in case of **broadband noise**.
4. **1D median filters** have better performance in preserving edges (discontinuities) than 2D median filters.

# 1D Median Filtering Example

3-point median filter

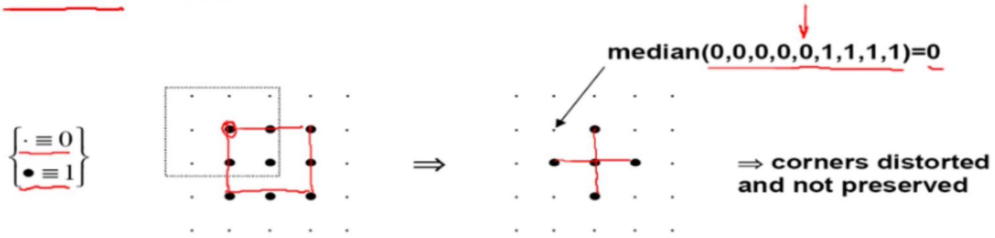
$$y(n) = \text{median}\{x(n-1), x(n), x(n+1)\}$$



# 2D Median Filtering Example

- 2D 3x3 median filter



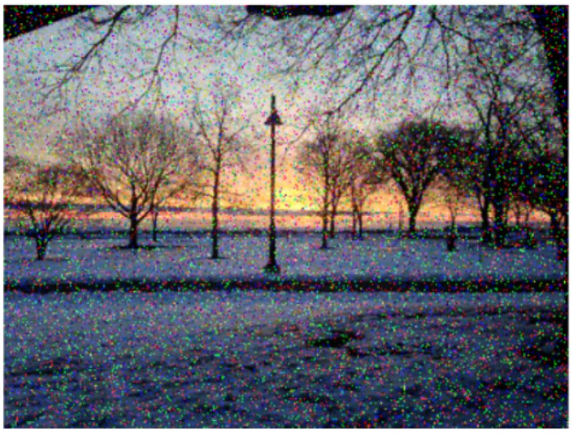

- Since 1D median filter tends to preserve edges of 1D sequences it is preferable to use separable median filtering.
  - Filter image with a vertical 1D median
  - Filter the resulting image with a horizontal 1D median

# 2D Median Filtering Example

a b
c d

*(a) Original image. (b) Original with additive salt-and-pepper noise. (c) Filtered image with a 3x3 median filter. (d) Filtered image with a 5x5 median filter.*

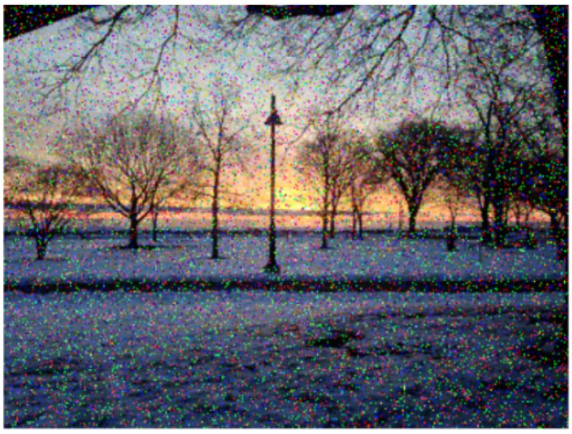
# Noise Reduction Example



Noisy Image

Gaussian 5x5 Filtering

# Noise Reduction Example



Noisy Image

3x3 Median Filtering

# Noise Reduction Example

Noisy Image

Gaussian 5x5 Filtering

# Noise Reduction Example

Noisy Image

3x3 Median Filtering (pass #1)

# Noise Reduction Example

Noisy Image

3x3 Median Filtering (pass #2)

# Noise Reduction Example

Noisy Image

3x3 Median Filtering (pass #3)

# Order Statistic Filters

*Median filter* $y(n_1, n_2) = \text{median}_{(m_1,m_2)\in\mathcal{N}}\{x(m_1, m_2)\}$

$\mathcal{N}$ neighborhood centered at $(n_1, n_2)$

*Max filter* $y(n_1, n_2) = \max\limits_{(m_1,m_2)\in\mathcal{N}}\{x(m_1, m_2)\}$ pepper - noise

*Min filter* $y(n_1, n_2) = \min\limits_{(m_1,m_2)\in\mathcal{N}}\{x(m_1, m_2)\}$ salt - noise

*Mid-point filter* $y(n_1, n_2) = \frac{1}{2}\left(\max\limits_{(m_1,m_2)\in\mathcal{N}}\{x(m_1, m_2)\} + \min\limits_{(m_1,m_2)\in\mathcal{N}}\{x(m_1, m_2)\}\right)$

