Filtering

Image Enhancement
Spatial and Frequency Based

Brent M. Dingle, Ph.D.
2015
Game Design and Development Program
Mathematics, Statistics and Computer Science
University of Wisconsin - Stout

Lecture Objectives

- Previously
  - What a Digital Image is
  - Acquisition of Digital Images
  - Human Perception of Digital Images
  - Digital Representation of Images
  - Various HTML5 and JavaScript Code
    - Pixel manipulation
    - Image Loading
    - Filtering

- Today
  - Image Filtering
  - Image Enhancement
    - Spatial Domain
    - Frequency Domain

Lecture Objectives

- Previously
  - What a Digital Image is
  - Acquisition of Digital Images
  - Human Perception of Digital Images
  - Digital Representation of Images
  - Various HTML5 and JavaScript Code
    - Pixel manipulation
    - Image Loading
    - Filtering

- **Today**
  - **Image Filtering**
  - **Image Enhancement**
    - **Spatial Domain**
    - **Frequency Domain**

Outline

- **Filtering**
  - Introduction
  - Low Pass Filtering
  - High Pass Filtering
  - Directional Filtering
  - Global Filters
    - Normalization
    - Histogram Equalization
- Image Enhancement
  - Spatial Domain
  - Frequency Domain

Filtering

- Filtering
  - modify an image based on image color content without any intentional change in image geometry
    - resulting image essentially has the same size and shape as the original

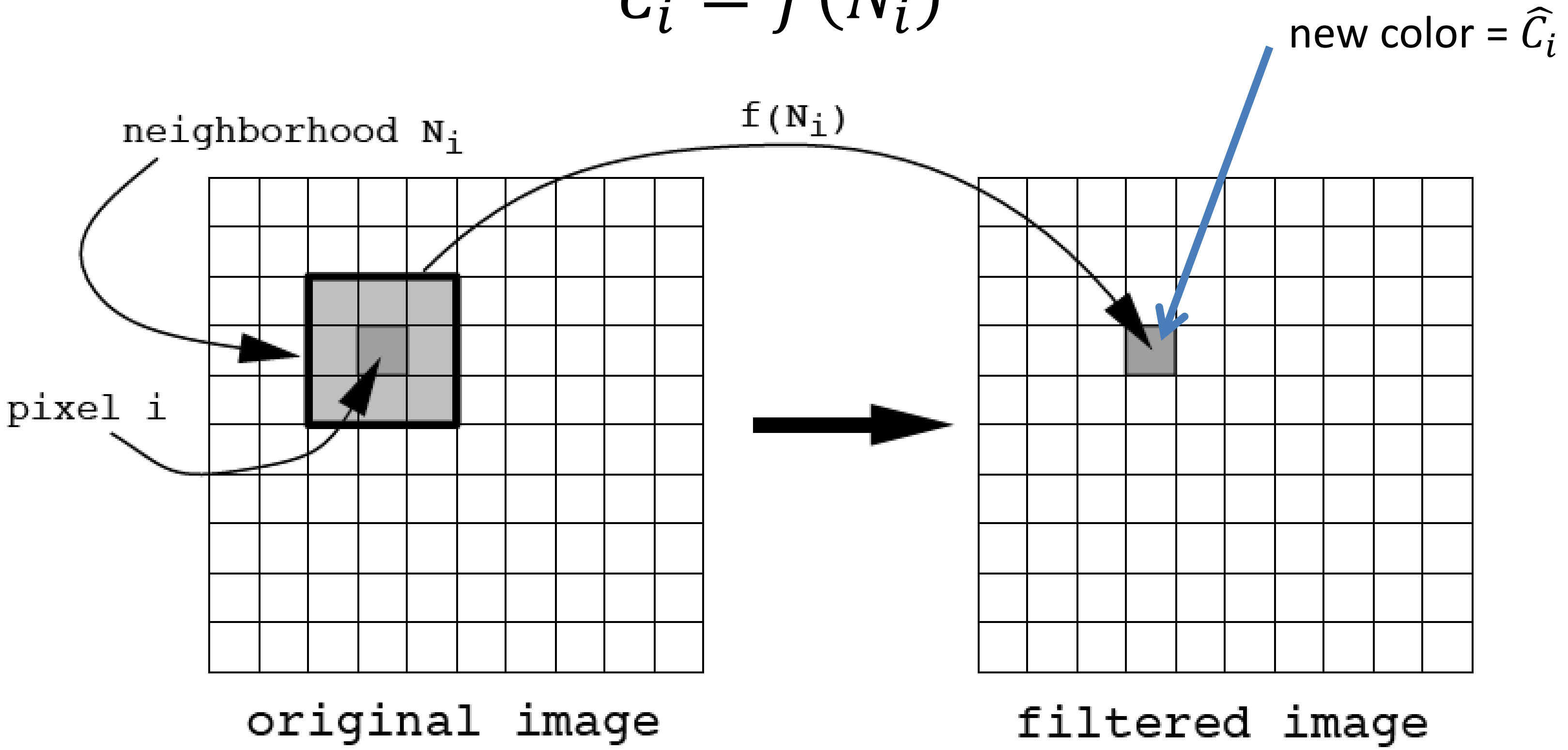
Image Filtering Operation

- Image Filtering Operation
  - Let $\boldsymbol{P_i}$ be the single ***input pixel*** with index *i* and color C_i.
  - *Let* $\widehat{\boldsymbol{P_i}}$ be the corresponding **output pixel** with color $\widehat{\boldsymbol{C}}_{\boldsymbol{i}}$.
  - ***A Filtering Operation***
    - associates each pixel, $\boldsymbol{P_i}$, with a neighborhood set of pixels, $\boldsymbol{N_i}$, and determines an output pixel color via a ***filter function***, ***f***, such that:

$$\widehat{C}_i = f(N_i)$$

Filtering Operation

$$\widehat{C}_i = f(N_i)$$



Moving into Filter Examples

- We will see more details on low and high pass filters in later lectures as the course continues
- What follows is a brief summary and demonstration

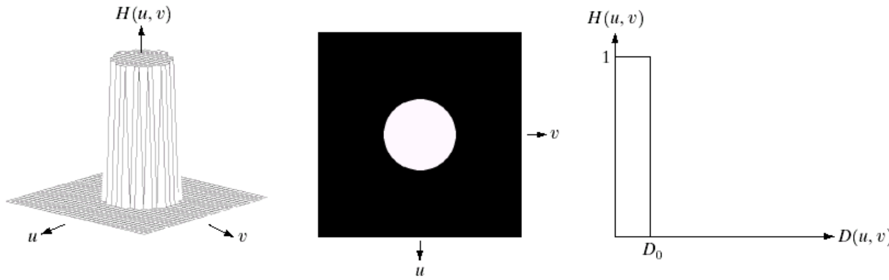
Low Pass Filtering

- Low pass filters are useful for smoothing or blurring images
  - The intent is to retain low frequency information while reducing high frequency information

Example Kernel:

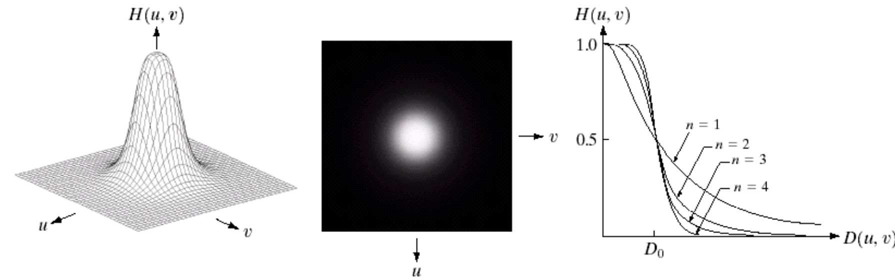
$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Low Pass Filter – Signal Side



a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.



a b c

FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Ideal Filter:

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)> D_0\end{cases}$$

- D(u, v): distance from point (u, v) to the origin
- cutoff frequency (D0)
- nonphysical
- radially symmetric about the origin

Butterworth filter:

$$H(u,v)=\frac{1}{1+\left[D(u,v)/D_0\right]^{2n}}$$

Gaussian Low Pass filter:

$$H(u,v)=e^{-D^2(u,v)/2D_0^{\,2}}$$

High Pass Filtering

- High pass filters are useful for sharpening images (enhancing contrast)
  - The intent is to retain high frequency information while reducing low frequency information

