University of Kurdistan

Digital Image Processing (DIP)

Lecture 3: Image Enhancement in the Spatial Domain

Instructor:

Kaveh Mollazade, Ph.D.

Department of Biosystems Engineering, Faculty of Agriculture, University of Kurdistan, Sanandaj, IRAN.

Content

• In this lecture, we will look at image enhancement techniques working in the spatial domain:

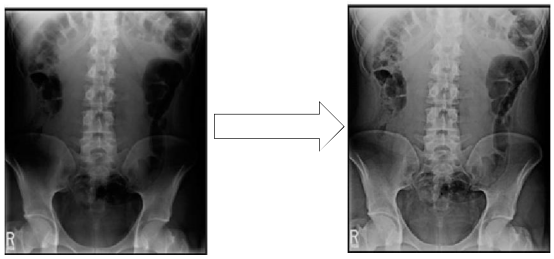
- What is image enhancement?
- Different kinds of image enhancement
- Histogram processing
- Point processing
- Neighborhood operations
- Spatial filtering operations

A note about grey levels

- So far when we have spoken about image grey level values we have said they are in the range [0, 255].
  - Where 0 is black and 255 is white.
- There is no reason why we have to use this range.
  - The range [0, 255] stems from display technologies.
- For many of the image processing operations in this lecture, grey levels are assumed to be given in the range [0.0, 1.0].

What is image enhancement?

- Image enhancement is the process of making images more useful.



- The reasons for doing this include:
  - Highlighting interesting detail in images.
  - Removing noise from images.
  - Making images more visually appealing.

Image enhancement examples

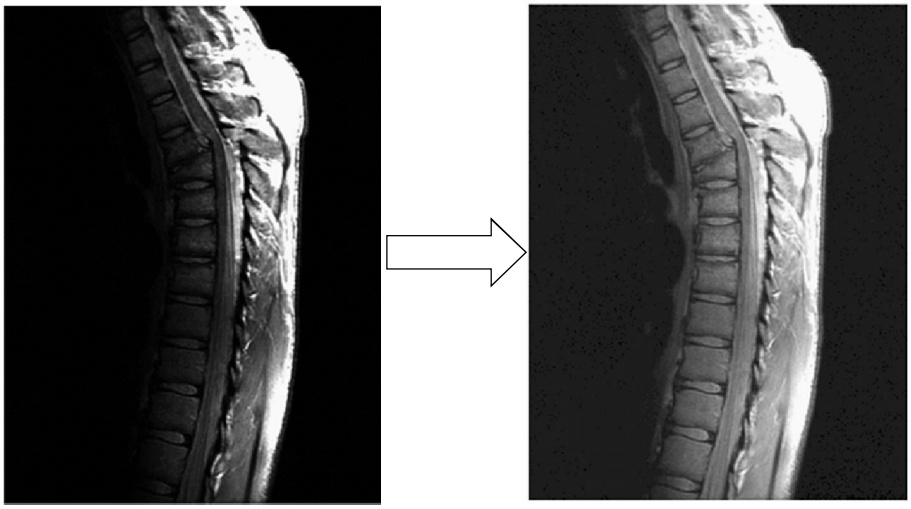
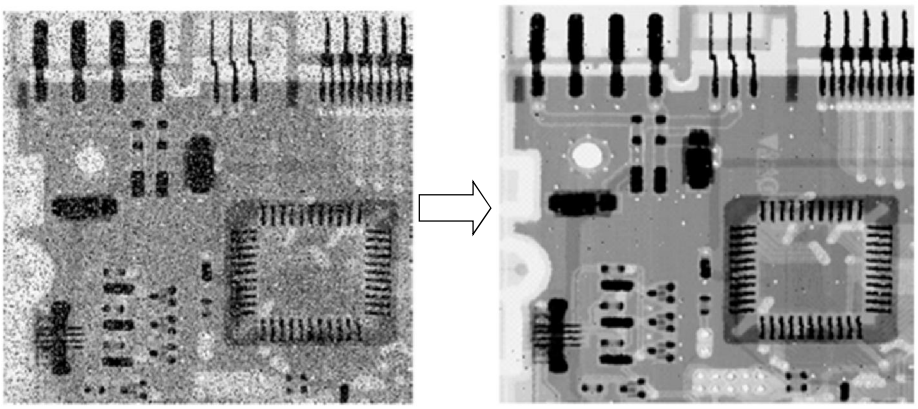


Image enhancement examples (cont …)

Image enhancement examples (cont …)



Spatial & frequency domains

- There are two broad categories of image enhancement techniques:

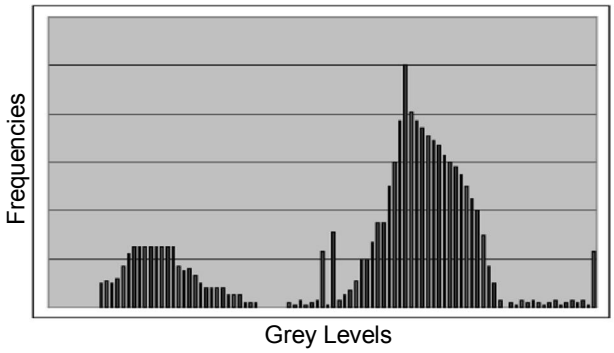
1. Spatial domain techniques:
   - Direct manipulation of image pixels.
2. Frequency domain techniques:
   - Manipulation of Fourier transform or wavelet transform of an image.

- For the moment we will concentrate on techniques that operate in the spatial domain.

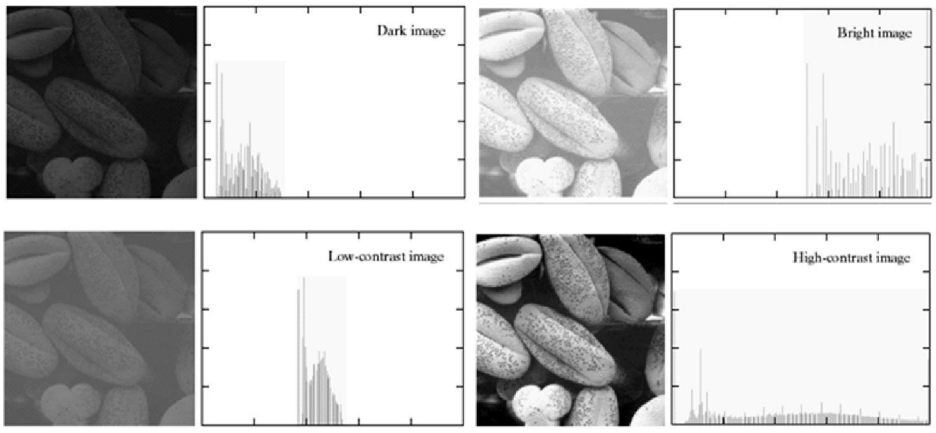
Image histograms

• The histogram of an image shows us the distribution of grey levels in the image.

• Massively useful in image processing, especially in segmentation.



Histogram examples



• Note that the high contrast image has the most evenly spaced histogram.

Normalized histogram

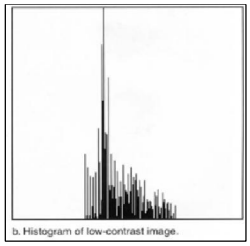
• Histogram: is the discrete function $h(r_k)=n_k$, where r_k is the k_{th} gray level in the range of [0, L-1] and n_k is the number of pixels having gray level r_k.

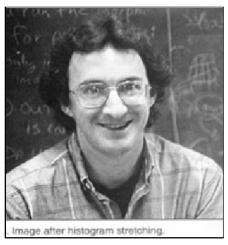
• Normalized histogram: is $p(r_k)=n_k/n$, for k=0,1,…,L-1 and $p(r_k)$ can be considered to give an estimate of the probability of occurrence of gray level r_k. n is the total number of pixels in image.

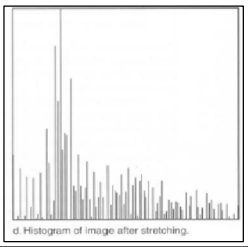
Contrast stretching

• We can fix images that have poor contrast by applying a pretty simple contrast specification.