*Alpha-trimmed mean filter* $y(n_1, n_2) = \frac{1}{|\mathcal{N}| - \alpha}\sum\limits_{(m_1,m_2)\in\mathcal{N}} x_r(m_1, m_2)$

$\alpha = 0$ : arithmetic mean
$\alpha = |N|-1$ : median

where $x_r(m_1, m_2)$ are the remaining values after we delete the $\frac{\alpha}{2}$ lowest and highest values

# Mean Filters

*Arithmetic mean filter* $y(n_1, n_2) = \frac{1}{|\mathcal{N}|}\sum\limits_{(m_1,m_2)\in\mathcal{N}} x(m_1, m_2)$

$\mathcal{N}$ neighborhood centered at $(n_1, n_2)$

*Geometric mean filter* $y(n_1, n_2) = \left[\prod\limits_{(m_1,m_2)\in\mathcal{N}} x(m_1, m_2)\right]^{\frac{1}{|\mathcal{N}|}}$

*Harmonic mean filter* $y(n_1, n_2) = |\mathcal{N}|\left[\sum\limits_{(m_1,m_2)\in\mathcal{N}} \frac{1}{x(m_1, m_2)}\right]^{-1}$

Gaussian
salt
Not pepper

*Contra-harmonic mean filter* Q order of filter

Q>0 eliminates pepper
Q<0 " salt

Q=0 : arithmetic mean
Q=-1 : harmonic mean

$y(n_1, n_2) = \left[\sum\limits_{(m_1,m_2)\in\mathcal{N}} x(m_1, m_2)^{Q+1}\right]\left[\sum\limits_{(m_1,m_2)\in\mathcal{N}} x(m_1, m_2)^{Q}\right]^{-1}$

