Image Enhancement in Spatial Domain
UNIT-2

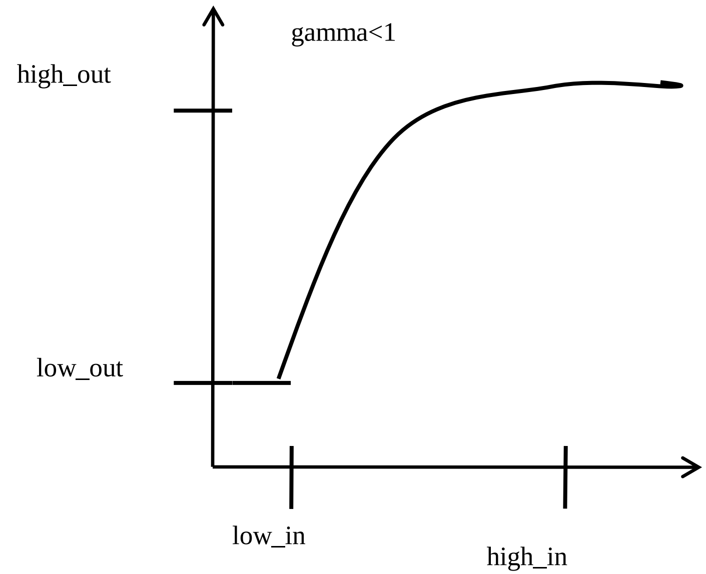
P. Balamurugan, Assistant Professor
PG & Research Department of Computer Science
Government Arts College, Coimbatore -641018
Email: spbalamurugan@rediffmail.com

Basic Gray level Transformations

- In a digital image, point = pixel
- Point Processing Operations
  - Intensity transformation
  - Histogram equalization
  - Spatial filtering

Intensity Transformation Functions

- s=T(r) , where r denotes the intensity of f and s is the intensity of g, both at any (x, y) in the image
- imadjust
  - g=imadjust(f,[low_in high_in], [low_out high_out], gamma)
  - Values between low_in and high_in is mapped to values between low_out and high_out



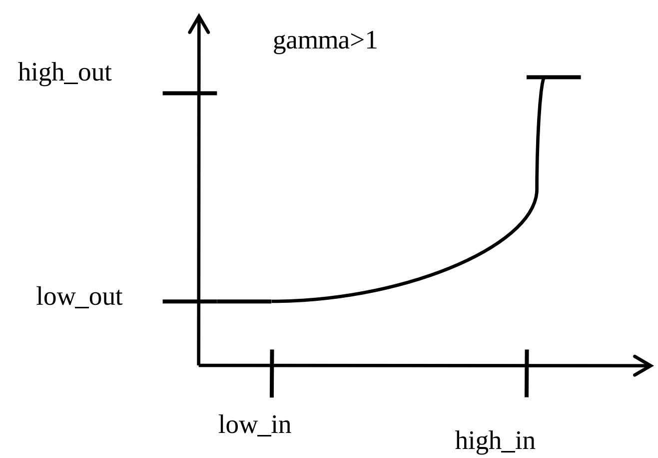
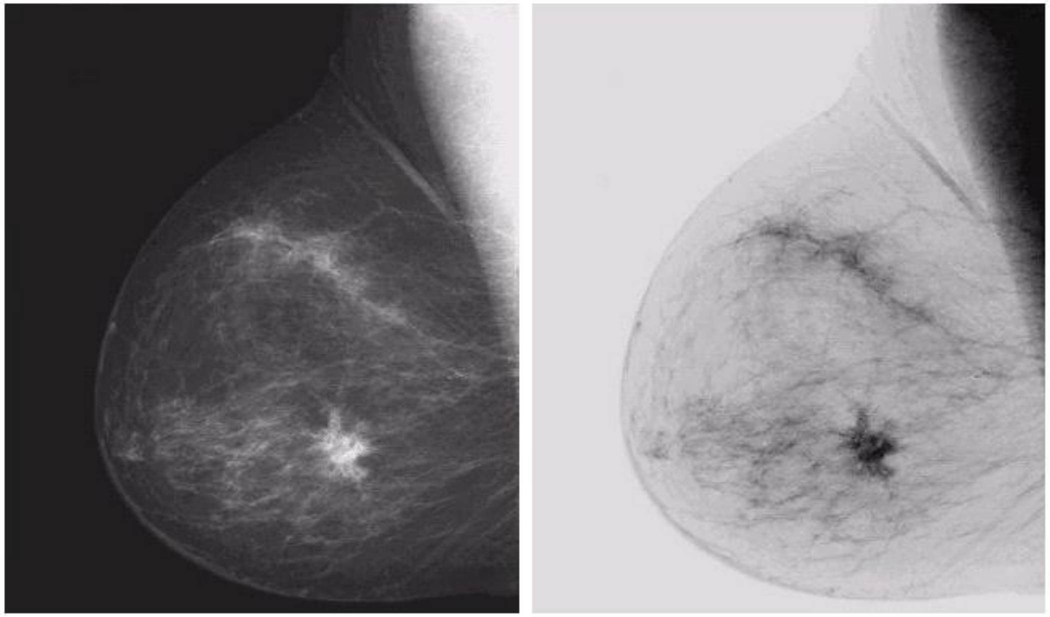