Low-contrast image.


b. Histogram of low-contrast image.


c. Image after histogram stretching.


d. Histogram of image after stretching.

Histogram equalization

• Spreading out the frequencies in an image (or equalizing the image) is a simple way to improve dark or washed out images.

• The formula for histogram equalization:

r_k: input intensity

s_k: processed intensity

k: the intensity range (e.g. 0 – 255)

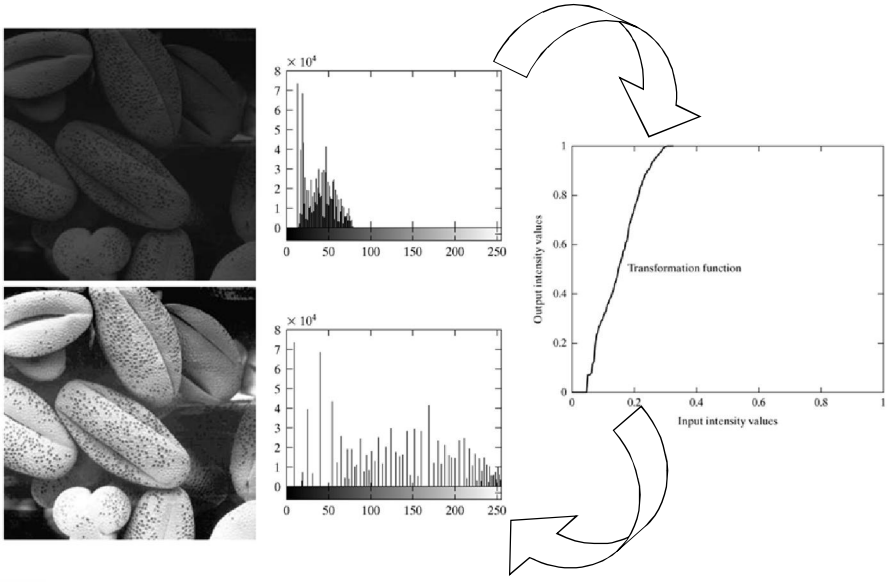
L: the number of intensity levels

n_j: the frequency of intensity j

n: the sum of all frequencies

$$s_k = T(r_k)$$
$$= (L-1)\sum_{j=0}^{k} p_r(r_j)$$
$$= (L-1)\sum_{j=0}^{k} \frac{n_j}{n}$$

Equalization transformation function



A numerical example for histogram equalization

Intensity distribution and histogram values for a 3-bit 64 ×64 digital image

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7p_r(r_0) + 7p_r(r_1) = 3.08$$

$$s_2 = 4.55$$

$$s_3 = 5.67$$

$$s_4 = 6.23$$

$$s_5 = 6.65$$

$$s_6 = 6.86$$

$$s_7 = 7.00$$

A numerical example for histogram equalization

$$s_0 = 1.33 \rightarrow 1$$

$$s_1 = 3.08 \rightarrow 3$$

$$s_2 = 4.55 \rightarrow 5$$

$$s_3 = 5.67 \rightarrow 6$$

$$s_4 = 6.23 \rightarrow 6$$

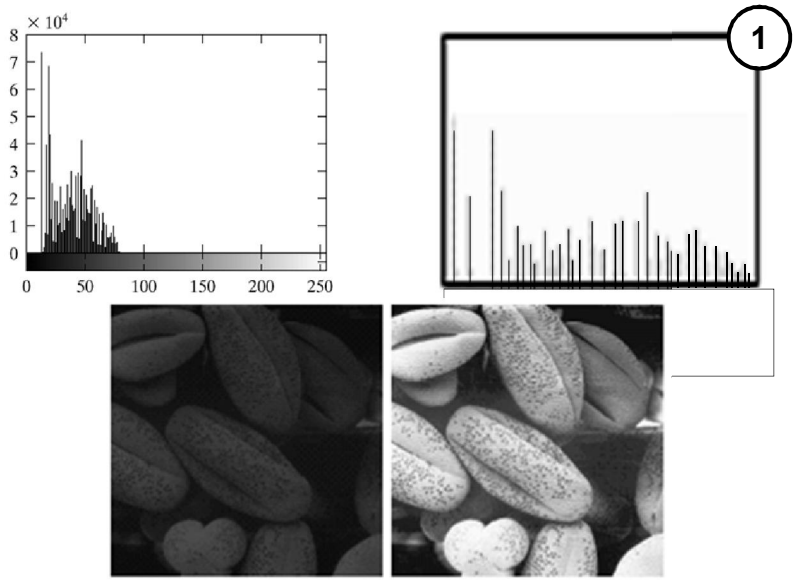
$$s_5 = 6.65 \rightarrow 7$$

$$s_6 = 6.86 \rightarrow 7$$

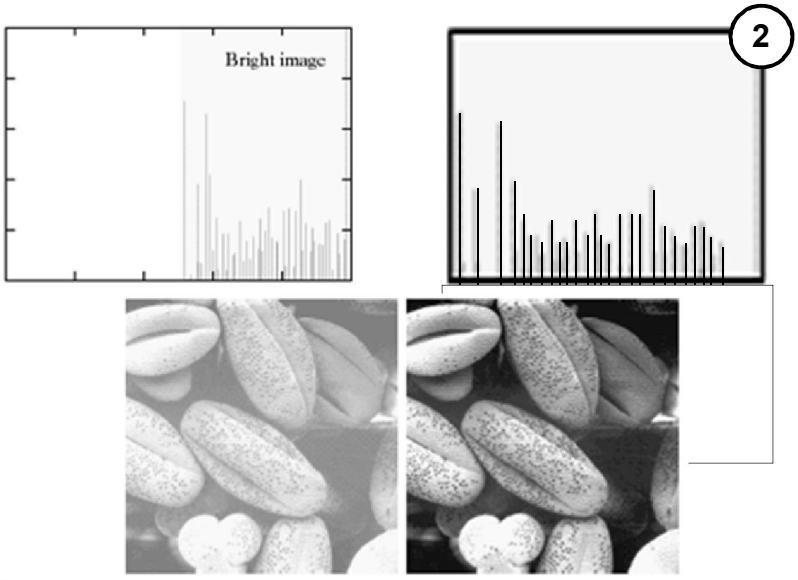
$$s_7 = 7.00 \rightarrow 7$$

Original histogram | Transfer function | Equalized histogram

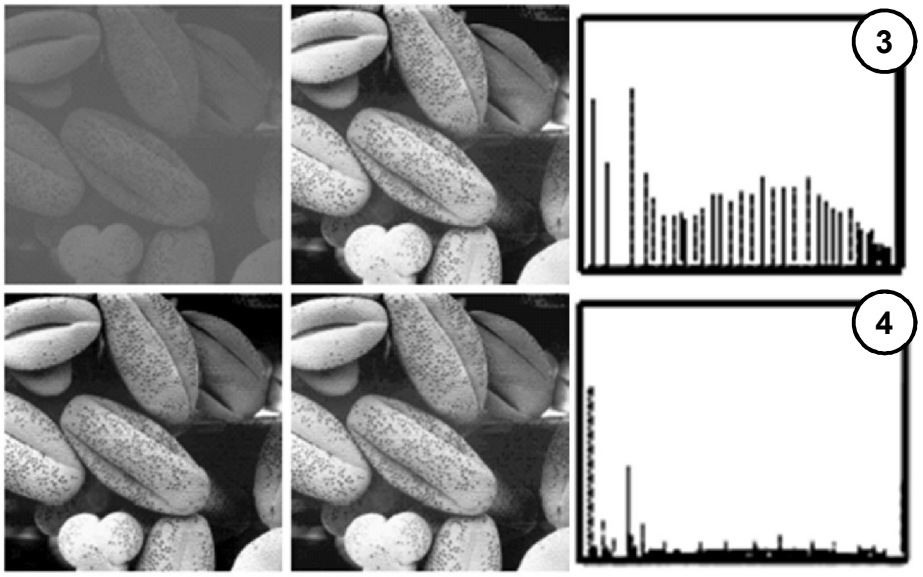
Equalization examples (cont …)