# Filtering Examples

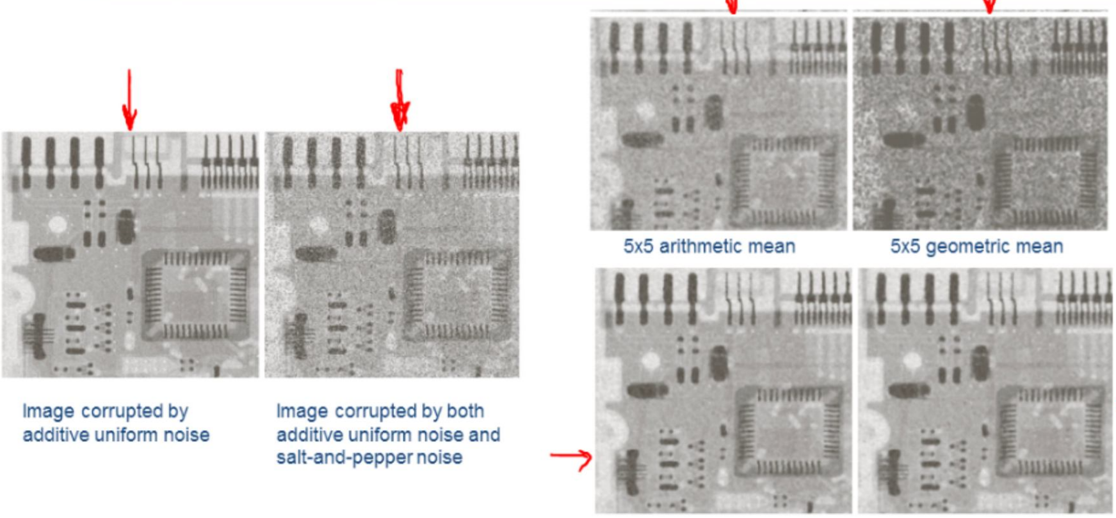


Image corrupted by additive uniform noise

Image corrupted by both additive uniform noise and salt-and-pepper noise

5x5 arithmetic mean

5x5 geometric mean

5x5 median

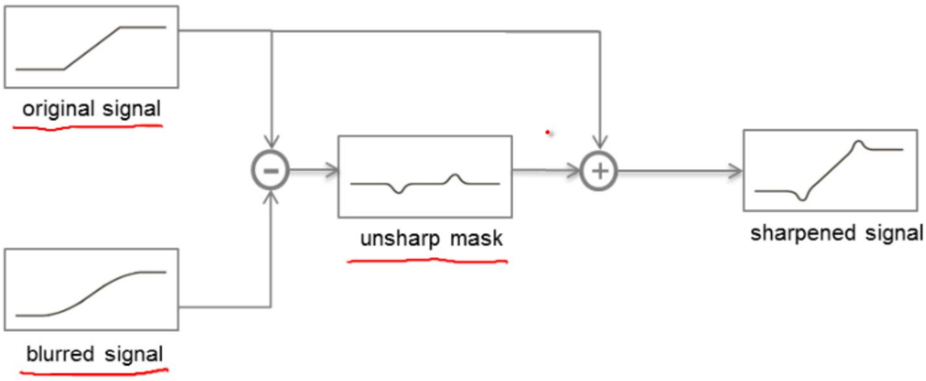
5x5 alpha-trimmed mean alpha=5

R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition, Pearson Prentice Hall,2008.

# Enhancement Topics

- Point-wise Intensity Transformations
  - Log, Power-law, Piecewise linear
- Histogram Processing
- Spatial Filtering (LSI vs non-linear)
  - Smoothing
  - Sharpening
  - Homomorphic Filtering
- Pseudo-Coloring
- Video Enhancement

# Unsharp Masking and Highboost Filtering




Unsharp version of signal

# Unsharp Masking and Highboost Filtering

$$y_{\text{mask}}(n_1, n_2) = x(n_1, n_2) - x_L(n_1, n_2)$$

*Unsharp mask*

$$y(n_1, n_2) = x(n_1, n_2) + a y_{\text{mask}}(n_1, n_2) = x_L(n_1, n_2) + (1 + a) x_H(n_1, n_2)$$

$X(n_1, n_2) = X_L(n_1, n_2) + X_H(n_1, n_2)$

$a > 1$ highboost filtering
$a < 1$ de-emphasizes mask

$a = 1$

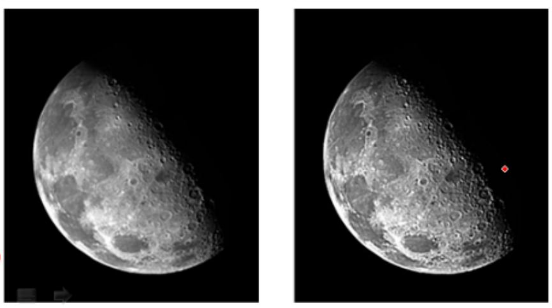
# Unsharp Masking

a b
c d e

a: 216x308 original
b: smoothed by 7x7 Gaussian
c: unsharp mask
d: result with alpha = 1
e: result with alpha = 2

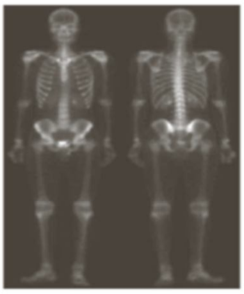
# Sharpening Spatial Filters

a: original
b: Laplacian (clipped to 0)
c: Laplacian (rescaled)
d: result with [0 1 0; 1 -4 1; 0 1 0]
e: result with [1 1 1; 1 -8 1; 1 1 1]

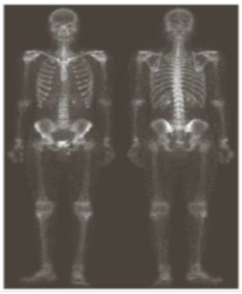
a b c
d e

# Combining Spatial Enhancement Methods

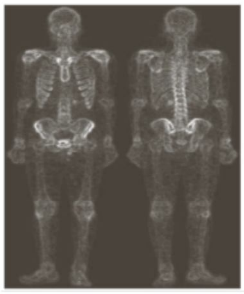


(a): Whole body bone scan

(b): Laplacian of (a)



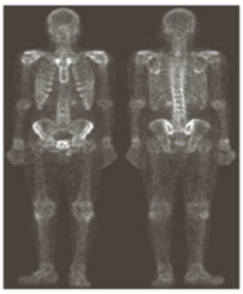
(c): (a)+(b)



(d) Sobel gradient of (a)

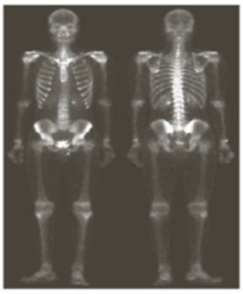
R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition, Pearson Prentice Hall,2008.

# Combining Spatial Enhancement Methods

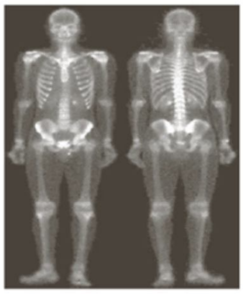


(e) Smoothed Sobel gradient
With a 5x5 average

(f) Product of sharp image (c)
and smoothed Sobel
gradient (e)



(g) Sharp image obtained
By sum of original (a) and (f)



(h) Power law transformation
of (g)

R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition, Pearson Prentice Hall,2008.