Example Kernel:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

High Pass Filter – Signal Side

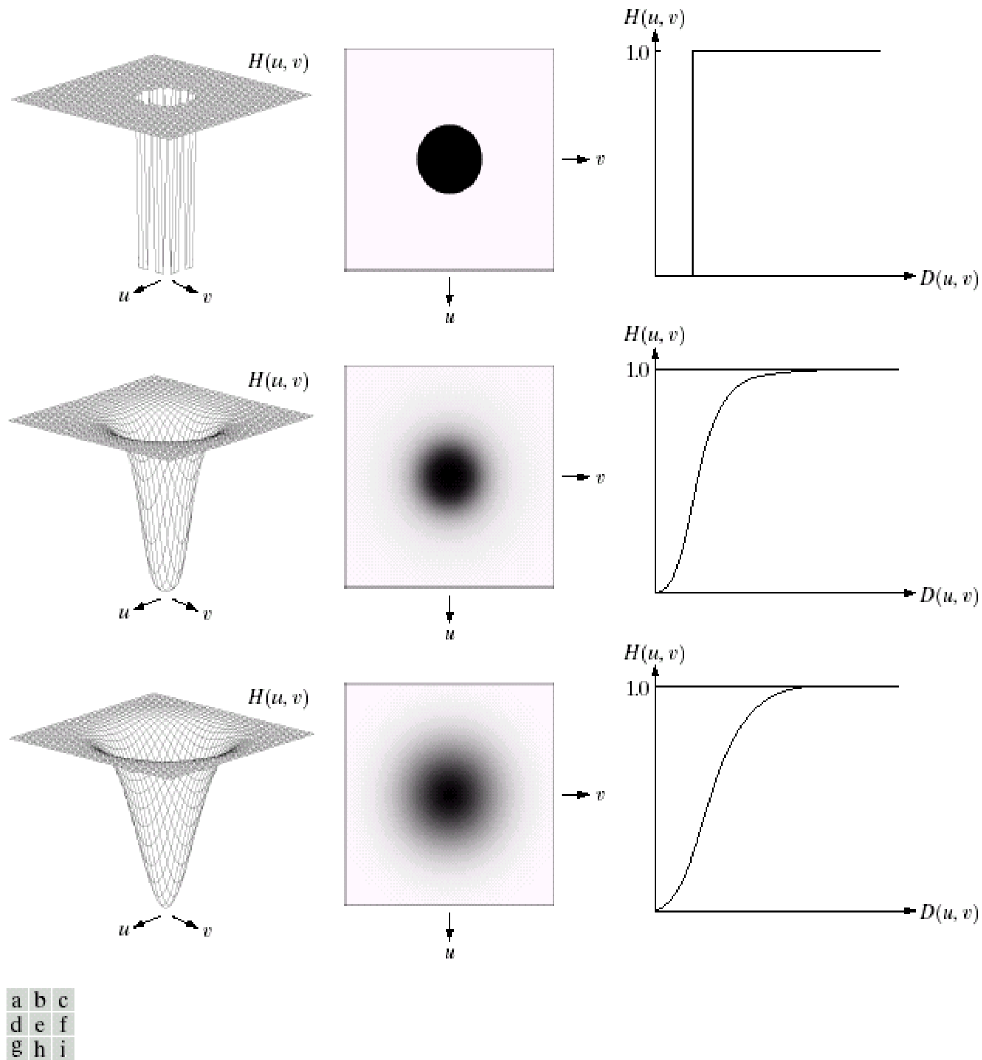


FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Ideal filter:

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Butterworth filter:

$$H(u,v) = \frac{1}{1 + \left[D_0 / D(u,v)\right]^{2n}}$$

Gaussian high pass filter:

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

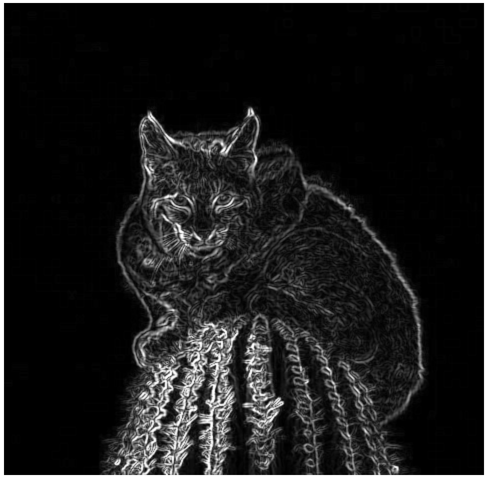
Directional Filtering

- Directional filters are useful for edge detection
  - Can compute the first derivatives of an image
    - Edges are typically visible in images when a large change occurs between adjacent pixels
      - a steep gradient, or slope, or rate of change between pixels

Example Kernels:

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

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



Outline

- **Filtering**
  - Introduction
  - Low Pass Filtering
  - High Pass Filtering
  - Directional Filtering
  - **Global Filters**
    - **Normalization**
    - **Histogram Equalization**

- Image Enhancement
  - Spatial Domain
  - Frequency Domain

Global Filters

- Global Filters
  - Modifies an image's color values
    by taking the entire image as the neighborhood
    about each pixel
  - Two common global filters
    - Normalization
    - Histogram Equalization

Global Filter: Normalization

- Rescale colors to [0, 1]
  - » or to [0, 255]… same as below just multiply by 255

$$\widehat{C_{ij}} = \frac{C_{ij} - C_{min}}{C_{max} - C_{min}}$$

i = pixel index
j = color channel

min and *max*
are across ALL channels

So *max* might be in the red channel
while *min* is in the blue channel

Normalization

a) original image

b) image at 1/2 brightness

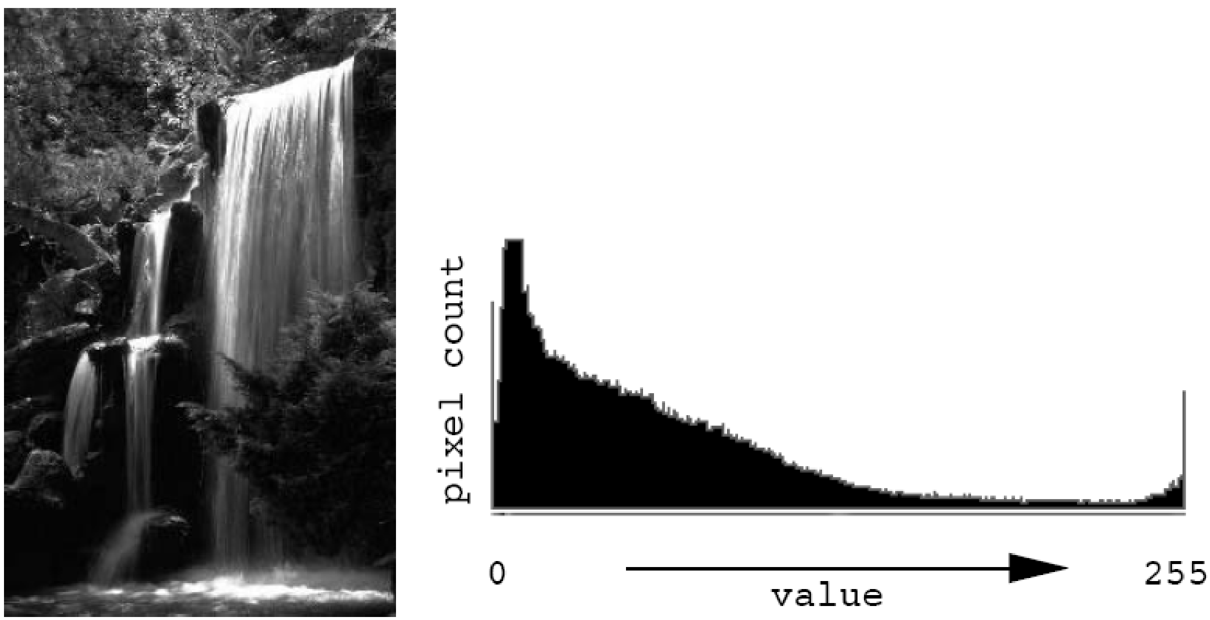
aka under-exposed

c) image renormalized

Normalization recovered
MOST of the value information

Histogram

- An image histogram is a set of tabulations recording how many pixels in the image have particular attributes
- Most commonly
  - A histogram of an image represents the relative frequency of occurrence of various gray levels in the image