Equalization examples (cont …)

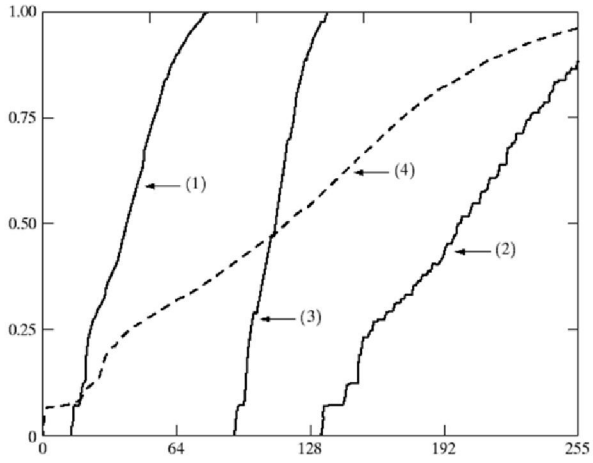


Equalization examples (cont …)



Equalization examples

- The functions used to equalize the images in the previous examples:



Point processing

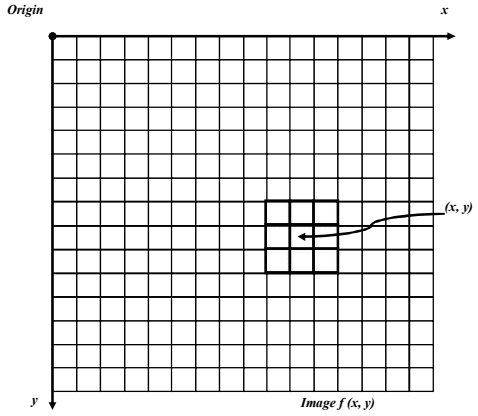
• In the following slides, we will look at image enhancement point processing techniques:

- What is point processing?
- Negative images
- Thresholding
- Logarithmic transformation
- Power law transforms
- Grey level slicing
- Bit plane slicing

Basic spatial domain image enhancement

Most spatial domain enhancement operations can be reduced to the form g(x, y) = T[f (x, y)].

where f(x,y) is the input image, g(x,y) is the processed image and T is some operator defined over some neighborhood of (x, y).



Point processing

The simplest spatial domain operations occur when the neighborhood is simply the pixel itself.

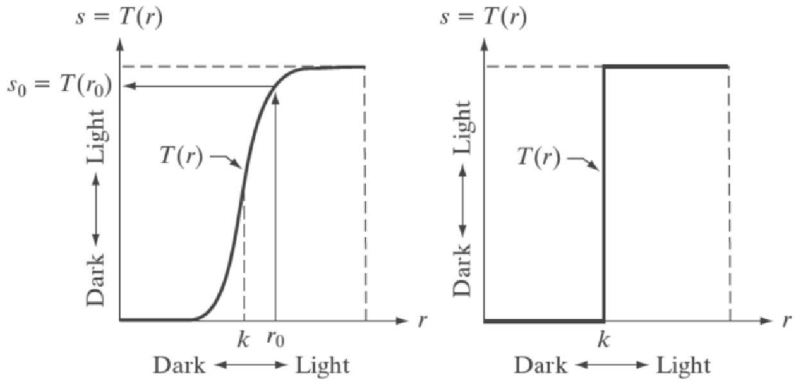
In this case T is referred to as a grey level transformation function or a point processing operation.

Point processing operations take the form

$$s = T(r)$$

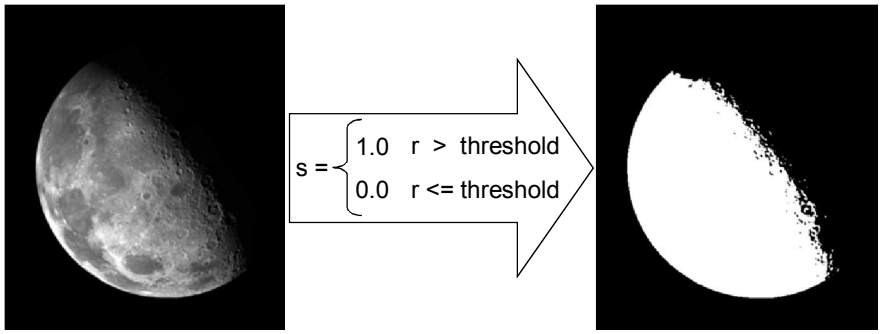
where *s* refers to the processed image pixel value and *r* refers to the original image pixel value.

Intensity transformation

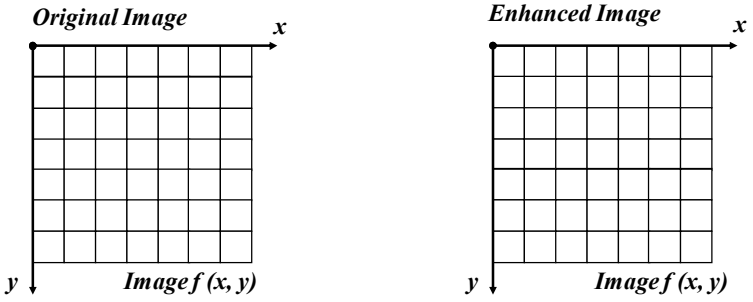


Point processing example: Thresholding

Thresholding transformations are particularly useful for segmentation in which we want to isolate an object of interest from a background.



Point processing example: Thresholding (cont …)



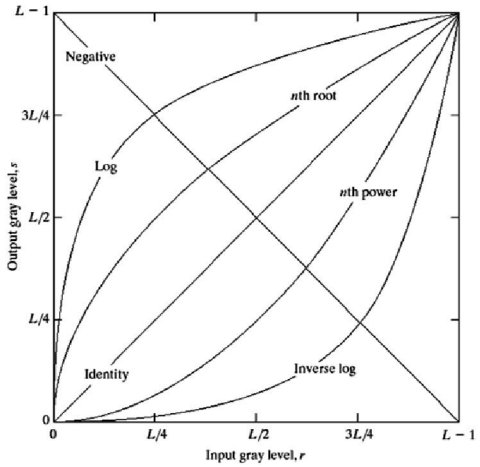
$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Basic grey level transformations

• There are many different kinds of grey level transformations.

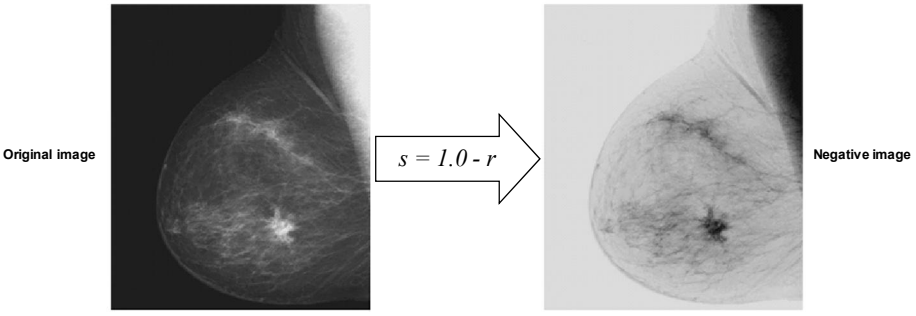
• Three of the most common are :

- Linear
  - Negative/Identity
- Logarithmic
  - Log/Inverse log
- Power law
  - n^{th} power/n^{th} root



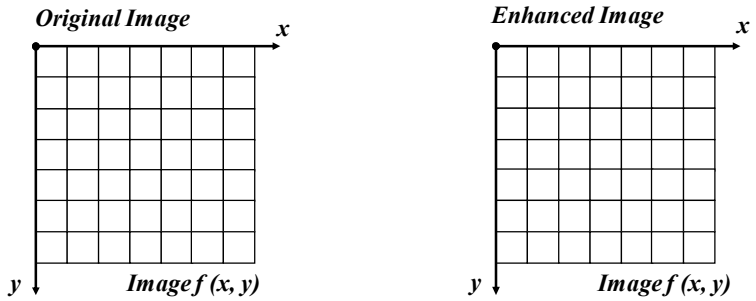
Negative images

Negative images are useful for enhancing white or grey detail embedded in dark regions of an image.