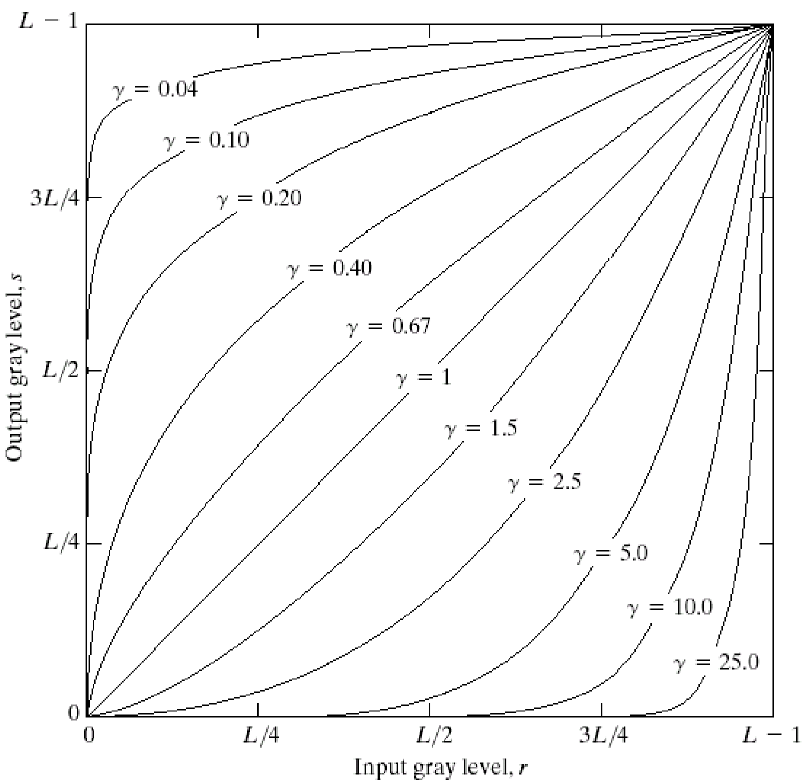
Image Negatives

- $s = \mathrm{T}(r) = L-1-r$
- *Similar to photo negatives.*
- *Suitable for enhancing white or gray details in dark background.*

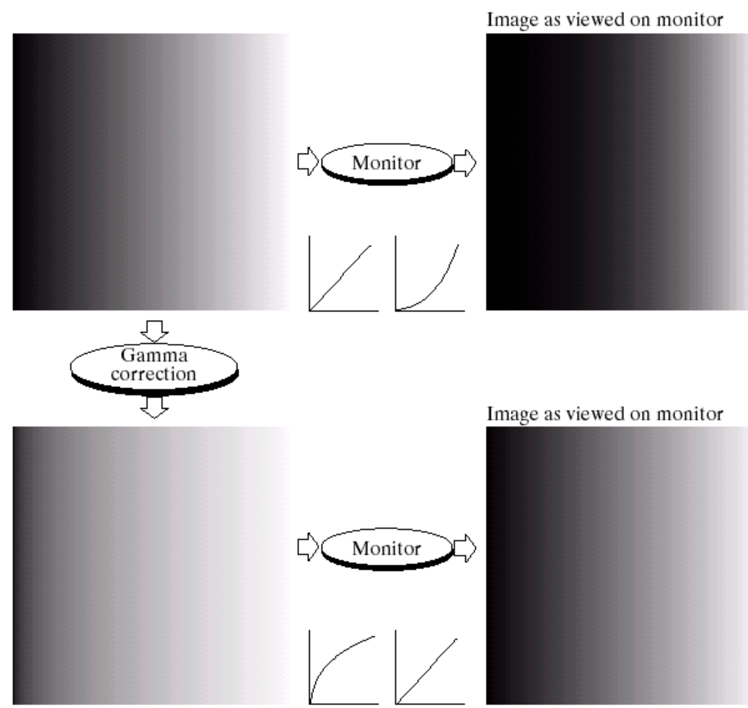


Power Law Gray-level Transform

$$s = \mathrm{T}(r) = cr^{\gamma}$$



- Gamma correction: to compensate the built-in power law compression due to display characteristics.



Contrast Enhancement

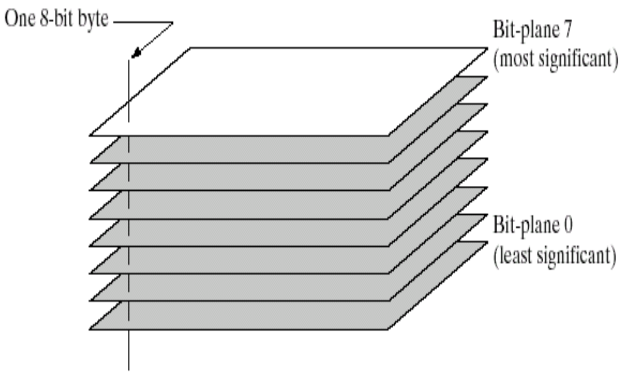
- Piecewise linear Transformation
- Input : Poor illumination images
- Lack of dynamic range
- Increase the dynamic range of gray levels
- In raw imagery, data occupies only a small portion of the available range of digital values (commonly 8 bits or 256 levels).
- Contrast enhancement involves changing the original values so that more of the available range is used,
- Increases the contrast between targets and their backgrounds.