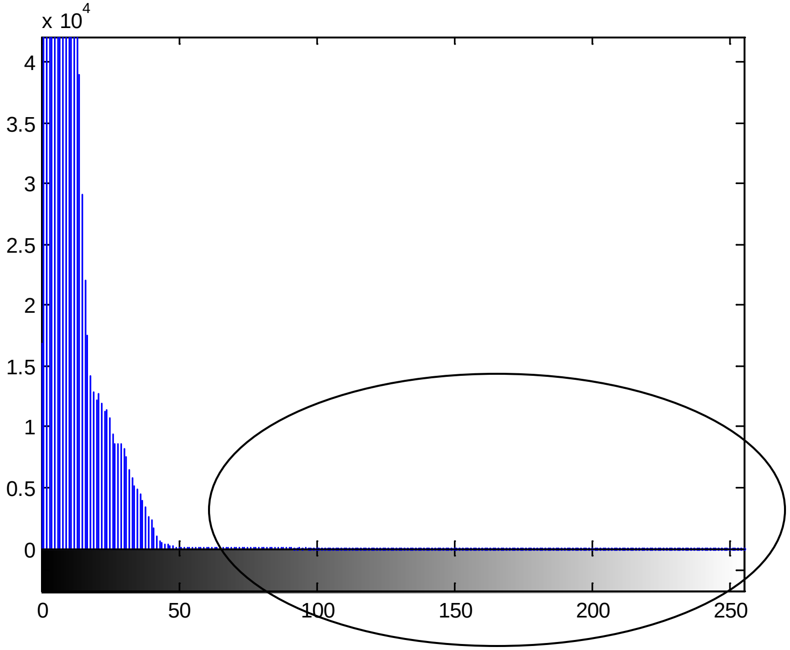


a) original image b) histogram

Figure 8.3: Normally Exposed Image and its Histogram

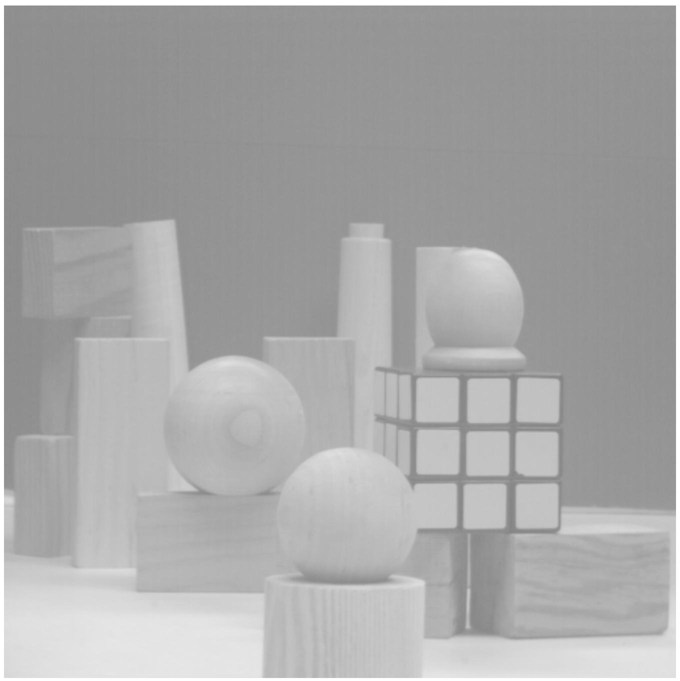
Detecting Exposure Level

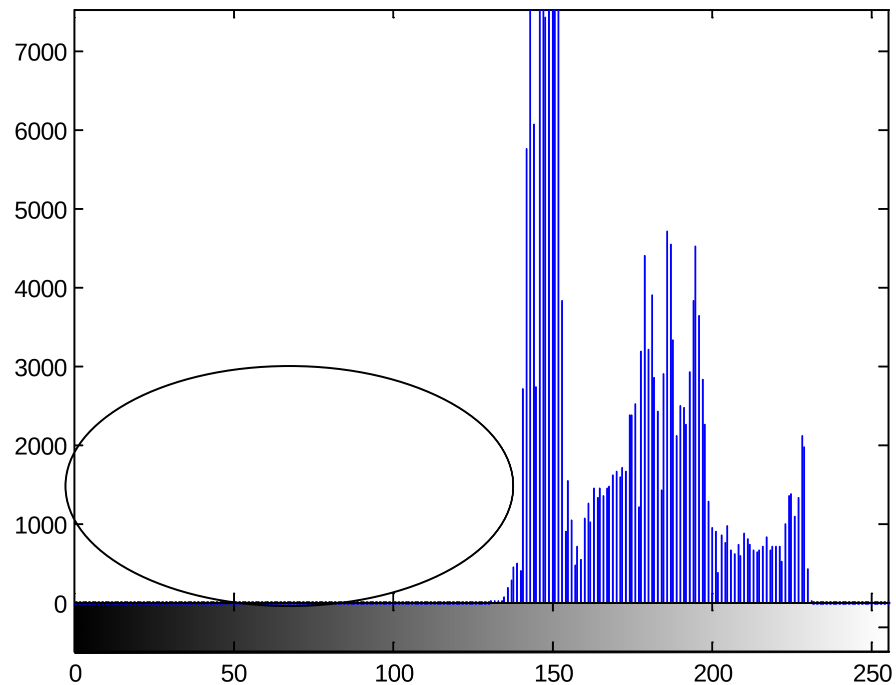
- Underexposure



Detecting Exposure Level

- Over-Exposed





Construction of Histogram: H

- The histogram is constructed based on luminance
  - Must select a luminance algorithm
    - See HTML5-Pixels presentation and Black-And-White Assignment

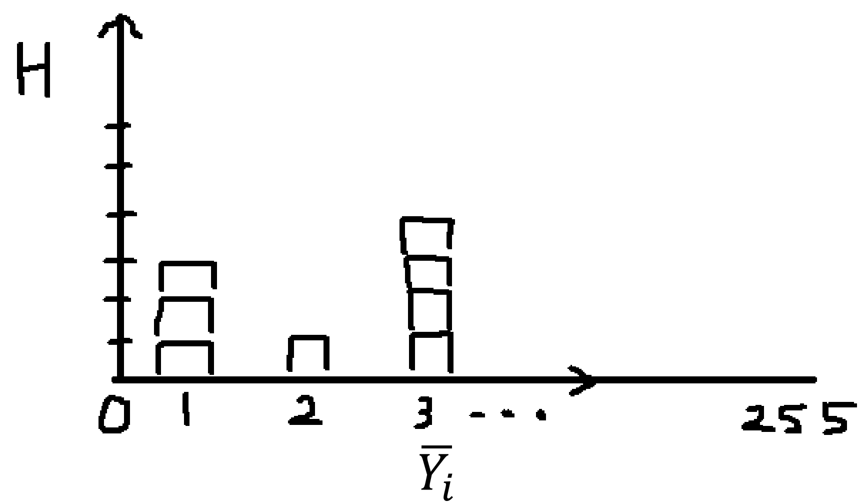
- Given the normalized R, G, B components of pixel *i*, it's perceived luminance is something like:

$$Y_i = 0.30R_i + 0.59G_i + 0.11B_i$$

Construction of Histogram: H

- Create an array H, of size 256
  - Initialize all entries to zero

- For each pixel *i* in the image
  - Compute $Y_i = 0.30R_i + 0.59G_i + 0.11B_i$
    - R_i ,G_i , and B_i all in [0, 1]
    - so Y_i is thus in range [0, 1]
  - Scale and round to nearest integer: 0 to 255
    - $\overline{Y_i}$= Round(Yi * 255)
  - Add 1 to H[$\overline{Y_i}$]

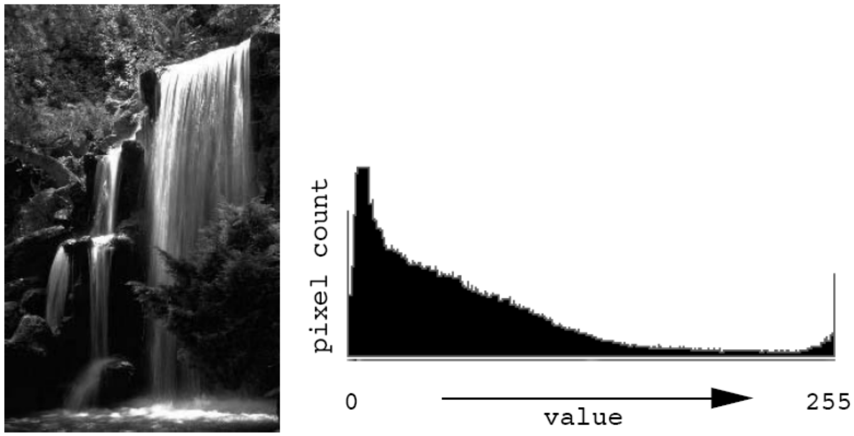


Goal of Equalization

- Goal is not just to use the full range of values, but that the distribution is as uniform as possible

- Basic idea:
  - Find a map f(x) such that the histogram of the modified (equalized) image is flat (uniform)
    - Motivation is to get the cumulative (probability) distribution function (cdf) of a random variable to approximate a uniform distribution
      - where H(i)/(number of pixels in image) offers the probability of a pixel to be of a given luminance

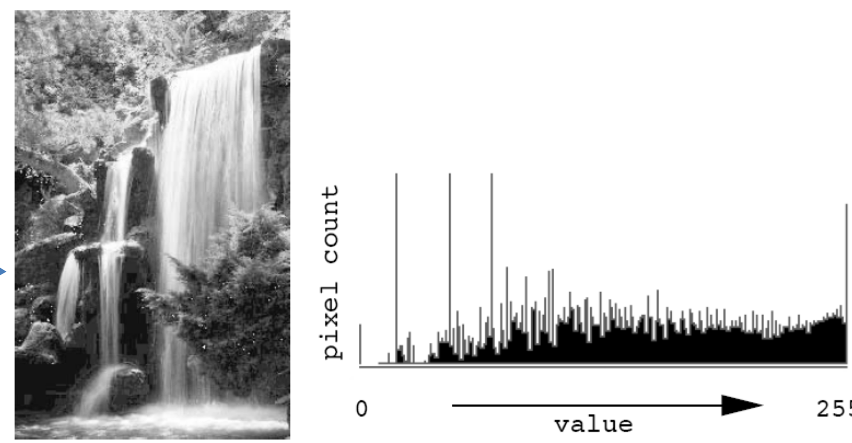
What’s the goal look like



a) original image b) histogram

Figure 8.3: Normally Exposed Image and its Histogram

Spread the distribution out



a) equalized image b) histogram

Figure 8.5: Histogram Equalized Image and its Histogram

Histogram Equalization: a Global Filter

For each pixel *i* in the image, compute the target luminance, $\widehat{Y_i}$

$$\widehat{Y_i} = \frac{1}{T} \sum_{j=0}^{\overline{Y_i}} H[j]$$

T = total number of pixels in the image

$\overline{Y_i}$ = the discrete luminance value of the pixel *i*

H is the histogram table array *(indices from 0 to 255)*

Histogram Equalization: a Global Filter

For each pixel *i* in the image, compute the target luminance, $\widehat{Y_i}$

$$\widehat{Y_i} = \frac{1}{T} \sum_{j=0}^{\overline{Y_i}} H[j]$$

We want the luminance value to be proportional to the accumulated count

This sum is the count of pixels that have luminance equal or less than current pixel i

T = total number of pixels in the image

$\overline{Y_i}$ = the discrete luminance value of the pixel *i*

H is the histogram table array *(indices from 0 to 255)*

Histogram Equalization: a Global Filter

For each pixel *i* in the image, compute the target luminance, $\widehat{Y}_i$

$$\widehat{Y}_i = \frac{1}{T}\underbrace{\sum_{j=0}^{\overline{Y_i}} H[j]}$$

We want the luminance value to be proportional to the accumulated count

This sum is the count of pixels that have luminance equal or less than current pixel i

T = total number of pixels in the image

$\overline{Y_i}$ = the discrete luminance value of the pixel *i*

H is the histogram table array *(indices from 0 to 255)*

For Example:

if there are 100 pixels in the image and 50 pixels have luminance 30 or less

then target luminance, $\widehat{Y}_i$ = 0.50 = 50% mark on histogram table = .5*256 = 128

i.e. luminance of $\overline{Y_i}$ = 30 gets adjusted to target luminance 128

or luminance Y_i = 0.1176 gets adjusted to target luminance 0.5

Histogram Equalization

- Scale factor

$$S_i = \frac{\widehat{Y}_i}{Y_i} \quad \begin{matrix} = 0.50 = \text{target value} \\ = 0.1176 = \text{original value} \end{matrix} \Bigg\} \quad S_i = 4.25$$