Note how much clearer the tissue is in the negative image of the mammogram.

Negative images (cont …)



$$s = intensity_{max} - r$$

Logarithmic transformations

• The general form of the log transformation is:

$$s = c * log(1 + r)$$

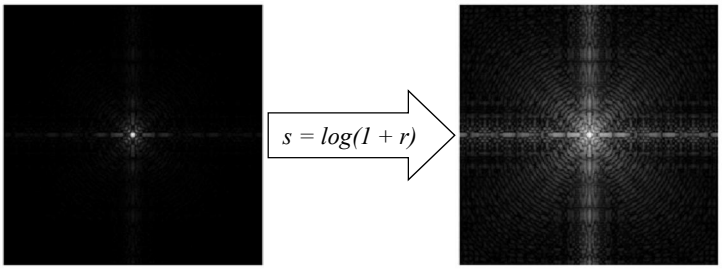
• The log transformation maps a narrow range of low input grey level values into a wider range of output values.

• The inverse log transformation performs the opposite transformation.

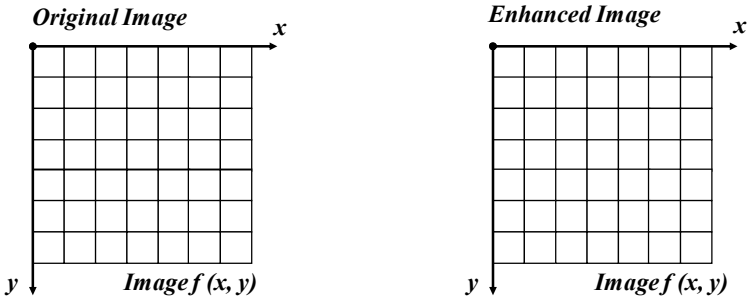
Logarithmic transformations (cont…)

• Log functions are particularly useful when the input grey level values may have an extremely large range of values.

In the following example the Fourier transform of an image is put through a log transform to reveal more detail.



Logarithmic transformations (cont…)



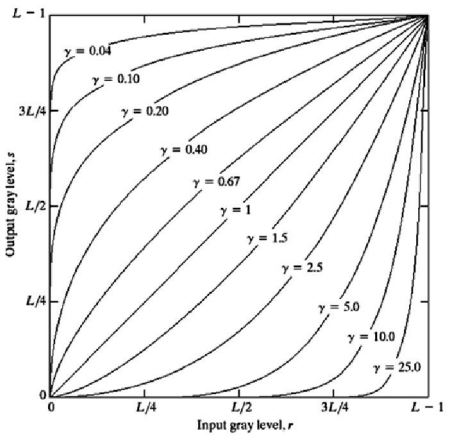
$$s = log(1 + r)$$

We usually set c to 1.
Grey levels must be in the range [0.0, 1.0].

Power law transformations

• Power law transformations have the following form

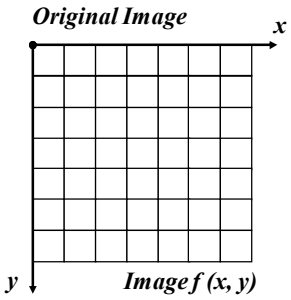
$$s = c * r^{\gamma}$$

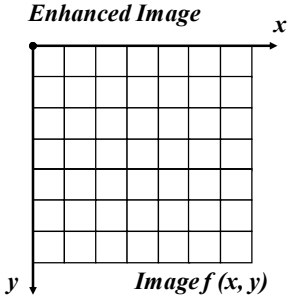


• Map a narrow range of dark input values into a wider range of output values or vice versa.

Varying γ gives a whole family of curves.

Power law transformations (cont…)





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

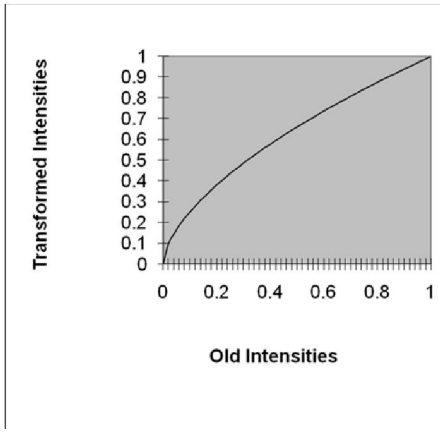
We usually set c to 1

Grey levels must be in the range [0.0, 1.0].

Power law example 1

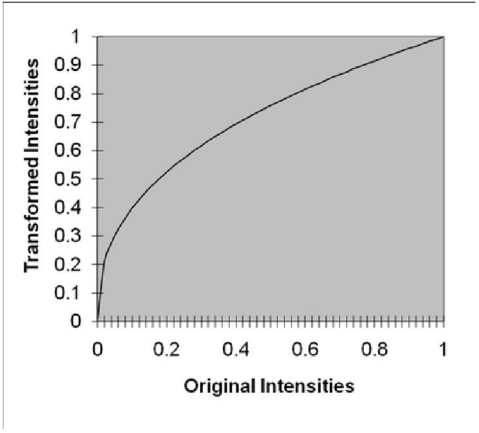
Power law example 1 (cont …)

$$\gamma = 0.6$$



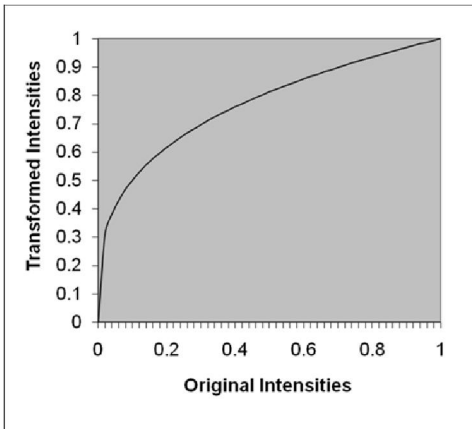
Power law example 1 (cont …)

$\gamma = 0.4$



Power law example 1 (cont …)

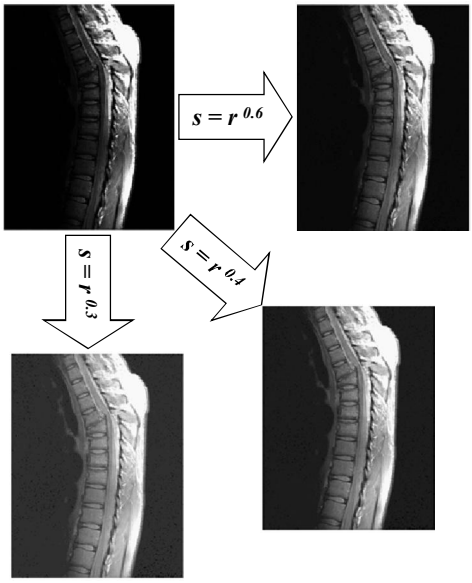
$\gamma = 0.3$



Power law example 1 (cont …)

The images to the right show a magnetic resonance (MR) image of a fractured human spine.

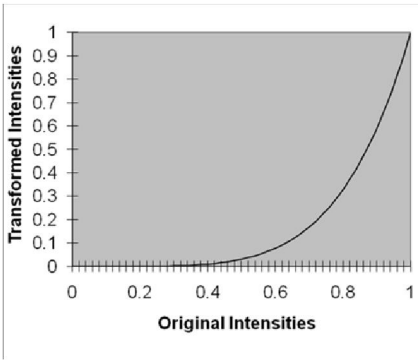
Different curves highlight different detail.