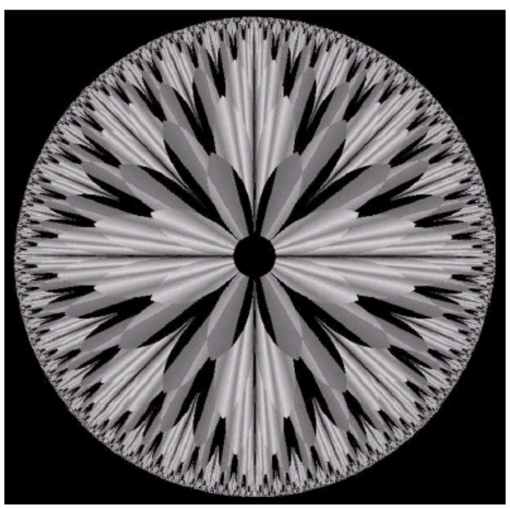
Other Piece-wise Transformation

- Gray level Slicing

- Bit plane slice

Bit-Slicing





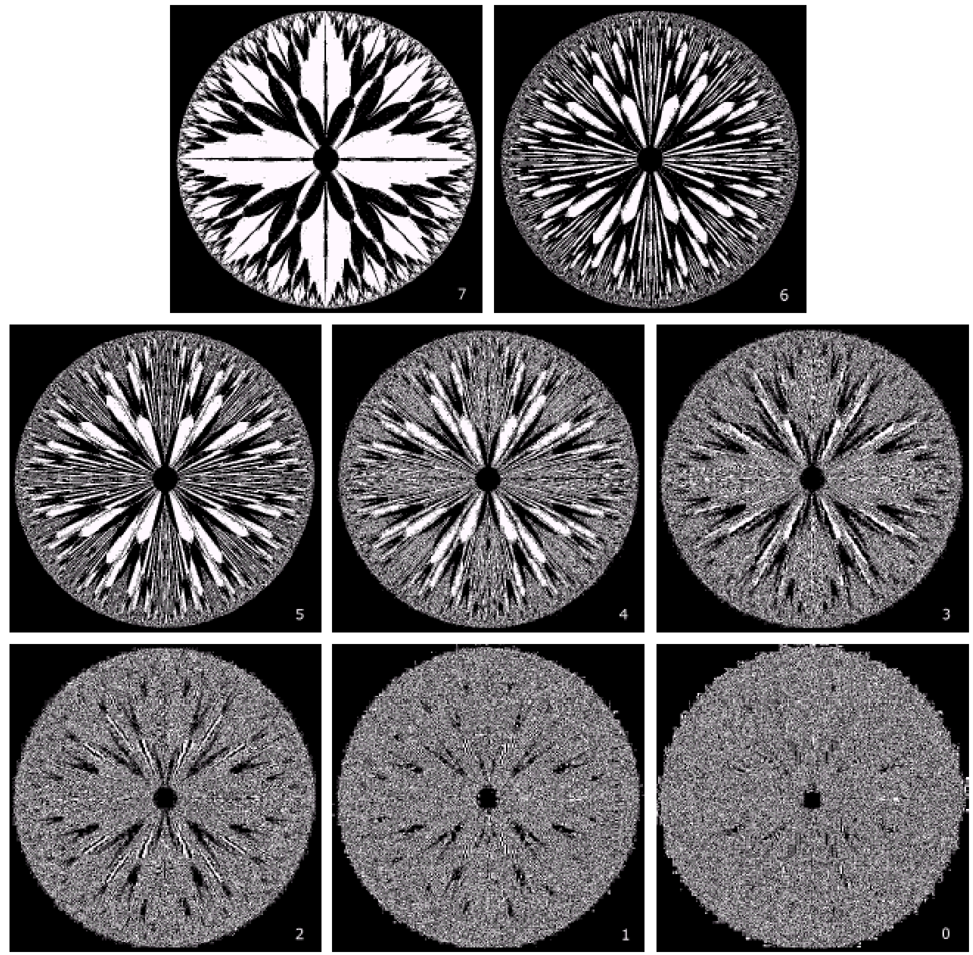


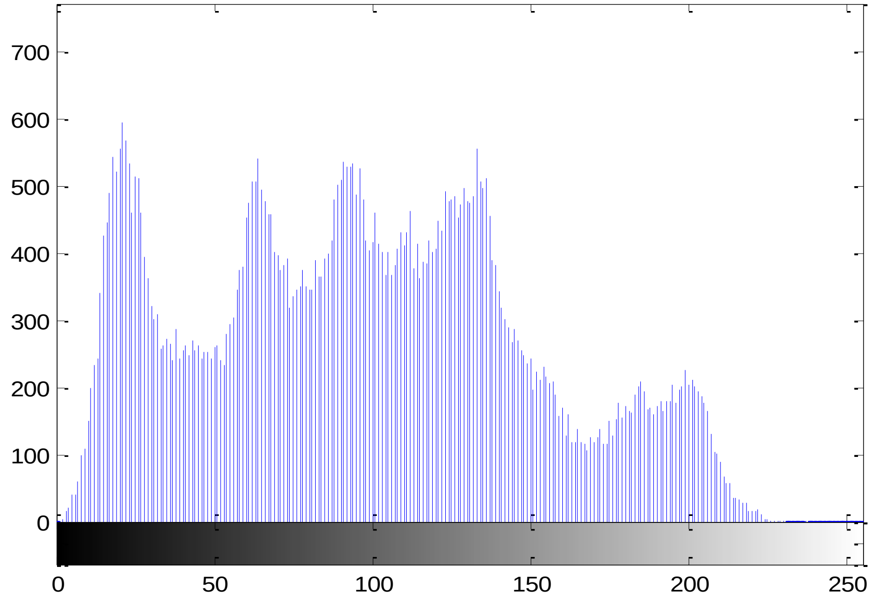
FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Histogram