# Enhancement Topics

- Point-wise Intensity Transformations
  - Log, Power-law, Piecewise linear
- Histogram Processing
- Spatial Filtering (LSI vs non-linear)
  - Smoothing
  - Sharpening
  - Homomorphic Filtering
- Pseudo-Coloring
- Video Enhancement

# Motivation

- **Image** with a **large dynamic range**, e.g., a natural scene on a bright sunny day, recorded on a **medium** with a **small dynamic range**, results in image contrast significantly reduced especially in the dark and bright regions
- **Objective**: **Reduce** dynamic range & increase local contrast **prior** to recording it on medium with a small dynamic range

# Image Model

$$x(n_1, n_2) = i(n_1, n_2) \cdot r(n_1, n_2)$$

*Image formed by recording light reflected from objects which are illuminated by some light source*

***Illumination:** assumed to be slowly varying and main contributor to dynamic range*

***Reflectance:** represents details of objects and assumed to vary rapidly and the primary contributor to local contrast*

- To decrease dynamic range, decrease $i(n_1, n_2)$
- To increase local contrast, increase $r(n_1, n_2)$

# Homomorphic Processing

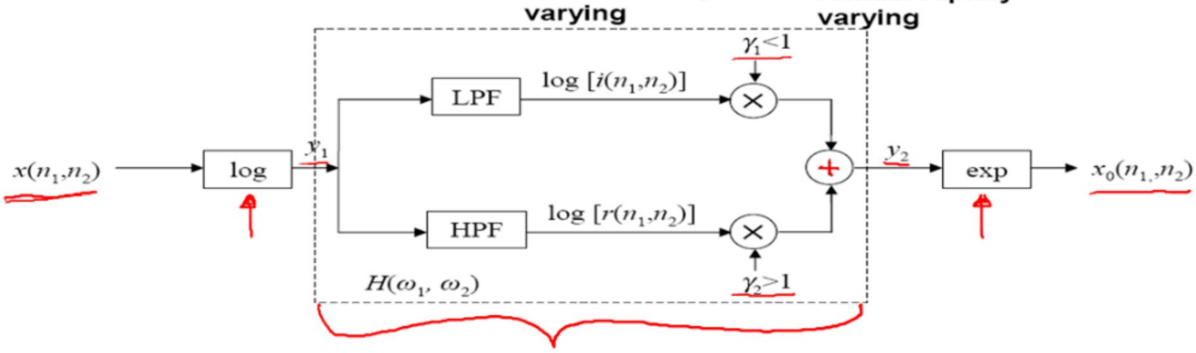
low frequency usually

high/mid frequency

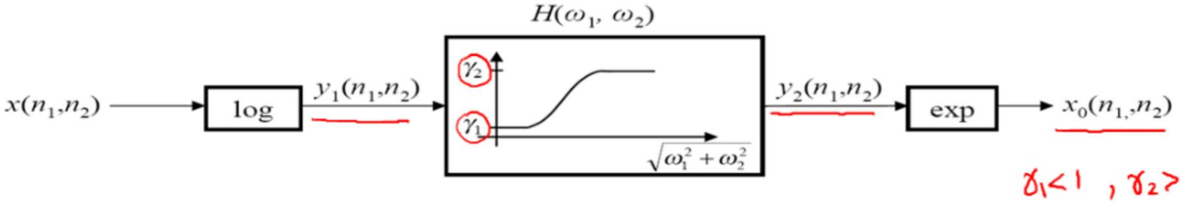
$$\underbrace{\log x(n_1,n_2)}_{y_1(n_1,n_2)} = \log\,[i(n_1,n_2)] + \log\,[r(n_1,n_2)]$$

assumed to remain slowly varying

assumed to remain rapidly varying



# Homomorphic Processing




$\gamma_1 < 1$ , $\gamma_2 > 1$

$$y_2(n_1,n_2) \approx \gamma_1 \log[i(n_1,n_2)] + \gamma_2 \log[r(n_1,n_2)] \Rightarrow x_0(n_1,n_2) = [i(n_1,n_2)]^{\gamma_1}\,[r(n_1,n_2)]^{\gamma_2}$$

- Filtering in the log intensity domain
- **Note:** Log intensity domain also considered as a perceptual domain since image intensities appear to be modified at the peripheral level of HVS by some form of nonlinearity such as a logarithmic operation.