Power law example 2

Power law example 2 (cont …)

$$\gamma = 5.0$$

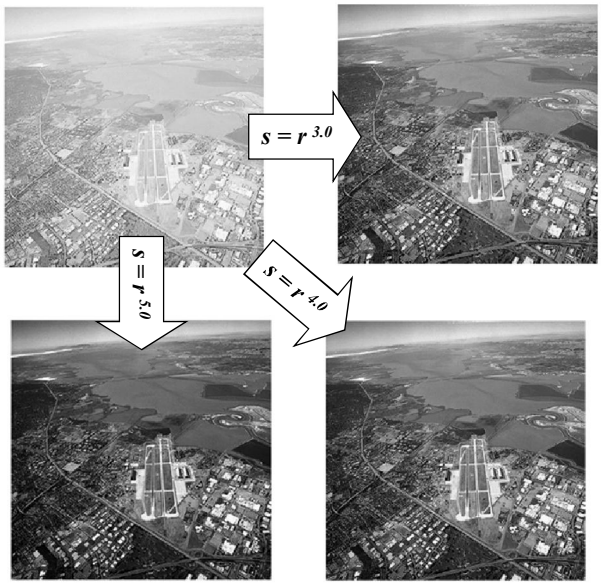


Power law example 2 (cont …)

An aerial photo of a runway is shown.

This time power law transforms are used to darken the image.

Different curves highlight different detail.

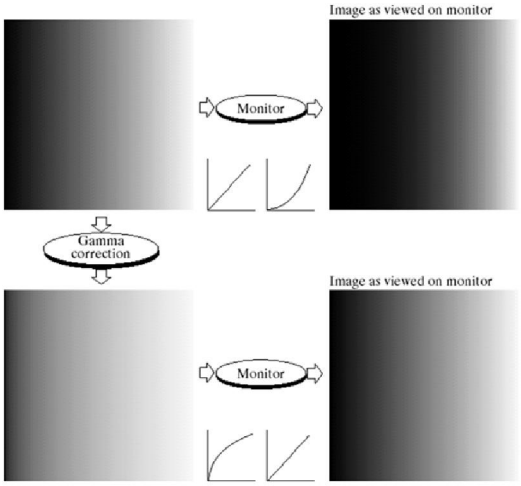


Gamma correction

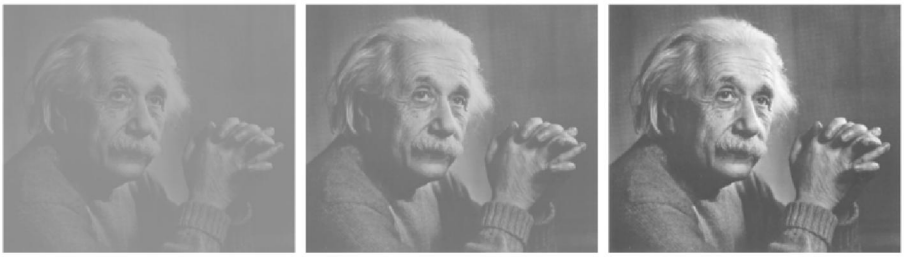
• Many of you might be familiar with gamma correction of computer monitors.

• Problem is that display devices do not respond linearly to different intensities.

• It can be corrected using a power transform.



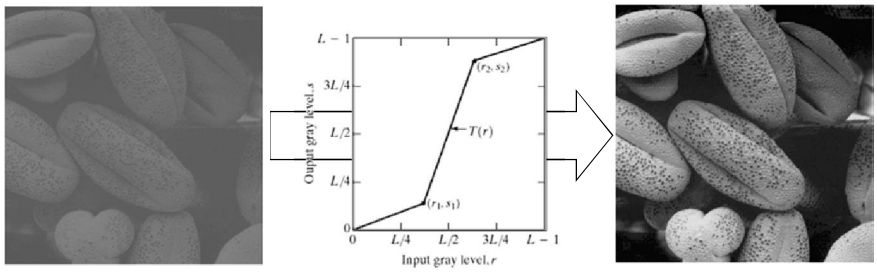
More contrast issues



Piecewise linear transformation functions

• Rather than using a well defined mathematical function we can use arbitrary user-defined transforms.

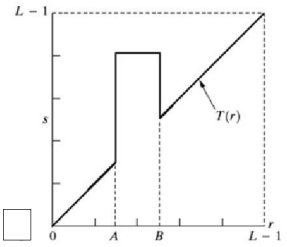
• The images below show a contrast stretching linear transform to add contrast to a poor quality image.

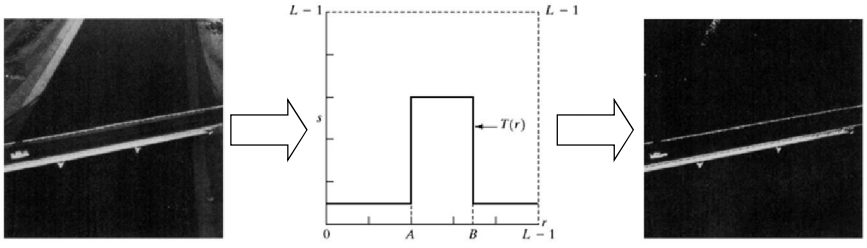


Gray level slicing

Highlights a specific range of grey levels

- Similar to thresholding
- Other levels can be suppressed or maintained
- Useful for highlighting features in an image



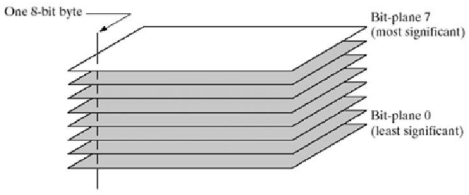


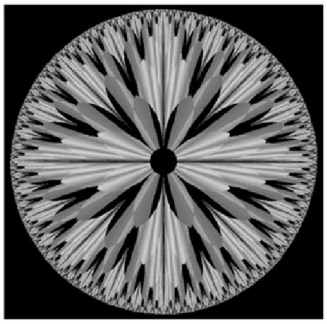
Bit plane slicing

• Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image.

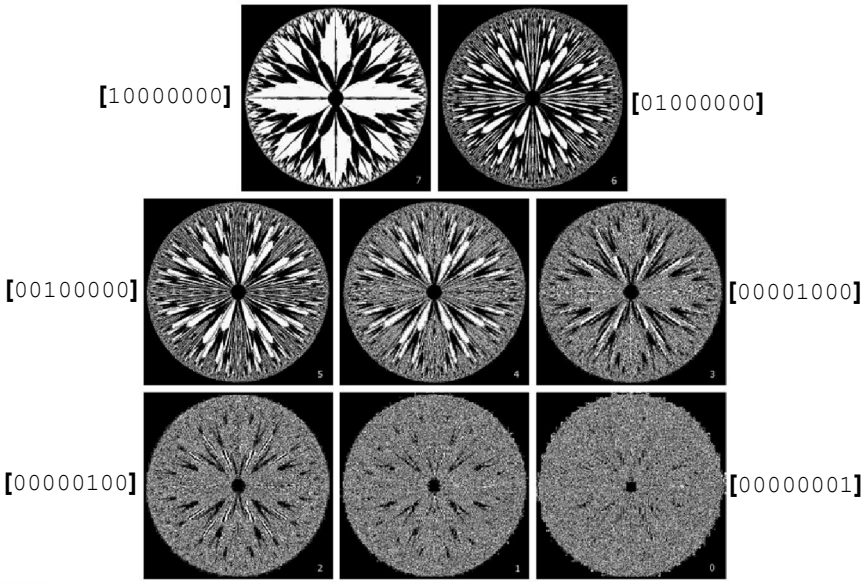
– Higher-order bits usually contain most of the significant visual information.

– Lower-order bits contain subtle details.

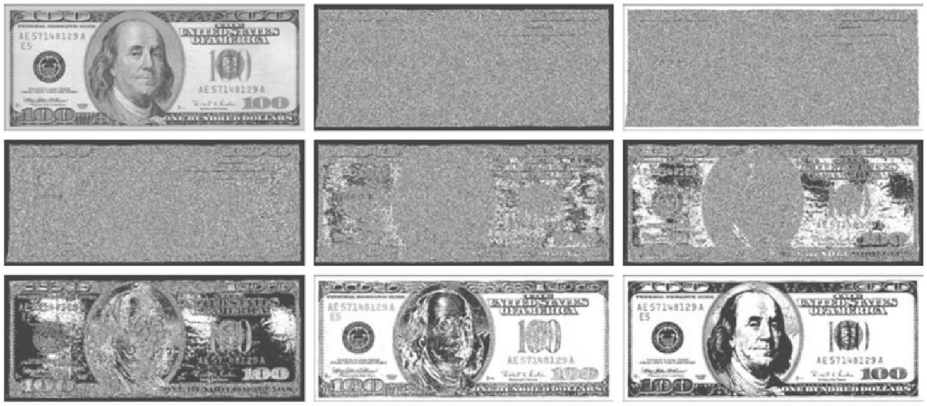




Bit plane slicing (cont ..)