Use scale factor to rescale the original Red, Green, and Blue channel values for the given pixel *i* :

new R = 4.25 * (old R)
new G = 4.25 * (old G)
new B = 4.25 * (old B)

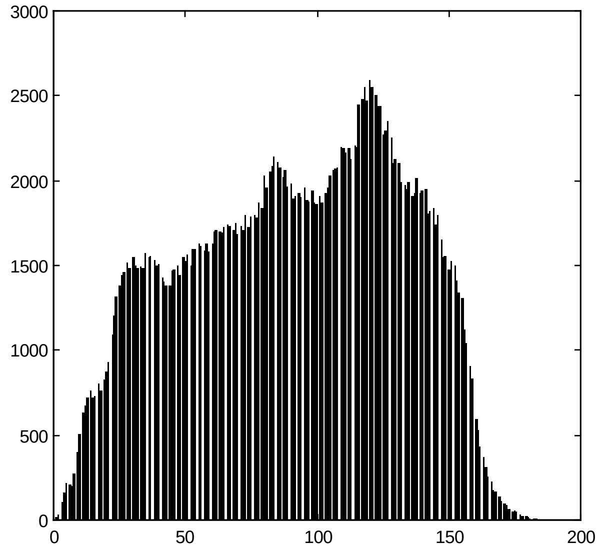
NOTE:
Various rounding/truncating/numerical errors may cause final values to be outside [0, 1]
Renormalize results by finding min and max as described earlier in this presentation

Example Equalization: Images

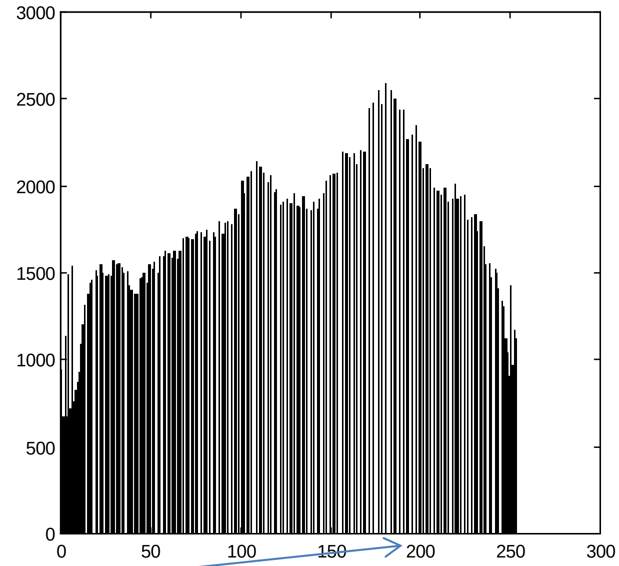
before

after

Example Equalization: Histograms



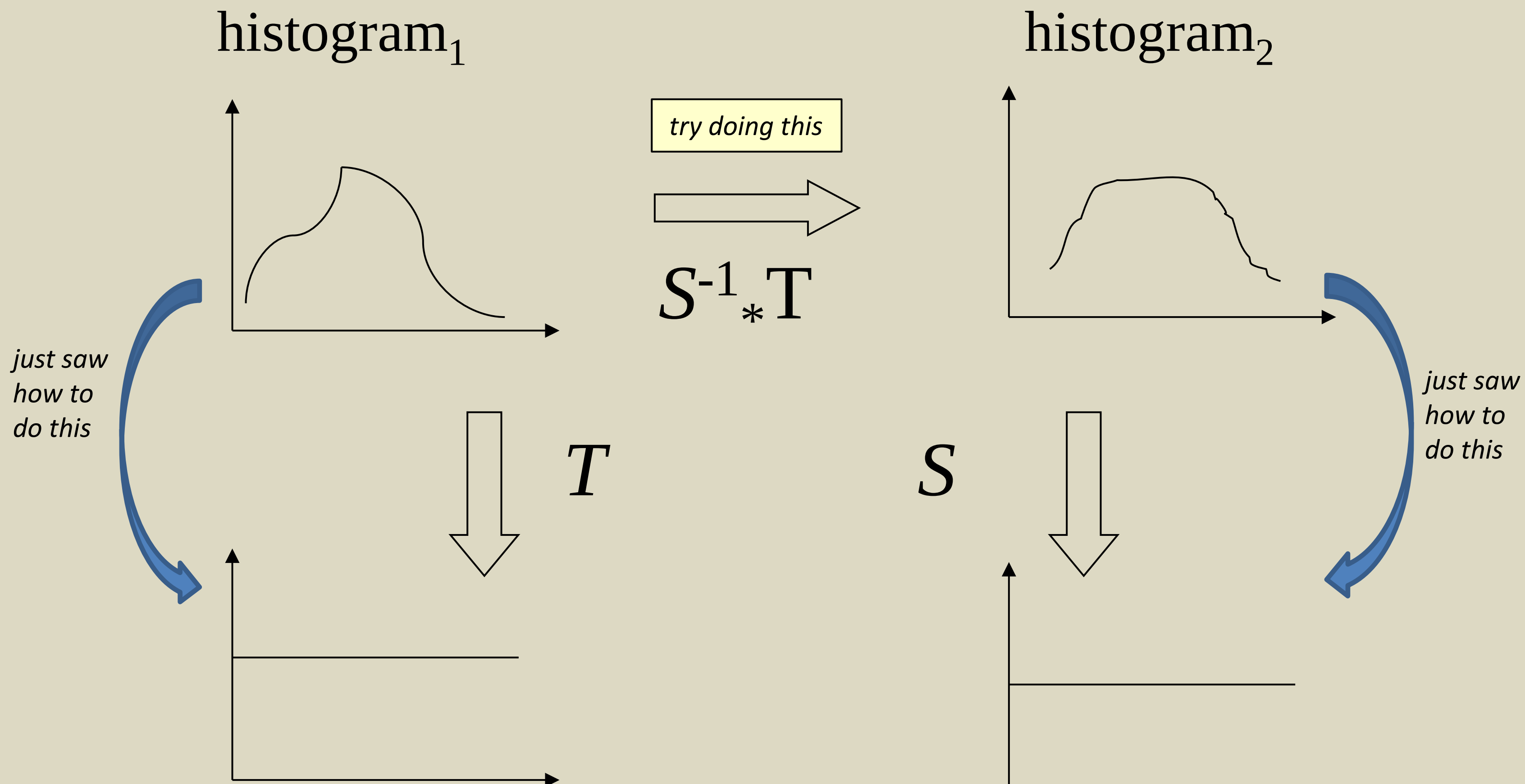
before equalization



after equalization

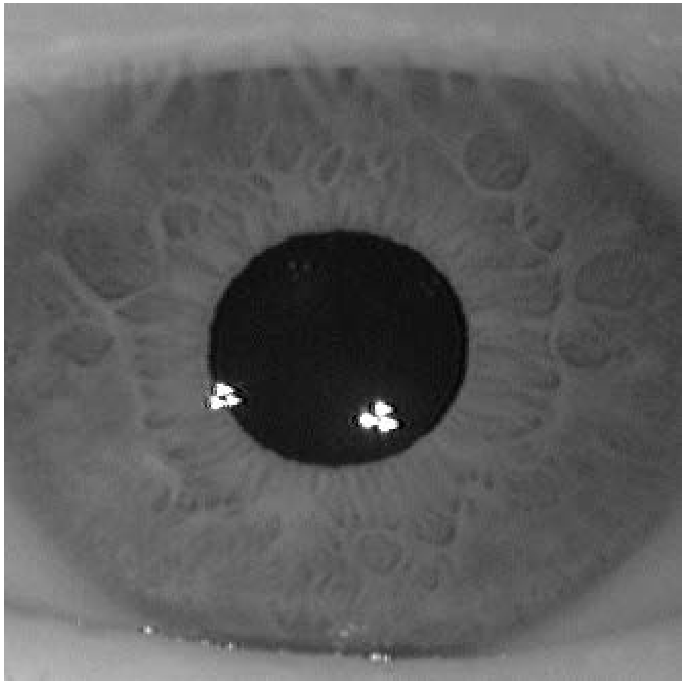
Challenge: Histogram Specification/Matching

- Given a target image B
  - How would you modify a given image A such that the histogram of the modified A matches that of target B?