# Homomorphic Processing Example

$\gamma_1 = 1$

$\gamma_2 = 2$

## Enhancement Topics

- Point-wise Intensity Transformations
  - Log, Power-law, Piecewise linear
- Histogram Processing
- Spatial Filtering (LSI vs non-linear)
  - Smoothing
  - Sharpening
  - Homomorphic Filtering
- Pseudo-Coloring
- Video Enhancement

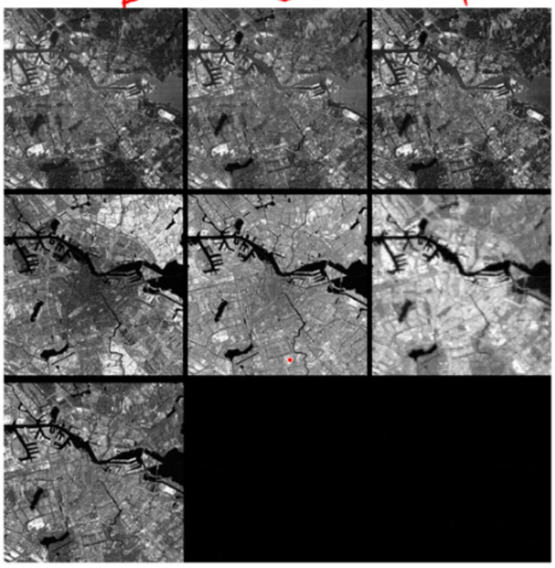
## Pseudo-Coloring

Reason:

1. Place normal objects in “strange” color for attention
2. Color a normal scene to match the color sensitivity of a human viewer
3. Exploit contrast sensitivity
4. Produce a natural color representation of a set of multi-spectral images of a scene
5. Colorization of black and white images and videos

# LANDSAT Multi-Spectral Images

B G R



ESA Landsat

# LANDSAT Multi-Spectral Images

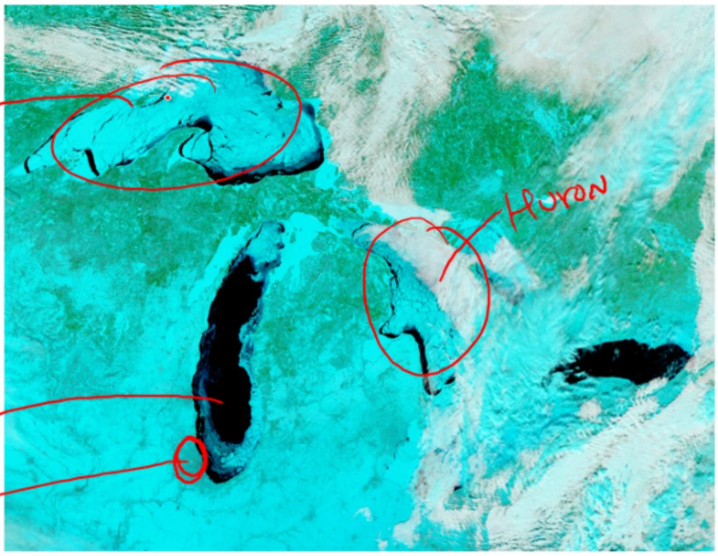
Bands 1-2-3

Bands 4-3-2

Bands 7-4-2

USGS/NASA Landsat

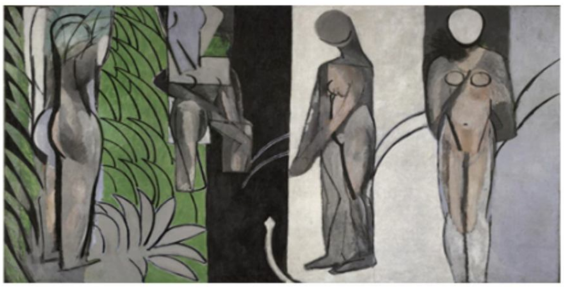
# Pseudo-coloring




SWIR
NIR
R

NASA/Jeff Schmaltz, MODIS Land Rapid Response Team, NASA GSFC

# Colorization



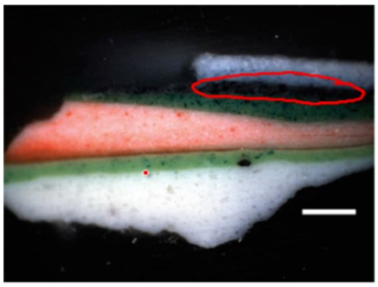
Henri Matisse, *Bathers by a River*

**March–May 1909, fall 1909–spring 1910, May–November 1913, early spring–November 1916, January–October (?) 1917.**

Druet, Nov. 1913

**The Art Institute of Chicago, Charles H. and Mary F. S. Worcester Collection, 1953.158. ©2010 Succession H. Matisse/ Artists Rights Society (ARS), New York.)**