Bit plane slicing (cont ..)



a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

Bit plane slicing (cont ..)

Reconstructed image using only bit planes 8 and 7

Reconstructed image using only bit planes 8, 7 and 6

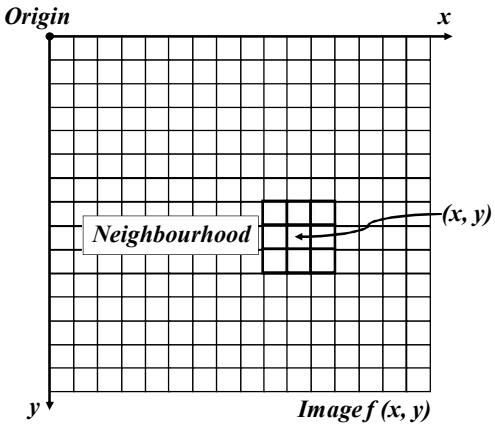
Reconstructed image using only bit planes 8, 7, 6 and 5

Spatial filtering

- In the following slides, will look at spatial filtering techniques:
  - Neighbourhood operations
  - What is spatial filtering?
  - Smoothing operations
  - What happens at the edges?
  - Correlation and convolution
  - Sharpening filters (1st derivative filters and 2nd derivative filters)
  - Combining filtering techniques

Neighborhood operations

- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations.
- Neighbourhoods are mostly a rectangle around a central pixel.
- Any size rectangle and any shape filter possible.

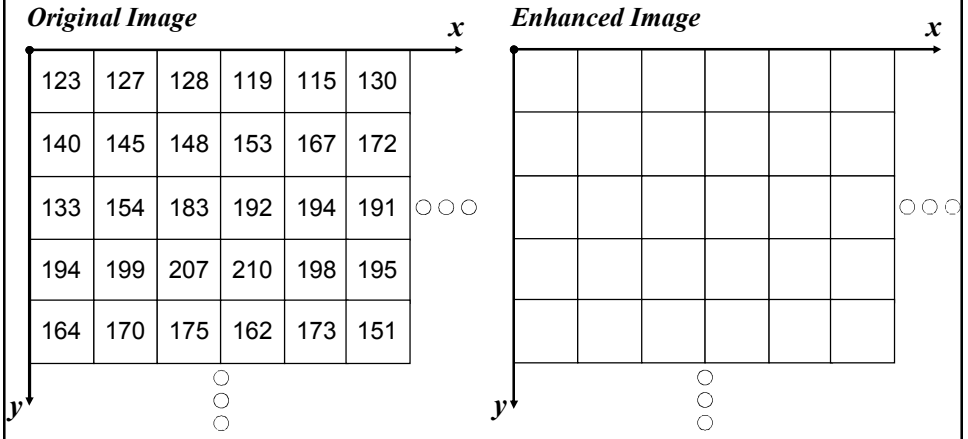


Simple neighborhood operations

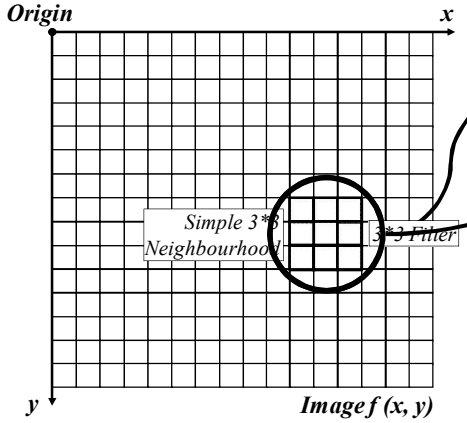
Some simple neighbourhood operations include:

- **Min:** Set the pixel value to the minimum in the neighbourhood.
- **Max:** Set the pixel value to the maximum in the neighbourhood.
- **Median:** The median value of a set of numbers is the midpoint value in that set (e.g. from the set [1, 7, 15, 18, 24] 15 is the median). Sometimes the median works better than the average.

Example: simple neighborhood operations



The spatial filtering process



a	b	c
d	e	f
g	h	i

Original Image Pixels

$*$

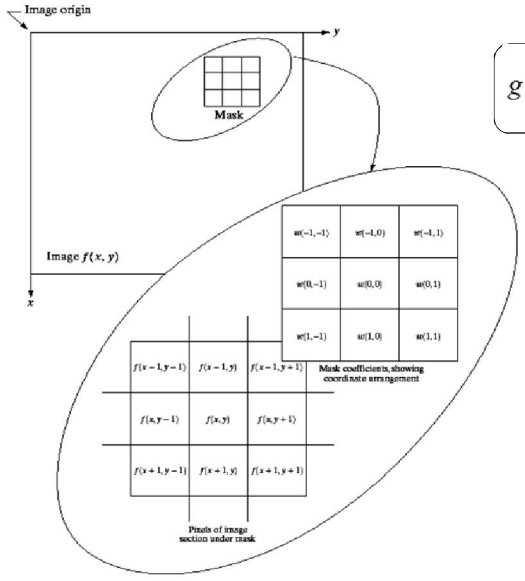
r	s	t
u	v	w
x	y	z

Filter

$$e_{processed} = v*e + r*a + s*b + t*c + u*d + w*f + x*g + y*h + z*i$$

The above is repeated for every pixel in the original image to generate the filtered image.

Spatial filtering: equation form



$$g(x,y) = \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- Filtering can be given in equation form as shown above.
- Notations are based on the image shown to the left.

Smoothing spatial filters

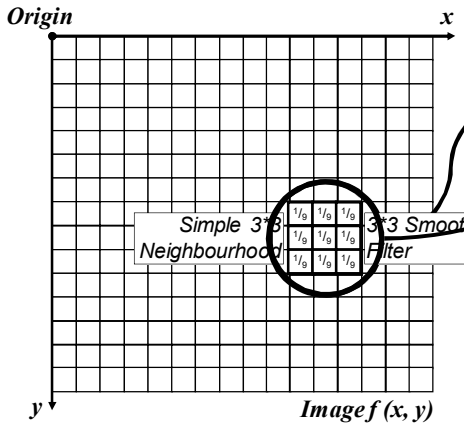
One of the simplest spatial filtering operations we can perform is a smoothing operation.

- Simply average all of the pixels in a neighbourhood around a central value.
- Especially useful in removing noise from images.
- Also useful for highlighting gross detail.

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Simple averaging filter

Smoothing spatial filtering



104	*100*	*108*
99	*106*	*98*
95	*90*	*85*

Original Image Pixels

$*$

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

Filter

$$e = {}^1/_9*106 + {}^1/_9*104 + {}^1/_9*100 + {}^1/_9*108 + {}^1/_9*99 + {}^1/_9*98 + {}^1/_9*95 + {}^1/_9*90 + {}^1/_9*85$$
$$= 98.3333$$

The above is repeated for every pixel in the original image to generate the smoothed image.

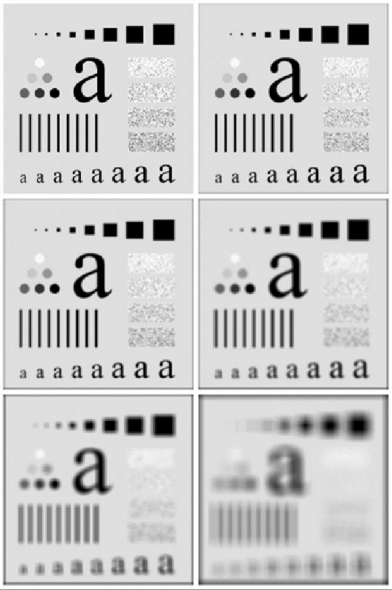
Image smoothing example

• The image at the top left is an original image of size 500*500 pixels.