- Let I be a grayscale image.
- $I(r,c)$ is an 8-bit integer between 0 and 255.
- Histogram, h_I, of I:
  - a 256-element array, h_I
  - $h_I(g)$, for g = 1, 2, 3, …, 256, is an integer
  - $h_I(g)$ = number of pixels in I that have value g-1.

Histogram of a Grayscale image

```
f=imread('lena.bmp');
Imhist(f);
```



Histogram of a Color image

- If I is a 3-band image (truecolor, 24-bit)
- Either I has 3 histograms:
  - $h_R(g)$ = # of pixels in I(:,:,1) with intensity value *g-1*
  - $h_G(g)$ = # of pixels in I(:,:,2) with intensity value *g-1*
  - $h_B(g)$ = # of pixels in I(:,:,3) with intensity value *g-1*
- or 1 vector-valued histogram, $h(g,1,b)$ where
  - $h(g,1,1)$ = # of pixels in I with red intensity value *g-1*
  - $h(g,1,2)$ = # of pixels in I with green intensity value *g-1*
  - $h(g,1,3)$ = # of pixels in I with blue intensity value *g-1*

Histogram Equalization

- Remap image *I(Mapping)*
- The transformation function is

$$s_k = \sum_{j=0}^{k} \frac{n_j}{n}$$

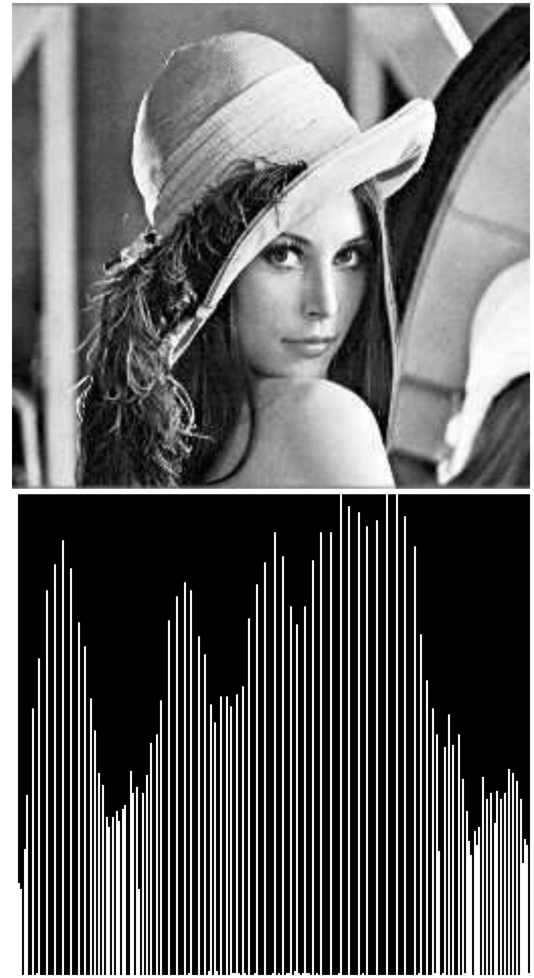
Where k=0,1,…L-1

Histogram Equalization

- >>g=histeq(f, nlev)
- *f* is the input image and *nlev* is the number of intensity levels specified for the output image
- If *nlev* is equal to *L* (total number of possible levels in the input image) then equalization is similar to input
- If *nlev* is less than *L*, then *histeq* attempts to distribute the levels so that they will approximate a flat histogram
- Default value for *nlev* is 64