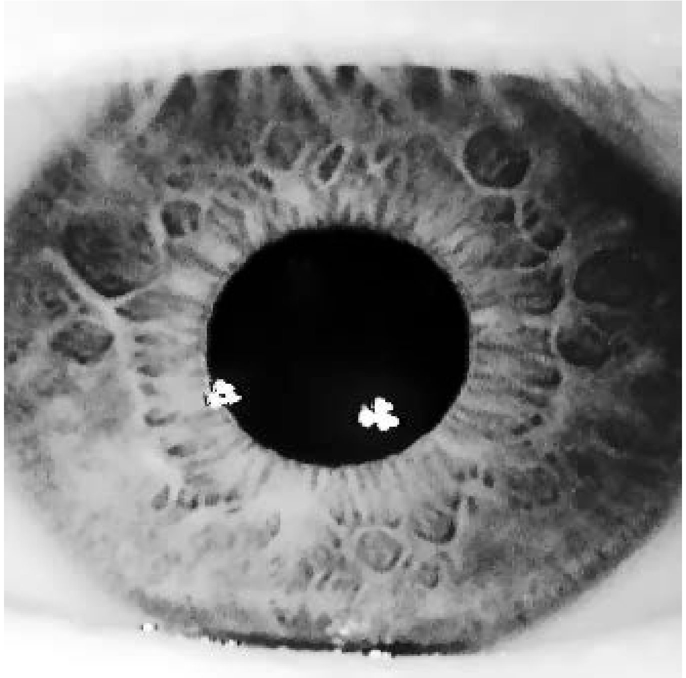


Application: Photography

Application: Bioinformatics



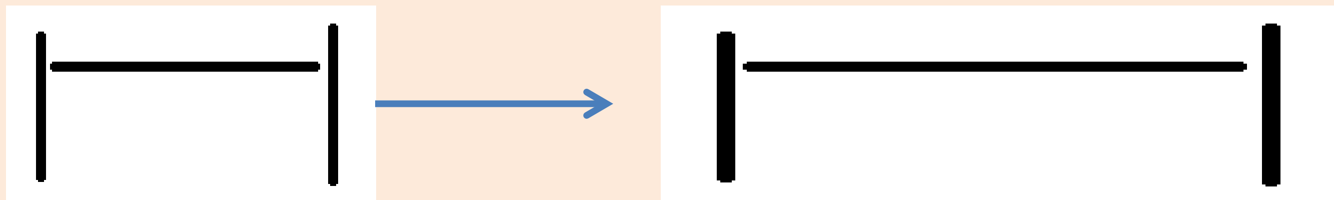
before



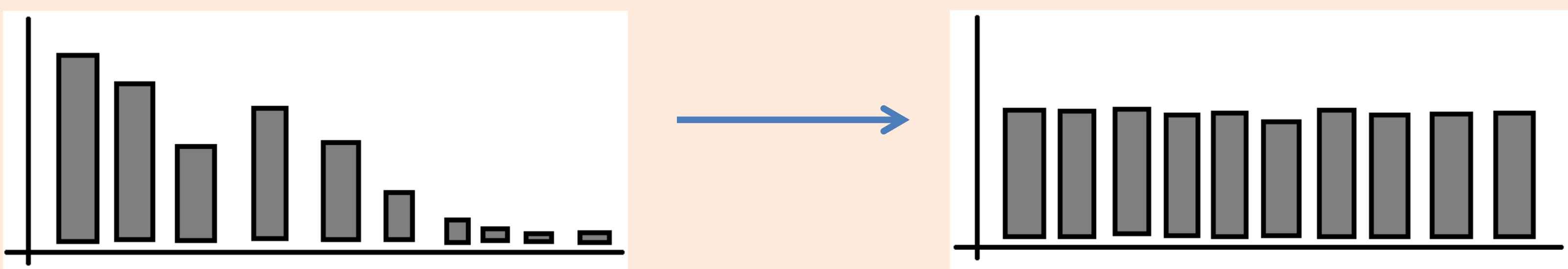
after

Summary: Normalization vs. Equalization

- Normalization: Stretches



- Histogram Equalization: Flattens



» And Normalization may be needed after equalization

Questions so far

- Any questions on Filtering?

- Next: Image Enhancement

Outline

- Filtering
  - Introduction
  - Global Filters
    - Normalization
    - Histogram Equalization

- Image Enhancement
  - Spatial Domain
  - Frequency Domain

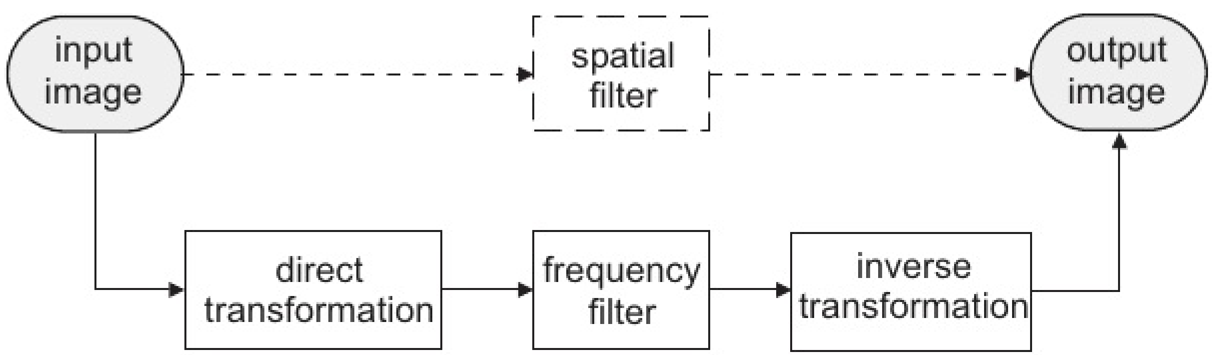
Image Enhancement

- Objective of Image Enhancement is to manipulate an image such that the resulting end image is more suitable than the original for a specific application

- Two broad categories to due this

  – Spatial Domain
    - Approaches based on the direct manipulation of pixels in an image

  – Frequency Domain
    - Approaches based on modifying the Fourier transform of an image

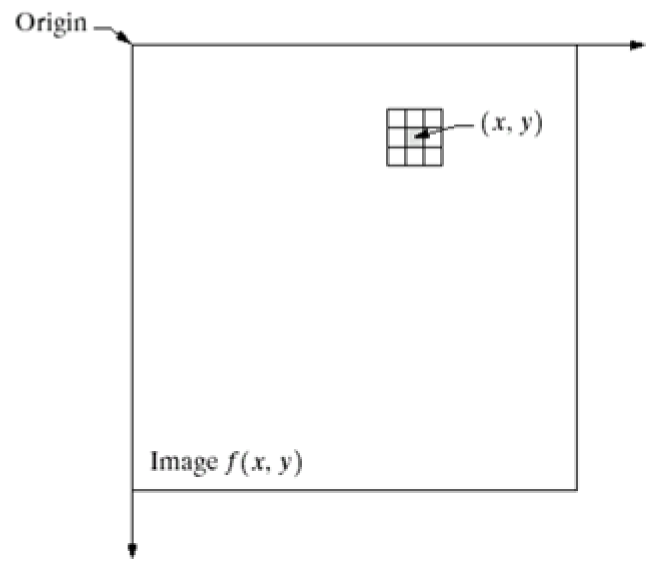


Spatial Domain Methods: Intro

Spatial domain methods
are procedures that operate directly on image pixels

$$g(x, y) = T[f(x, y)]$$

$f(x, y)$ is the input image
$g(x, y)$ is the output image
T is an operator on *f*, defined over some neighborhood of *(x, y)*
sometimes T can operate on a set of input images
(e.g. difference of 2 images)



Types of Spatial Methods

- Spatial Domain Methods
  - Image Normalization
  - Histogram Equalization
  - Point Operations

We just saw these!

Types of Spatial Methods

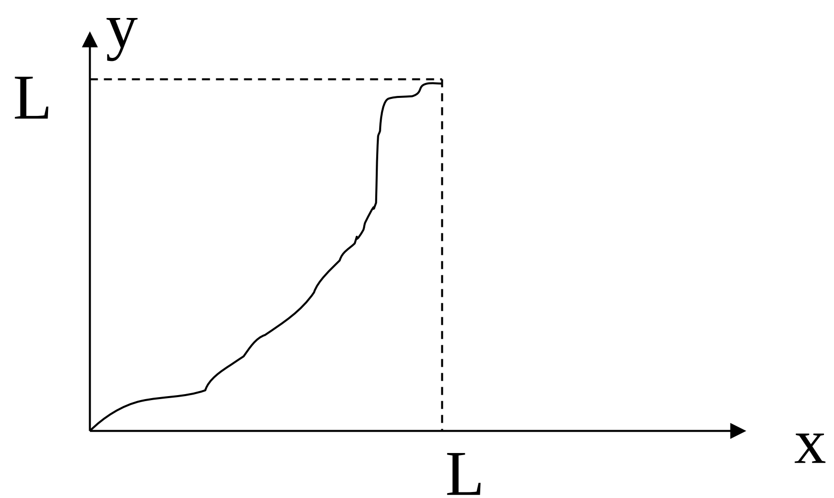
- Spatial Domain Methods
  - Image Normalization
  - Histogram Equalization
  - Point Operations

Let’s look at some of these

Point Operations: Overview

- Point operations are zero-memory operations
  - where a given gray level *x* in [0, L] is mapped to another gray level *y* in [0, L] as defined by