# Scribbles

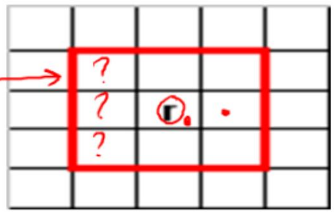


An example of a microscopic cross section at 200x original magnification taken from the white head of the standing figure at farthest right. The dark gray layer that is present between the darker green and light blue relates to the November 1913 campaign documented in Druet's photograph. The thick white layer at the bottom is the ground, or preparation layer, applied on top of the canvas support.

# Basic Principle of the Algorithm

- Changes in color correspond in general to changes in intensity
- Create a weighing scheme: assign a pixel **chrominance** based on **chrominance** of **nearby** pixels with **similar intensity Y**
- **Minimize** (in YUV color space)

$$J(U) = \sum_{\mathbf{r}} \Big( U(\mathbf{r}) - \sum_{\mathbf{s} \in N(\mathbf{r})} w_{\mathbf{rs}} U(\mathbf{s}) \Big)^2$$




$$w_{\mathbf{rs}} \propto e^{-(Y(\mathbf{r})-Y(\mathbf{s}))^2 / 2\sigma_{\mathbf{r}}^2}$$

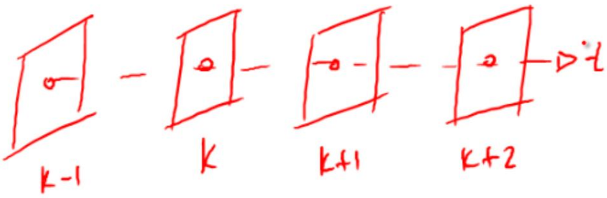
Ax=b

# Druet Colorized

Druet, Nov. 1913

Colorized Druet

# Video Enhancement

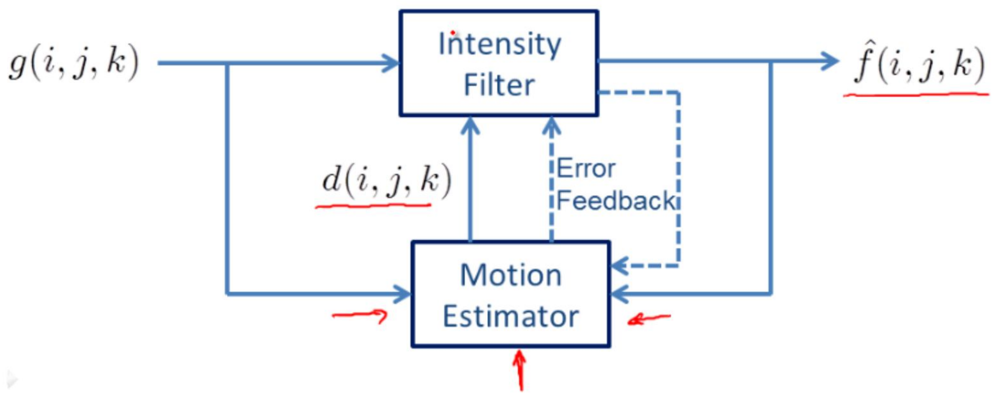



- 2D vs 3D
- Noise Filtering
  - 3D filtering (no ME, motion-dependent, explicit ME)
- Motion Picture/Animation Video Artifact removal

# Noise Filtering

- Degradation Model

$$g(i,j,k) = f(i,j,k) + n(i,j,k)$$




# Non-MC Spatio-temporal Filtering

FIR:

$$\hat{f}(i,j,k) = \sum_{(p,q,l)\in S} w(p,q,l)\,g(i-p,j-q,k-l)$$

- Average filter
- 3D Wiener
- $w(p,q,l)$ is a function of the spatio-temporal edge

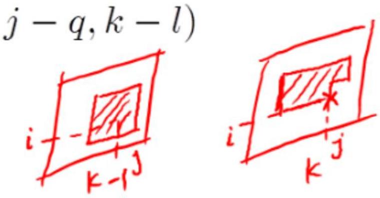
Brailean, J. C., R. P. Kleihorst, S. Efstratiadis, A. K. Katsaggelos, and R. L. Lagendijk, "Noise Reduction Filters for Dynamic Image Sequences: A Review", *IEEE Proceedings,* vol. 83, issue 9, pp. 1272-1292, Sept. 1995.