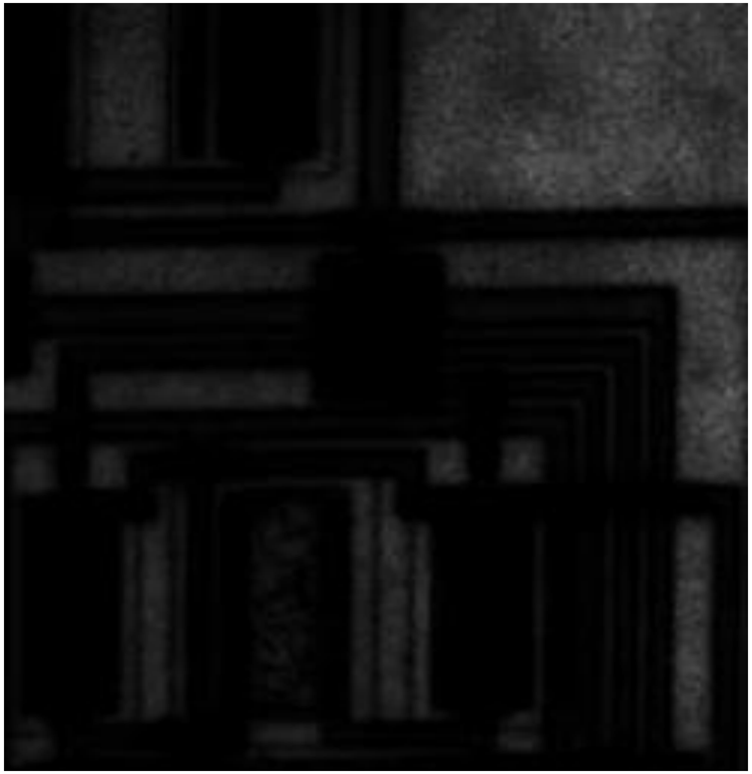
Histogram Equalization: Example

Original

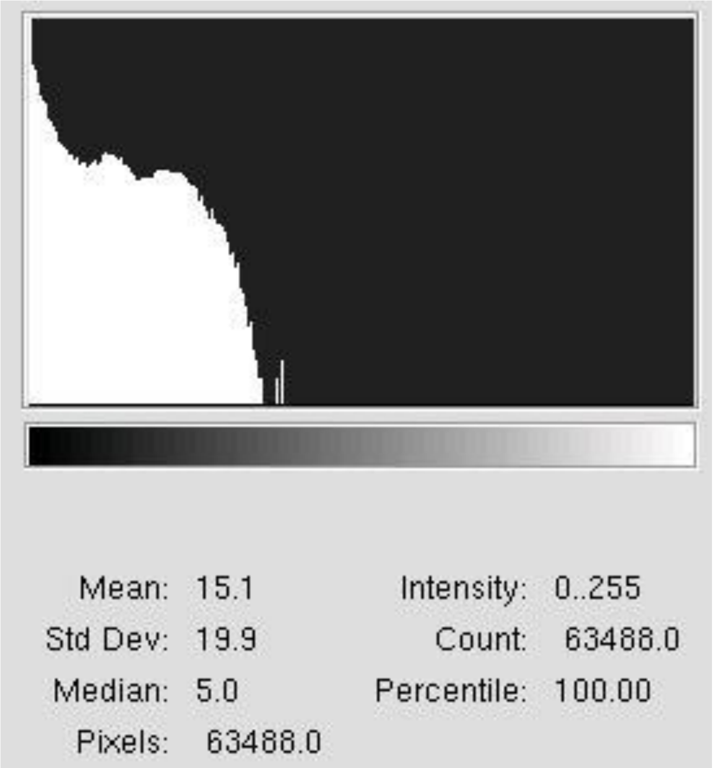


Equalized

HE -examples

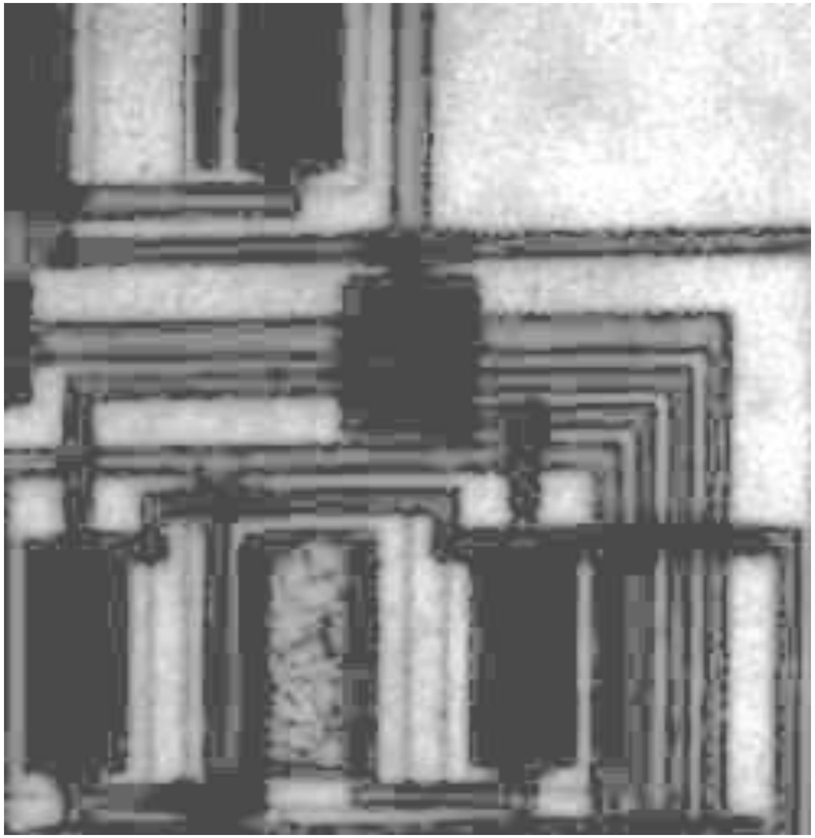


A dark image



Histogram

HE-examples



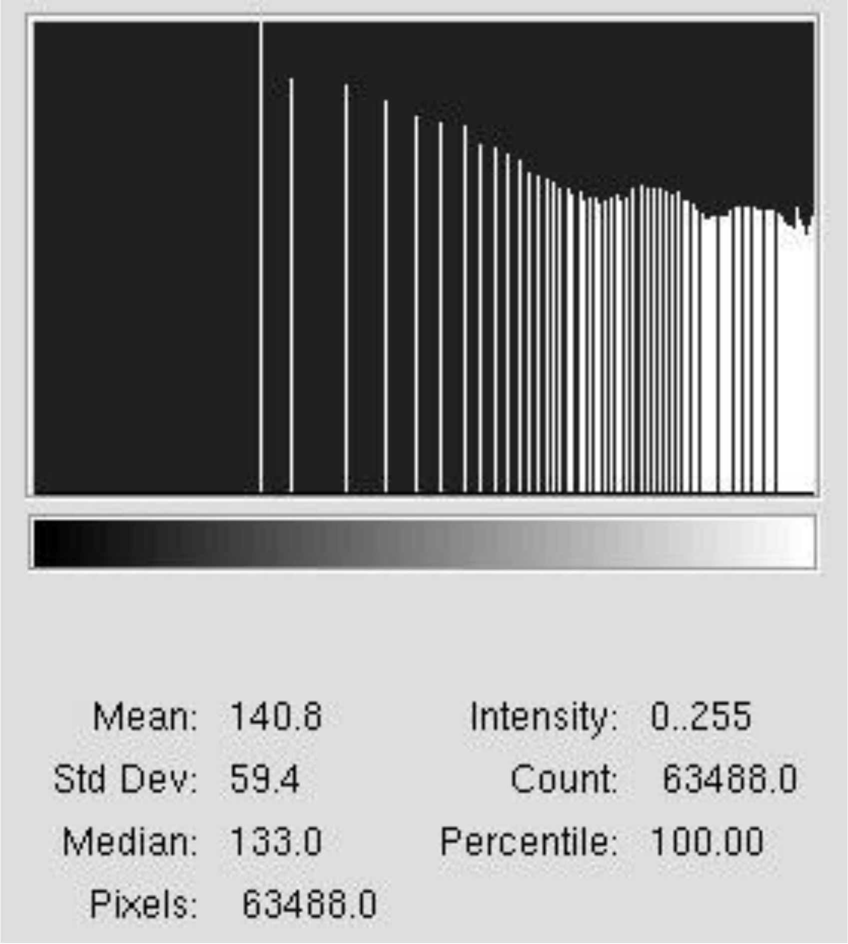

Mean: 140.8
Std Dev: 59.4