$$y = f(x)$$

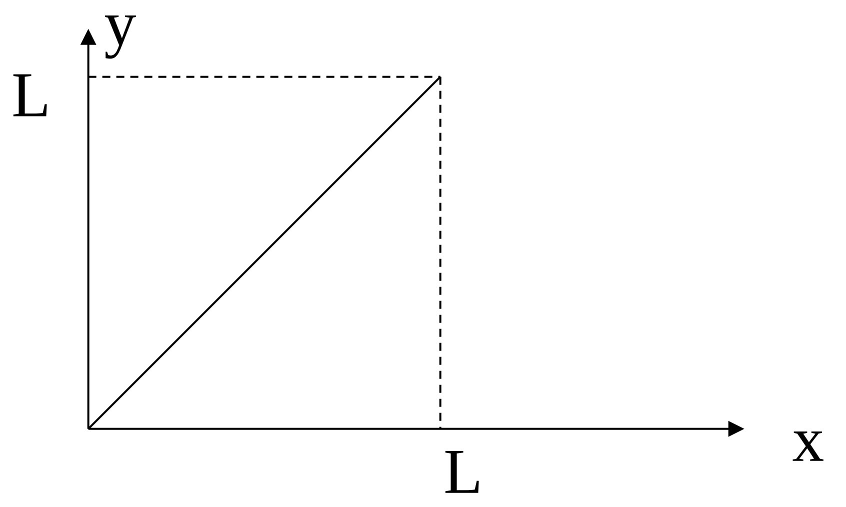


L=255: for grayscale images

ASIDE: 1 is okay for L also,
just keep your abstraction consistent

Trivial Case

$$y = x$$

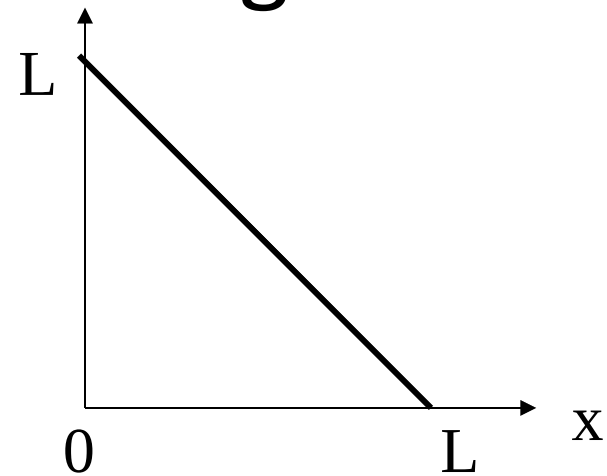


No influence on visual quality at all

L=255: for grayscale images

Negative Image

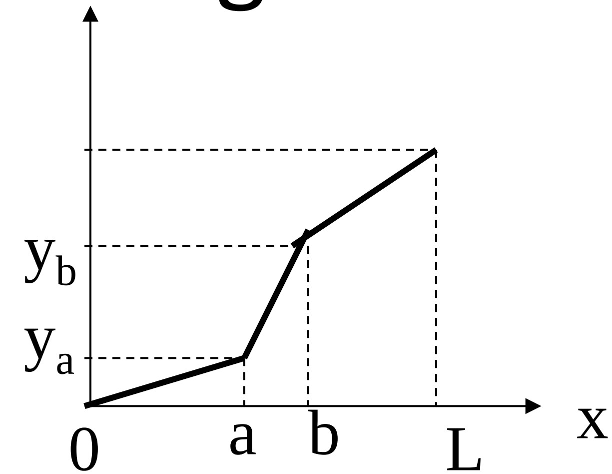
$$y = L - x$$

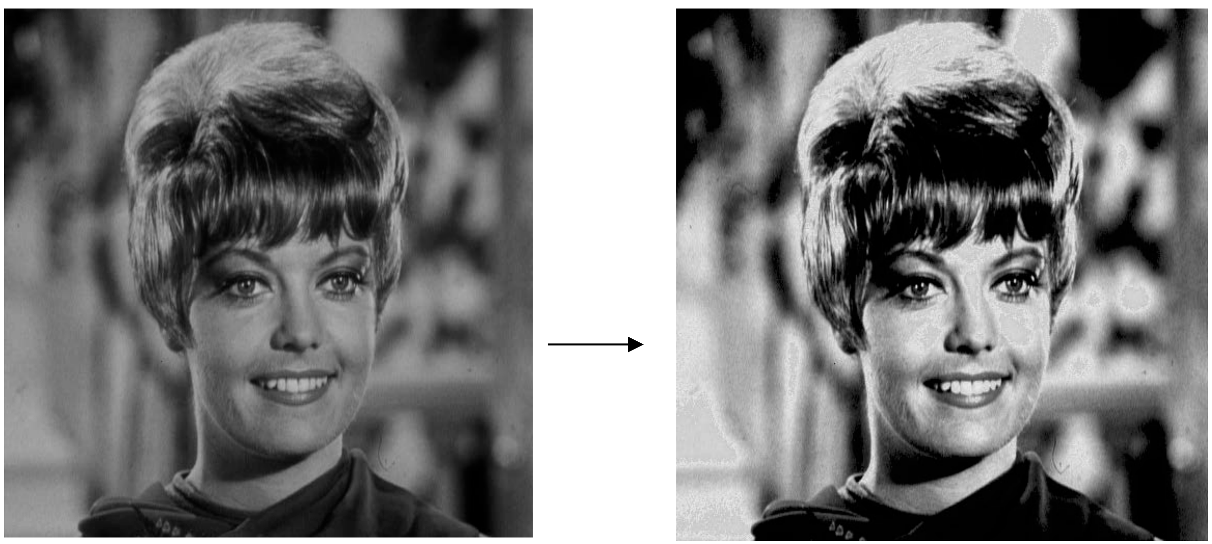


L=255: for grayscale images

Contrast Stretching

$$y=\begin{cases}\alpha x & 0\le x<a\\ \beta(x-a)+y_a & a\le x<b\\ \gamma(x-b)+y_b & b\le x<L\end{cases}$$



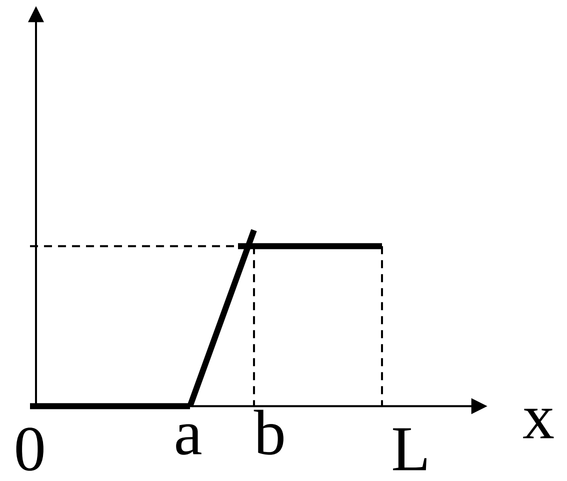


$a=50, b=150, \alpha=0.2, \beta=2, \gamma=1, y_a=30, y_b=200$

L=255: for grayscale images

Clipping

$$y = \begin{cases} 0 & 0 \le x < a \\ \beta(x-a) & a \le x < b \\ \beta(b-a) & b \le x < L \end{cases}$$



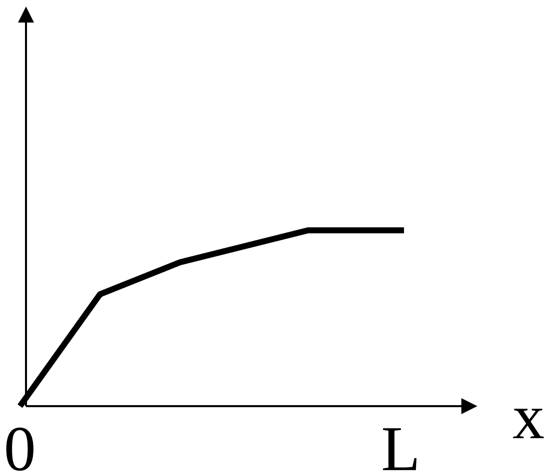
$a = 50, b = 150, \beta = 2$

L=255: for grayscale images

Range Compression

aka Log Transformation

$$y = c\log_{10}(1+x)$$



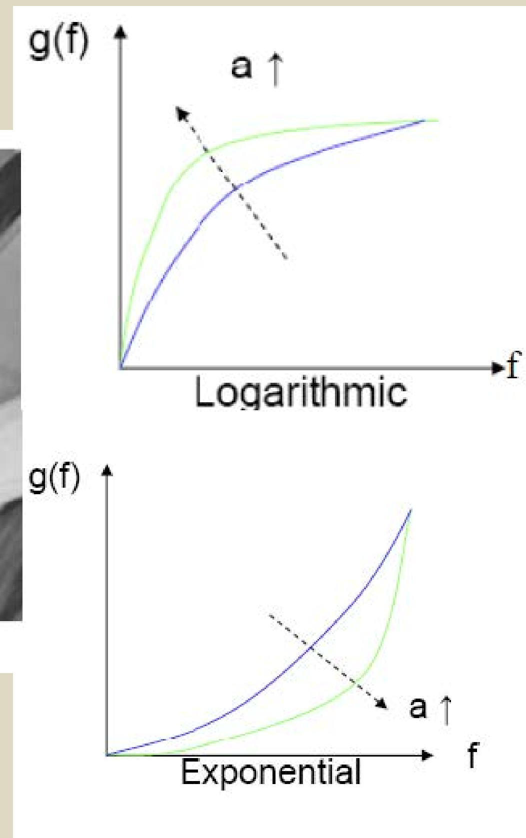
c=100

L=255: for grayscale images

Challenge

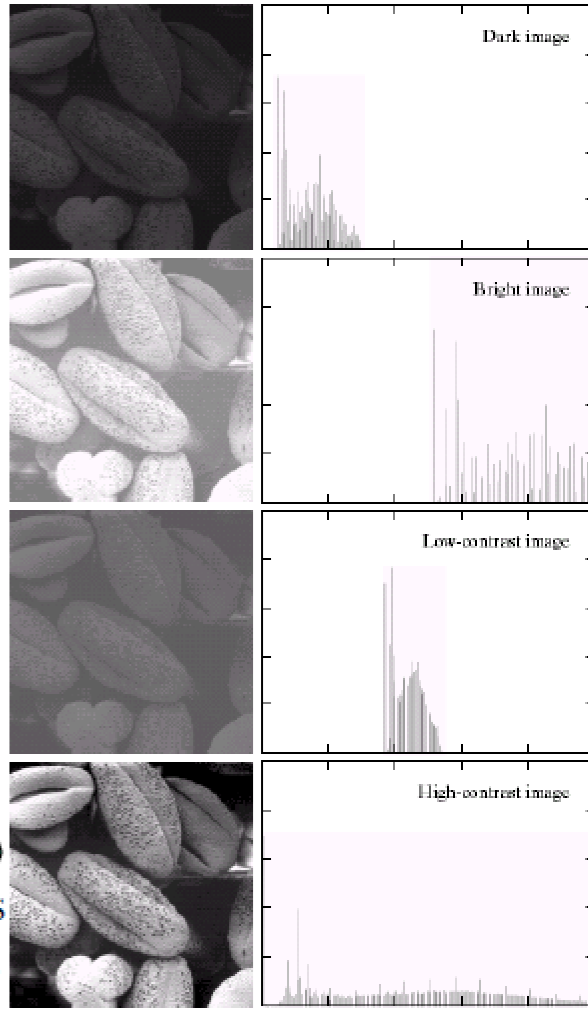
- Investigate Power-Law Transformations
  - HINT: see also gamma correction

$$y = cx^{\gamma}$$



Relation to Histograms

An image's histogram may help decide what operation may be needed for a desired enhancement



a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Dark image

The components of the histogram are concentrated on the low side of the gray scale.