# Non-MC Spatio-temporal Filtering

IIR:

$$\hat{f}(i,j,k) = [1 - \alpha(i,j,k)]\hat{f}_b(i,j,k) + \alpha(i,j,k)g(i,j,k)$$

$$\hat{f}_b(i,j,k) = \sum_{(p,q,l)\in\mathcal{S}} \gamma(p,q,l)\hat{f}(i-p, j-q, k-l)$$



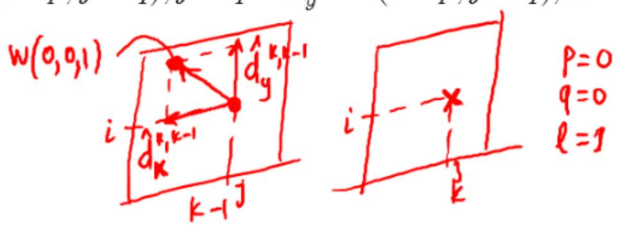

- 3D Kalman Filter
- Order statistics based filters

$$\hat{f} = MEDIAN[max(m_1, m_2, m_3, m_4, m_5), g, min(m_1, m_2, m_3, m_4, m_5)]$$

# MC Spatio-temporal Filtering

FIR

$$\hat{f}(i,j,k) = \sum_{(p,q,l)\in\mathcal{S}} w(p,q,l)g(i-p-\hat{d}_x^{k,k-l}(i-p,j-q), j-q-\hat{d}_y^{k,k-l}(i-p,j-q), k-l)$$



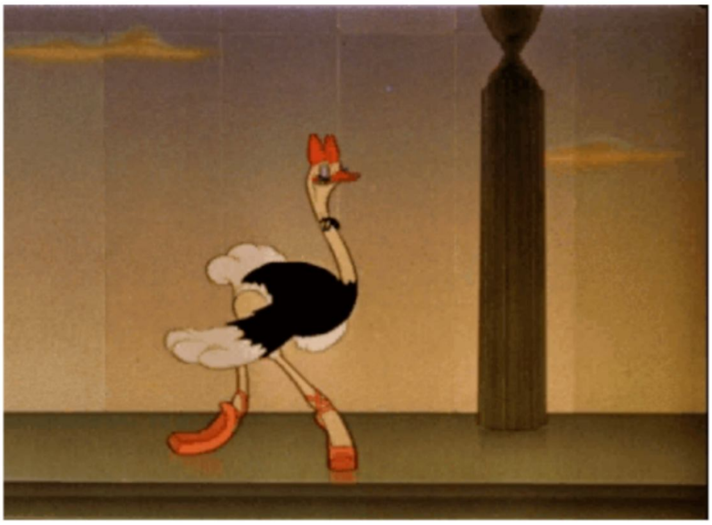

IIR

$$\hat{f}(i,j,k) = [1 - \alpha(i,j,k)]\hat{f}_b(i,j,k) + \alpha(i,j,k)g(i,j,k)$$

with

$$\hat{f}_b(i,j,k) = \sum_{(p,q,l)\in\mathcal{S}} \gamma(p,q,l)\hat{f}(i-p-\hat{d}_x^{k,k-l}(i,j), j-q-\hat{d}_y^{k,k-l}(i,j), k-l)$$

# Degraded Animation Video



Tom, B., M. G. Kang, M. - C. Hong, and A. K. Katsaggelos, "Detection and Removal of Anomalies in Digitized Animation Film", *Int. Journal of Imaging Systems and Technology*, vol. 9, issue 4, pp. 283-293, 1998.

# Enhancement of Motion Pictures

- Blotches (due to dirt, sparkle, film abrasion, scratches and fingerprints on the cellulose surface)
- Intensity flicker (or "boiling effect")
- Vinegar syndrome

## Animation Characteristics

- Large image sizes (Fantasia, 2048x1216 pixels, 36 bits/pixel)
- Large “un-natural” motion
- Scratches, fingerprints, spots
- Duplicated frames and severe illumination changes