Median: 133.0
Pixels: 63488.0
Intensity: 0..255
Count: 63488.0
Percentile: 100.00

Image Subtraction

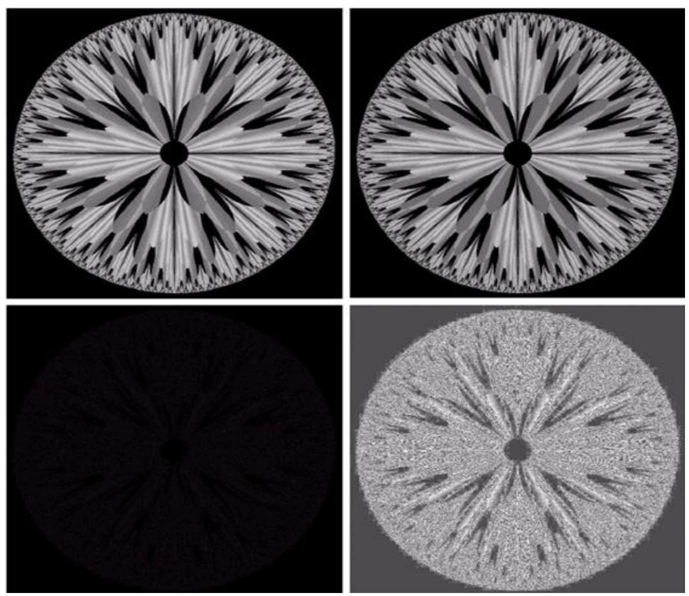
- A more interesting arithmetic operation is pixel-wise subtraction of two images.

$$g(x,y) = f(x,y) - h(x,y)$$

- Refer to the fractal image again.