Bright image

The components of the histogram biased toward the high side of the gray scale.

Low Contrast image

The histogram will be narrow and will be concentrated on toward the middle of the gray scale.

High Contrast image

whose pixels have a large variety of gray tones. The histogram is not far from a uniform.

Summary: Spatial Domain Enhancements

- Spatial domain methods
  are procedures that operate directly on image pixels

- Spatial Domain Example Methods
  - Image Normalization
  - Histogram Equalization
  - Point Operations

- All can be viewed as a type of filtering
  - Point operations have a neighborhood of self
  - Normalization and Equalization have a neighborhood of the entire image

- Histogram can help “detect” what type of enhancement might be useful

Questions so far

- Any questions on
  Spatial Domain Image Enhancement ?

- Next:
  Frequency Domain Image Enhancement

Outline

- Filtering
  - Introduction
  - Global Filters
    - Normalization
    - Histogram Equalization
- Image Enhancement
  - Spatial Domain
  - **Frequency Domain**

Unsharp Masking

- Unsharp Mask
  - can be approached as a spatial filter
  - or in the frequency domain

- It is mentioned here as a transition
  - from spatial to frequency

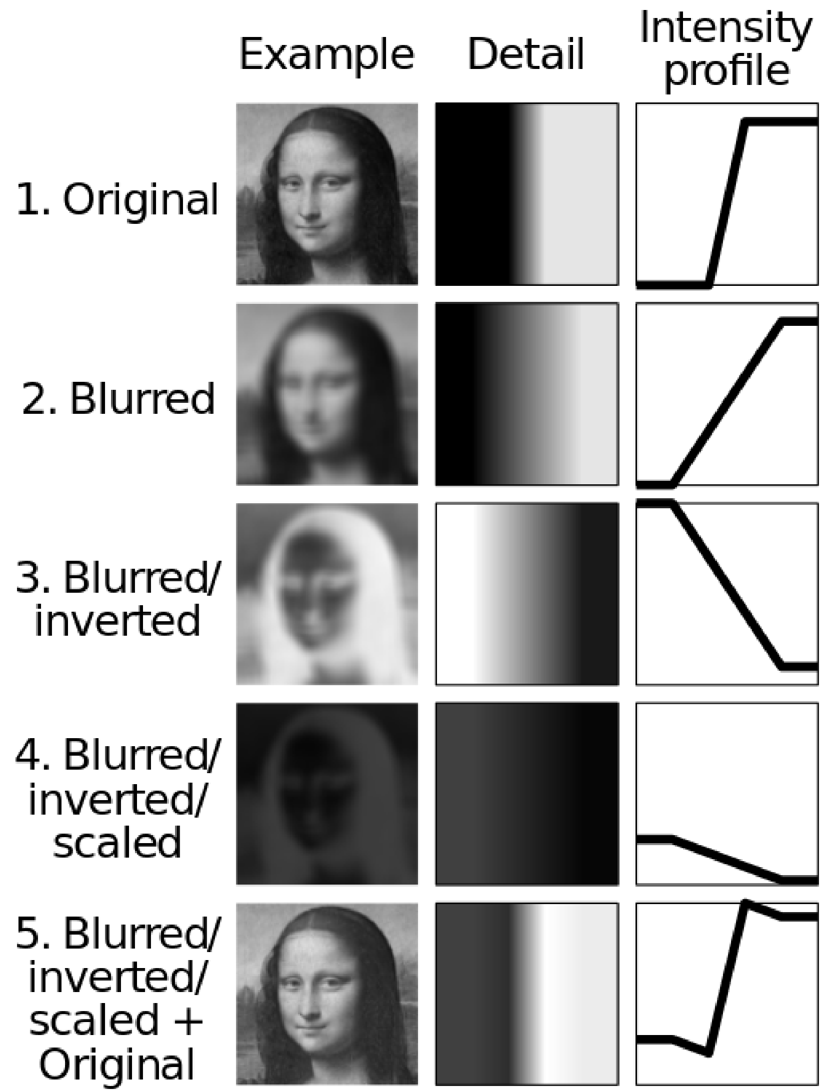
- It would be an advised learning experience to implement and understand this filter in both domains

Unsharp Masking *aka high-boost filtering*

- Image Sharpening Technique
  – Unsharp is in the name because image is first blurred

- A summary of the method is outlined to the right

- In general the method amplifies the high frequency components of the signal image



public domain image from wikipedia

Unsharp Mask: Mathematically

$$y(m,n) = x(m,n) + \lambda g(m,n), \lambda > 0$$

$g(m,n)$ is a high-pass filtered version of $x(m,n)$

Example using Laplacian operator:

$$g(m,n) = x(m,n) - \frac{1}{4}[x(m-1,n) + x(m+1,n) + x(m,n-1) + x(m,n+1)]$$

Recall:
Laplacian L(x, y) of an image
with pixel intensity values I(x, y) is

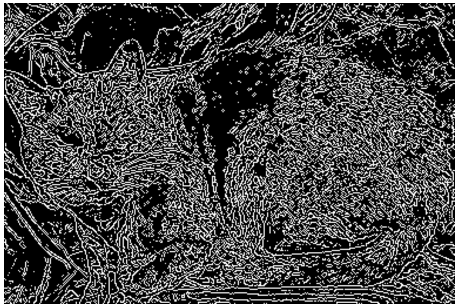
$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

The above equation is using a discrete approximation filter/kernel of the Laplacian:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

This will be highly sensitive to noise
Applying a Gaussian blur and then Laplacian might be a better option

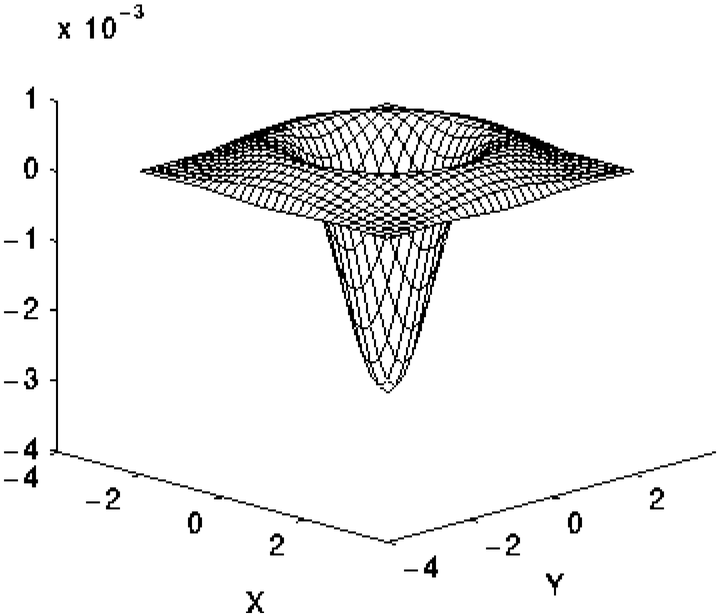
Alternate option



More smoothing reduces number of edges detected… Do you see the cat?

- Laplacian of Gaussian → LoG

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Discrete Kernel for this with sigma = 1.4 looks like:

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Questions so far?

- Questions on Unsharp Frequency Filter?

  - Next: Homomorphic Frequency Filter

Noise and Image Abstraction

Noise is usually abstracted as additive:

$$\bar{I}(x,y) = I(x,y) + n(x,y)$$

Homomorphic filtering considers noise to be multiplicative:

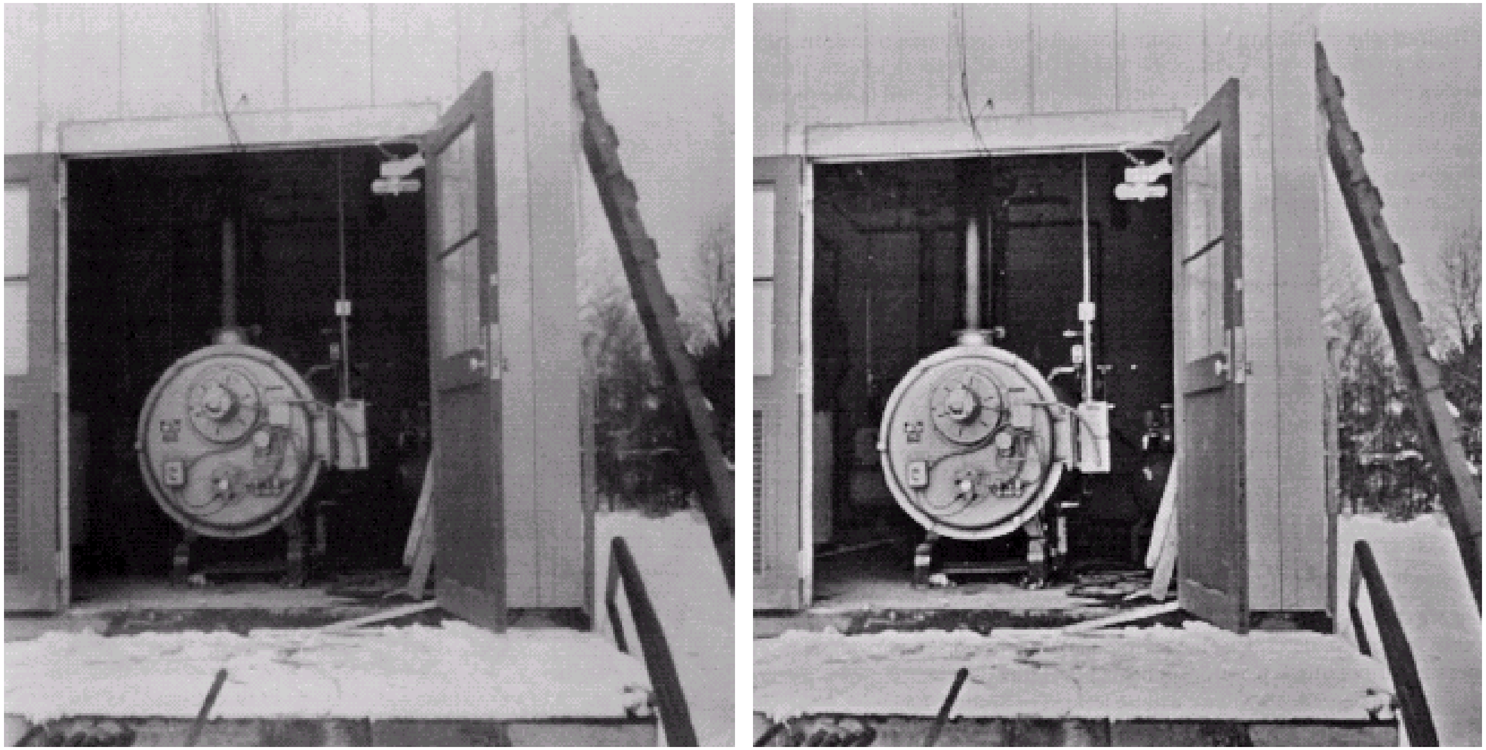
$$\bar{I}(x,y) = I(x,y)n(x,y)$$

Homomorphic filtering is useful in correcting non-uniform illumination in images
→normalizes illumination and increases contrast
→suppresses low frequencies and amplifies high frequencies (in a log-intensity domain)

Homomorphic Frequency Filter: Example

a b

FIGURE 4.33 (a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)



before

after

Homomorphic Filtering: Intro

- **Illumination** varies slowly across the image
  - low frequency
  - slow spatial variations
- **Reflectance** can change abruptly at object edges
  - high frequency
  - varies abruptly, particularly at object edges
- **Homomorphic Filtering** begins with a transform of multiplication to addition
  - via property of logarithms

Recall previous def of image:

$$I(x, y) = L(x, y)R(x, y)$$

$$\ln(I(x, y)) = \ln(L(x, y)R(x, y))$$

$$\ln\big(I(x, y)\big) = \ln\big(L(x, y)\big) + \ln(R(x, y))$$

I is the image (gray scale),
L is the illumination
R is the reflectance

0 < L(x,y) < infinity
0 < R(x, y) < 1

Apply High Pass Filter

- Once we are in a log-domain
  - Remove low-frequency illumination component by applying a high pass filter in the log-domain
    - so we keep the high-frequency reflectance component

Homomorphic Filtering

Apply High Pass Filter

- Once we are in a log-domain
  - Remove low-frequency illumination component by applying a high pass filter in the log-domain
    - so we keep the high-frequency reflectance component

This is an example of Frequency Domain Filtering

Homomorphic Filtering

Apply High Pass Filter

Homomorphic Filtering

This is an example of Frequency Domain Filtering

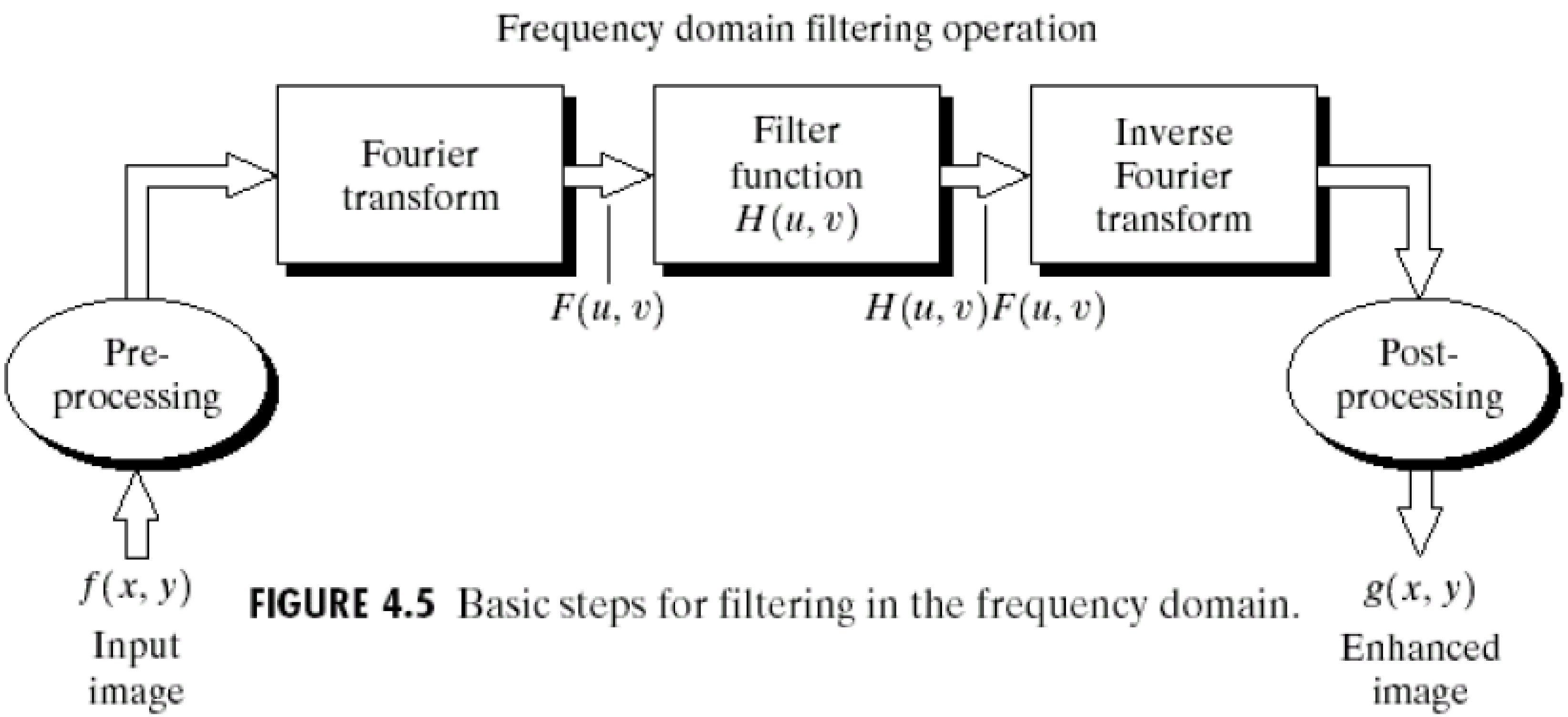
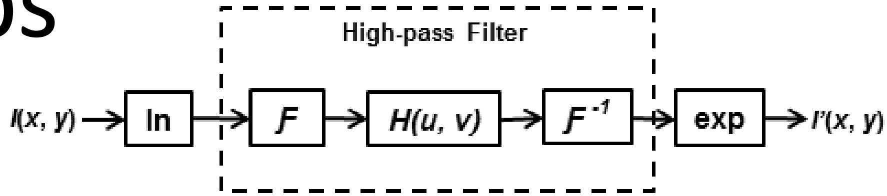


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Steps



Step 1: Convert the image into the log domain

$$z(x, y) = \ln\big(I(x, y)\big) = \ln\big(L(x, y)\big) + \ln(R(x, y))$$

Step 2: Apply a High Pass Filter

$$F\{z(x, y)\} = F\big\{\ln\big(I(x, y)\big)\big\} = F\big\{\ln\big(L(x, y)\big)\big\} + F\{\ln(R(x, y))\}$$

$$Z(u, v) = F_L(u, v) + F_R(u, v)$$

$$S(u, v) = H(u, v)F_L(u, v) + H(u, v)F_R(u, v)$$

$$s(x, y) = L'(x, y) + R'(x, y)$$

Step 3: Apply exponential to return from log domain

$$g(x, y) = exp[s(x, y)] = exp[L'(x, y)] + \exp[R'(x, y)]$$

Summary: Filters and Enhancement

Filtering

- Introduction
- Low Pass Filtering
- High Pass Filtering
- Directional Filtering
- Global Filters
  - Normalization
  - Histogram Equalization

Image Enhancement

- Spatial Domain Methods
  - Image Normalization
  - Histogram Equalization
  - Point Operations
    - Image Negatives
    - Contrast Stretching
    - Clipping
    - Range Compression
- Frequency Domain Methods
  - Unsharp Filter
  - Homomorphic Filter

Questions?

- Beyond D2L
  - Examples and information can be found online at:
    - *http://docdingle.com/teaching/cs.html*

    - *Continue to more stuff as needed*

Extra Reference Stuff Follows

Credits

- Much of the content derived/based on slides for use with the book:
  - *Digital Image Processing,* Gonzalez and Woods

- Some layout and presentation style derived/based on presentations by
  - Donald House, Texas A&M University, 1999
  - Bernd Girod, Stanford University, 2007
  - Shreekanth Mandayam, Rowan University, 2009
  - Igor Aizenberg, TAMUT, 2013
  - Xin Li, WVU, 2014
  - George Wolberg, City College of New York, 2015
  - Yao Wang and Zhu Liu, NYU-Poly, 2015
  - Sinisa Todorovic, Oregon State, 2015