## Blotch Detection

- They occur only on one frame
- Compute

$$g_k = \frac{1}{2}(f_{k-1} + f_{k+1})$$

$$d_1 = |g_k - f_k|$$

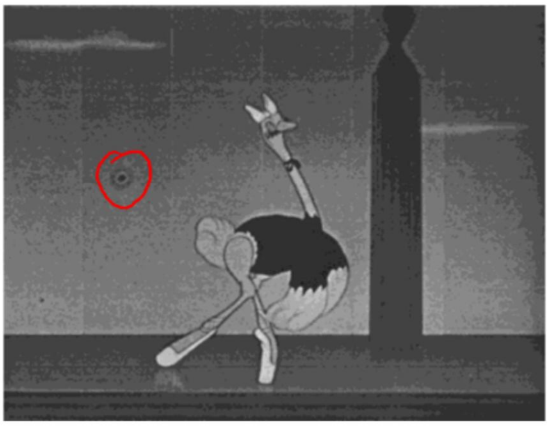
$$d_2 = |f_{k-1} - f_k|$$

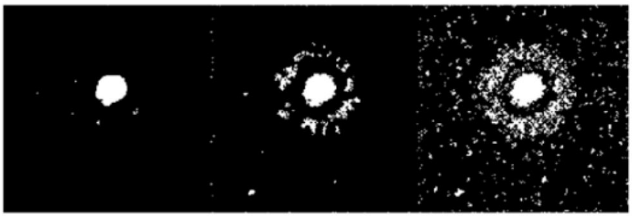
$$d_3 = |f_k - f_{k+1}|$$

$$\tau_i = \mu_{d_i} + \eta_i \sigma_{d_i}, \quad i = 1, 2, 3$$

$$b_i(m,n) = \begin{cases} 1, & \text{if } d_i(m,n) \geq \tau_i \\ 0, & \text{if } d_i(m,n) \leq \tau_i \end{cases}$$

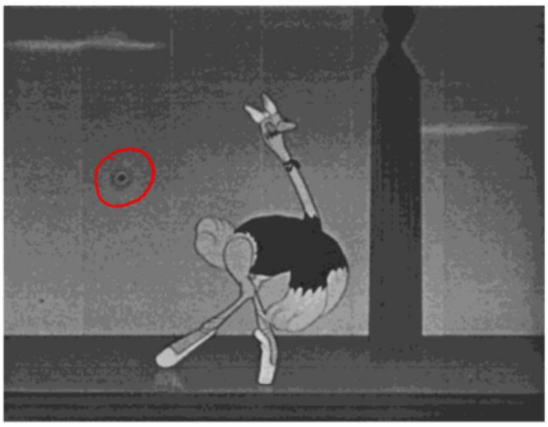
# Detected Blotches

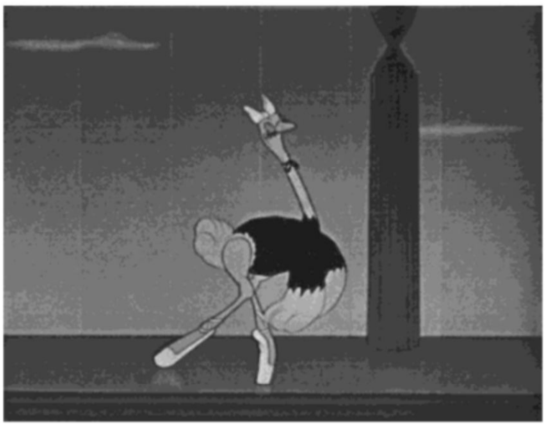




$\eta = 3, 2, 1$

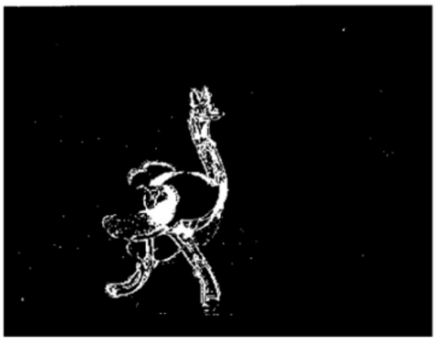
# Blotch Removal



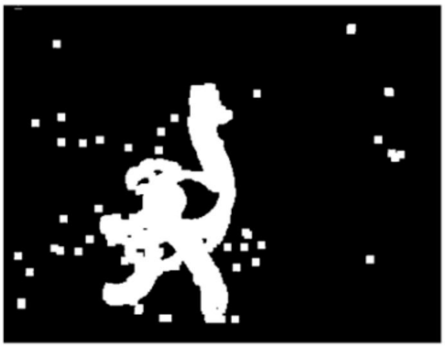


# Motion-Dependent Filtering

$$\hat{f}_k(m,n) = (1-\alpha(mask))f_k(m,n) + \alpha(mask)\mu_k(m,n)$$



mask



expanded mask

NO MOTION $\alpha = 1$ motion $\alpha = 0$

# Enhanced Animation Video

# Degraded Animation Video