original

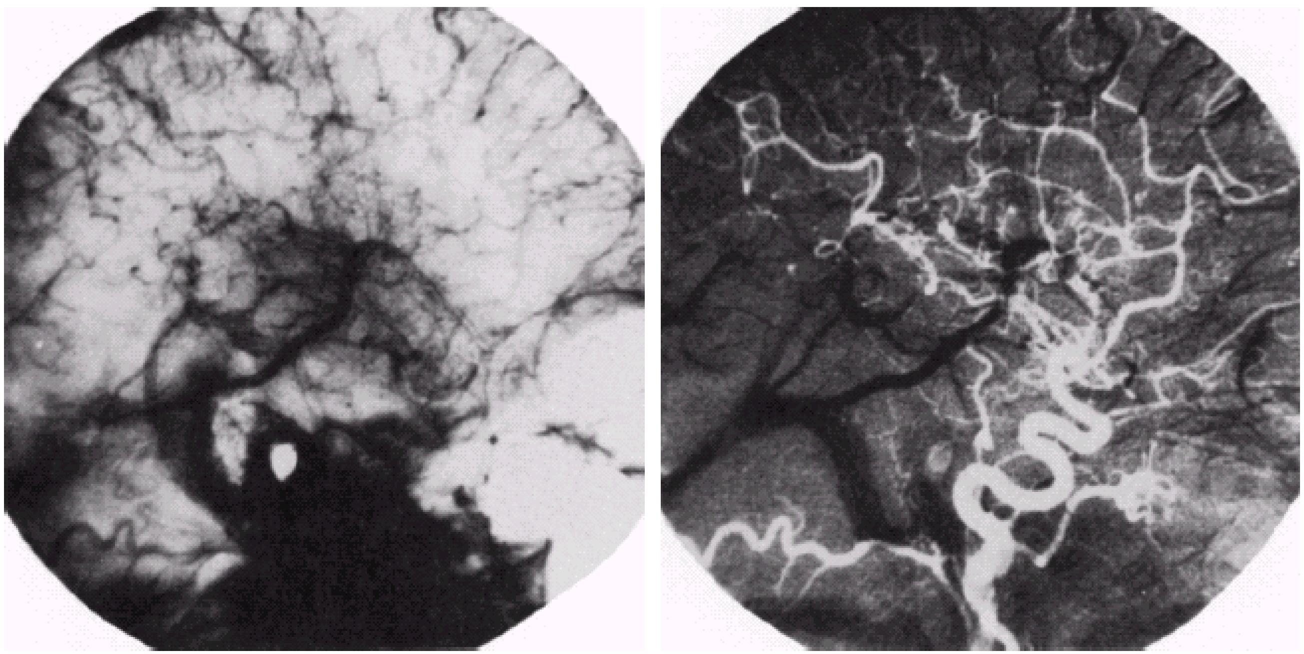
4 lower-order bit planes zeroed out



difference

contrast enhanced

Image Subtraction

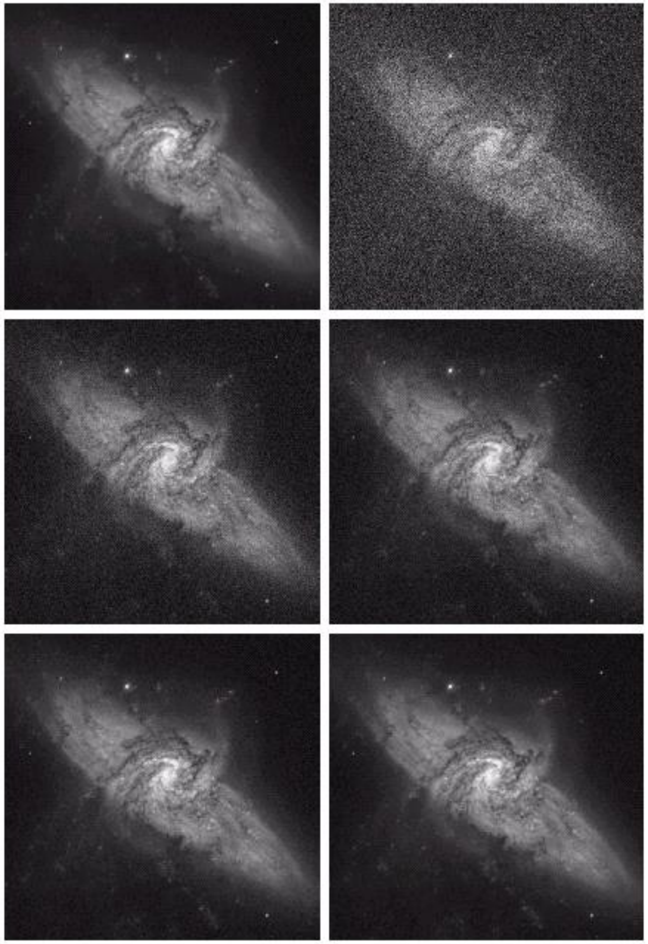


a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

Mask mode radiography

Image Averaging



- Same signal, but different noise realization.
- Averaging of many such images will enhance SNR.