• The subsequent images show the image after filtering with an averaging filter of increasing sizes.

– 3, 5, 9, 15 and 35.

• Notice how detail begins to disappear.



Weighted smoothing filters

More effective smoothing filters can be generated by allowing different pixels in the neighborhood different weights in the averaging function.

– Pixels closer to the central pixel are more important.

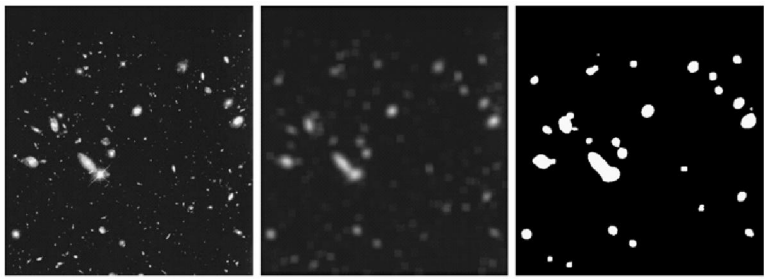
– Often referred to as a *weighted averaging.*

$^1/_{16}$	$^2/_{16}$	$^1/_{16}$
$^2/_{16}$	$^4/_{16}$	$^2/_{16}$
$^1/_{16}$	$^2/_{16}$	$^1/_{16}$

Weighted averaging filter

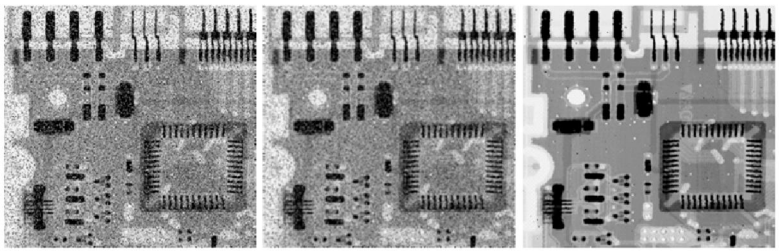
Another smoothing example

By smoothing the original image we get rid of lots of the finer detail which leaves only the gross features for thresholding.



Original Image | Smoothed Image | Thresholded Image

Averaging filter vs. median filter example

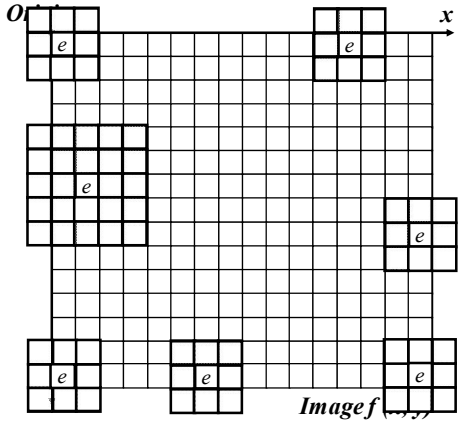


Original Image With Noise | Image After Averaging Filter | Image After Median Filter

- Filtering is often used to remove noise from images.
- Sometimes a median filter works better than an averaging filter.

Strange things happen at the edges!

At the edges of an image we are missing pixels to form a neighbourhood.

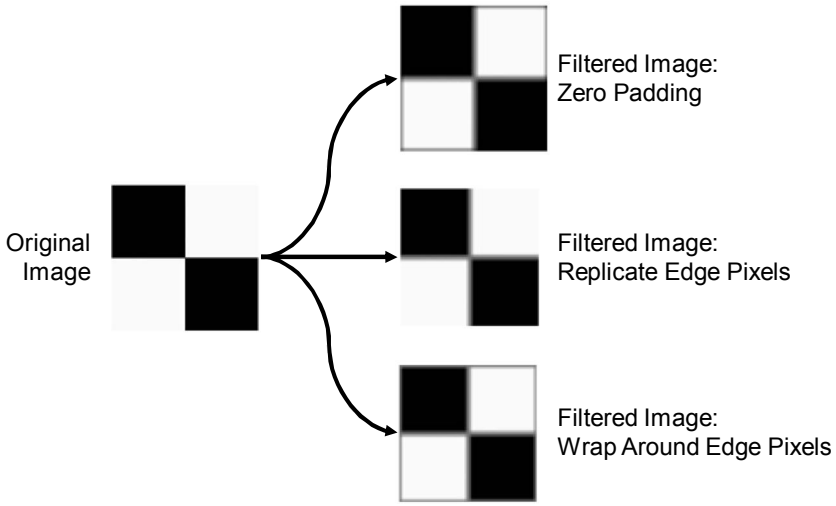


Strange things happen at the edges! (cont …)

There are a few approaches to dealing with missing edge pixels:

- Omit missing pixels
  - Only works with some filters
  - Can add extra code and slow down processing
- Pad the image
  - Typically with either all white or all black pixels
- Replicate border pixels
- Truncate the image
- Allow pixels *wrap around* the image
  - Can cause some strange image artefacts

Strange things happen at the edges! (cont …)



Correlation & convolution

The filtering we have been talking about so far is referred to as *correlation* with the filter itself referred to as the *correlation kernel.*

Convolution is a similar operation, with just one subtle difference.

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

Original Image Pixels

$*$

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

Filter

$$e_{processed} = v*e + z*a + y*b + x*c + w*d + u*e + t*f + s*g + r*h$$

For symmetric filters it makes no difference.

Sharpening spatial filters

• Previously we have looked at smoothing filters which remove fine detail.

• *Sharpening spatial filters* seek to highlight fine details:

  – Remove blurring from images

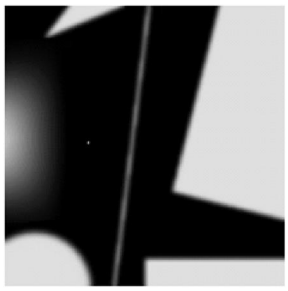
  – Highlight edges

• Sharpening filters are based on *spatial differentiation.*

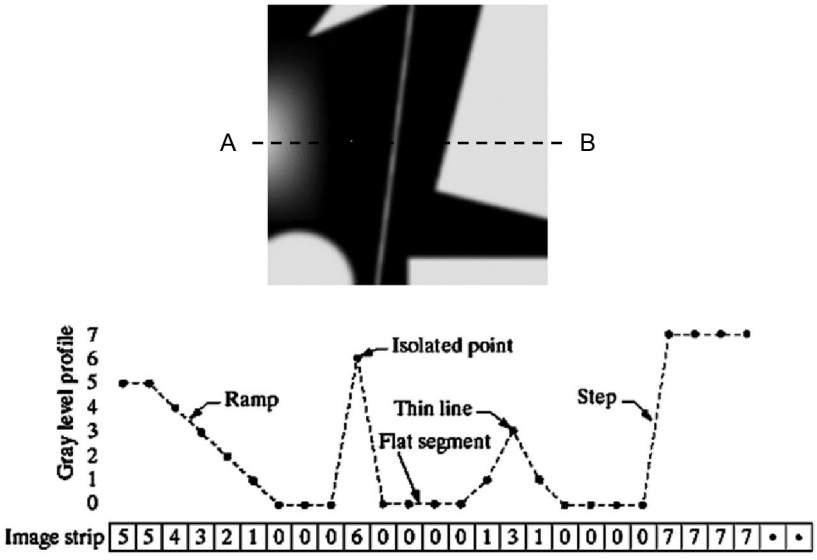
Spatial differentiation

• Differentiation measures the *rate of change* of a function.

• Let's consider a simple 1 dimensional example.