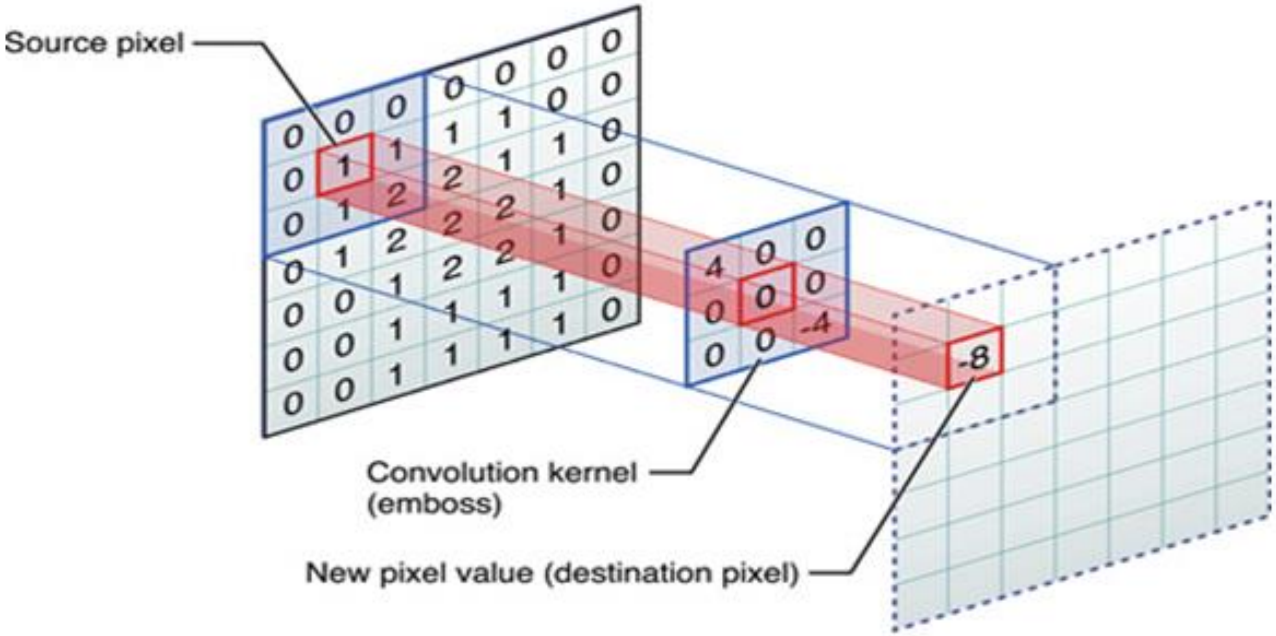
Basics of Spatial Filtering

- Defining a center point (x, y)
- Performing an operation that involves only the pixels in a predefined neighborhood about that center point and sub image.
- Sub image has the same dimensions as neighborhood.
- The sub image is called a *filter, mask, kernal, template or window*.
- Letting the result of that operation be the response of the process at that point
- Repeating the process for every point in the image

Process of Spatial Filtering

The process consists of,

- Moving the filter mask from point to point in an image.
- At each point (x, y), the response of the filter at that point is calculated.
- The response is sum of products of the filter coefficients and the corresponding image pixels in the area spanned by the filter mask.
- It is similar to frequency domain concept called convolution.
- So, the linear filtering process is often referred to as ***"Convolving a mask with an image"***



a	*b*	*c*
d	*e*	*e*
f	*g*	*h*

Source Pixels

r	*s*	*t*
u	*v*	*w*
x	*y*	*z*

Kernel

$$\textbf{New pixel} = v*e + z*a + y*b + x*c + w*d + u*e + t*f + s*g + r*h$$

Convolution

- Response R of an m × n mask at any point (x, y), is expressed as follows:

$$R = w_1 z_1 + w_2 z_2 + \cdots w_{mn} z_{mn}$$
$$= \sum_{i=1}^{mn} w_i z_i$$

- Example : For the 3 × 3 general mask, the response at any point (x,y) in the image is given by,

$$R = w_1 z_1 + w_2 z_2 + \cdots w_9 z_9$$
$$= \sum_{i=1}^{9} w_i z_i$$

Correlation and Convolution

- Correlation is the process of passing the mask w by the image array f
- Convolution is the same process, except that w is rotated by 180^0 prior to passing it by f

	f	w
Correlation	0 0 0 1 0 0 0 0	1 2 3 2 0
Convolution	0 0 0 1 0 0 0 0	0 2 3 2 1

Smoothing Filter

- Smoothing filters are used for blurring and for noise reduction.
- It is preprocessing step such as removal of small details, bridging of small gaps.
- Noise reduction can be done by blurring with linear and non-linear filtering.
- Pixel averaging in the spatial domain:
  - Each pixel in the output is a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 1.

8-Neighbor Mean filter

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

4-Neighbor Mean filter

0	1/5	0
1/5	1/5	1/5
0	1/5	0

1/5	0	1/5
0	1/5	0
1/5	0	1/5

Smoothing Filter

- These filters are also called averaging filter/low-pass filter.
- Idea behind Smoothing filters are straightforward.
- i.e. Replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask.
- The result is reduced "sharp" transitions in gray levels.
- Order statistics filter- Mean, median
- Reduces Salt & pepper noise

1/9 ×

1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

Sharpening Filter

- These filters are also called Differencing/High-pass/Laplacian filter.
- Based on first and second order derivatives.
- Highlight fine details in an image.
- Applications are electronic printing, medical imaging, industrial inspections, and autonomous guidance in military systems.
- Pixel-differenced in the spatial domain:
  - Each pixel in the output is a difference between itself and a weighted average of its neighbors.
  - Is a convolution whose weight matrix sums to 0.

Blurring vs. Sharpening

- Blurring/smooth is done in spatial domain by pixel averaging in a neighbors, it is a process of integration.

- Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation.

Derivative operator

- The strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied.

- Image differentiation
  – enhances edges and other discontinuities (noise)
  – deemphasizes area with slowly varying gray-level values.

High-Pass Filters

Laplacian Operator

0	1	0
1	-4	1
0	1	0

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

1	1	1
1	-8	1
1	1	1

Sobel Operator

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

0	1	2
-1	0	1
-2	-1	0

-2	-1	0
-1	0	1
0	1	2

References

- https://www.slideshare.net/CristinaPrezBenito/simultaneous-smoothing-and-sharpening-of-color-images.
- https://www.javatpoint.com/digital-image-processing-tutorial.
- https://www.tutorialspoint.com/dip/index.html.

TEXT BOOKS

1) Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Second Edition, PHI/Pearson Education.
2) Alexander M., Abid K., "OpenCV-Python Tutorials", 2017.

REFERENCE BOOKS

1) B. Chanda, D. Dutta Majumder, "Digital Image Processing and Analysis", PHI, 2003.
2) Nick Efford, "Digital Image Processing a practical introducing using Java", Pearson Education, 2004.