Spatial differentiation



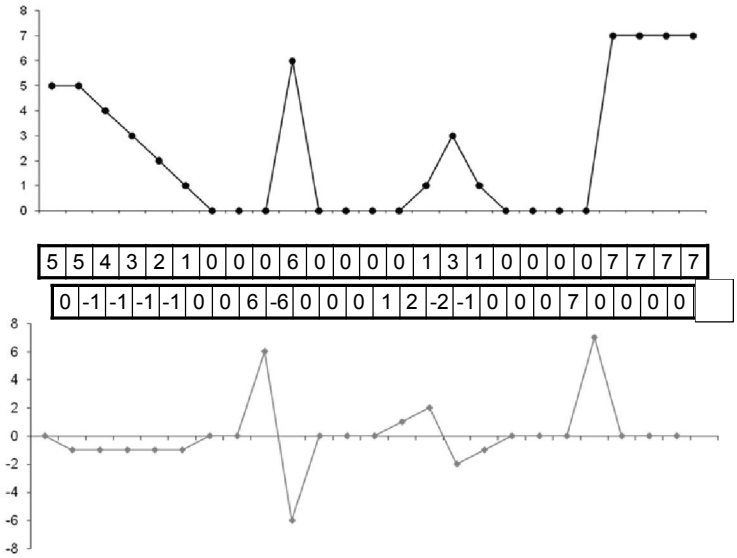
1st derivative

The formula for the 1st derivative of a function is as follows:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

It's just the difference between subsequent values and measures the rate of change of the function.

1st derivative (cont …)



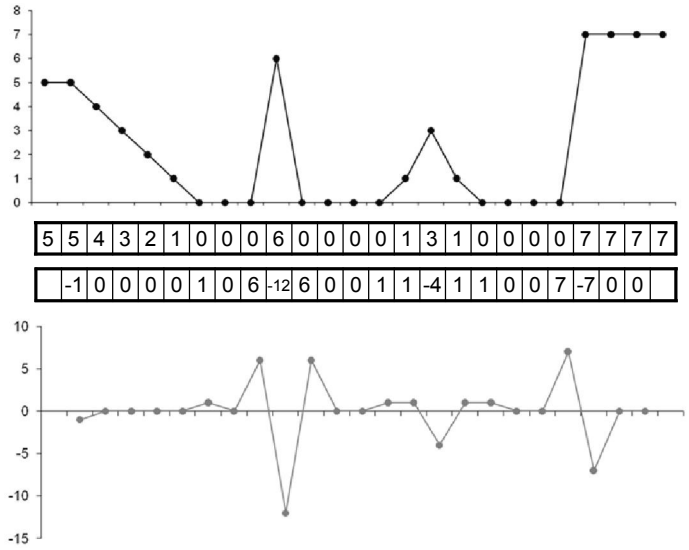
2nd derivative

The formula for the 2nd derivative of a function is as follows:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Simply takes into account the values both before and after the current value.

2nd derivative (cont…)



Using second derivatives for image enhancement

- The 2nd derivative is more useful for image enhancement than the 1st derivative
  - Stronger response to fine detail
  - Simpler implementation
  - We will come back to the 1st order derivative later on

- The first sharpening filter we will look at is the *Laplacian*
  - Isotropic
  - One of the simplest sharpening filters
  - We will look at a digital implementation

The Laplacian

The Laplacian is defined as follows:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

where the partial 2nd order derivative in the x direction is defined as follows:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

and in the y direction as follows:

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

The Laplacian (cont …)

So, the Laplacian can be given as follows:

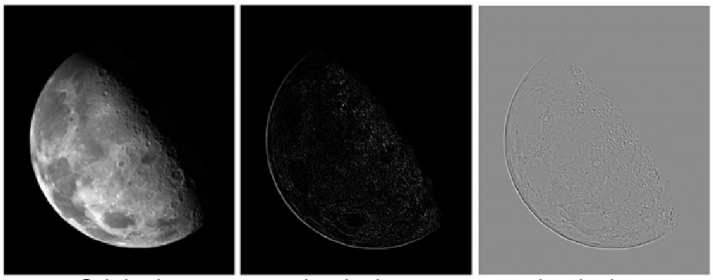
$$\begin{aligned}\nabla^2 f = [&f(x+1, y) + f(x-1, y) \\ &+ f(x, y+1) + f(x, y-1)] \\ &- 4f(x, y)\end{aligned}$$

We can easily build a filter based on this:

0	1	0
1	-4	1
0	1	0

The Laplacian (cont …)

Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities.



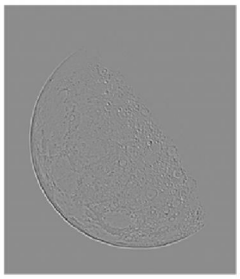
Original Image | Laplacian Filtered Image | Laplacian Filtered Image Scaled for Display

But that is not very enhanced!

• The result of a Laplacian filtering is not an enhanced image.

• We have to do more work in order to get our final image.

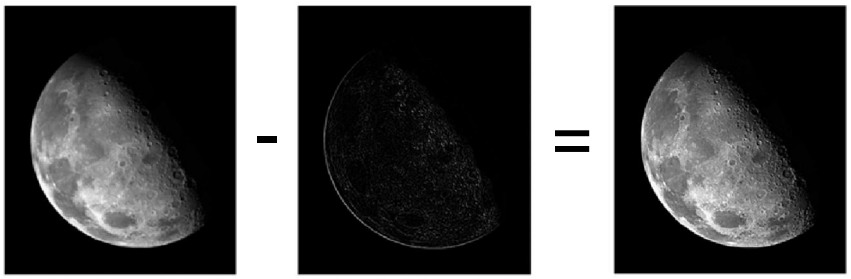
• Subtract the Laplacian result from the original image to generate our final sharpened enhanced image.



Laplacian Filtered Image Scaled for Display

$$g(x,y) = f(x,y) - \nabla^2 f$$

Laplacian image enhancement



Original Image | Laplacian Filtered Image | Sharpened Image

In the final sharpened image edges and fine detail are much more obvious.

Laplacian image enhancement

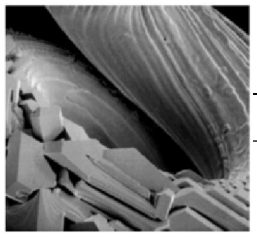
Simplified image enhancement

The entire enhancement can be combined into a single filtering operation:

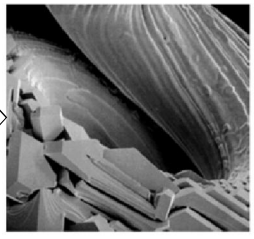
$$
\begin{aligned}
g(x,y) &= f(x,y) - \nabla^2 f \\
&= f(x,y) - [f(x+1,y) + f(x-1,y) \\
&\qquad + f(x,y+1) + f(x,y-1) - 4f(x,y)] \\
&= 5f(x,y) - f(x+1,y) - f(x-1,y) - f(x,y+1) - f(x,y-1)
\end{aligned}
$$

Simplified image enhancement (cont…)

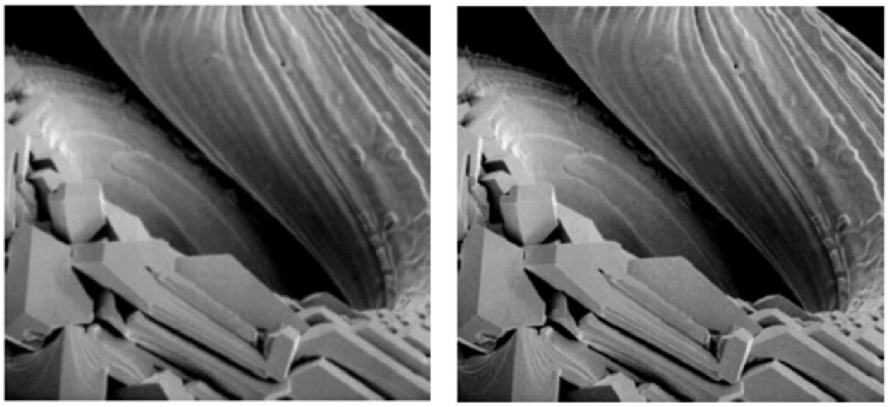
This gives us a new filter which does the whole job for us in one step.



0	-1	0
-1	5	-1
0	-1	0



Simplified image enhancement (cont…)



Variants on the simple Laplacian

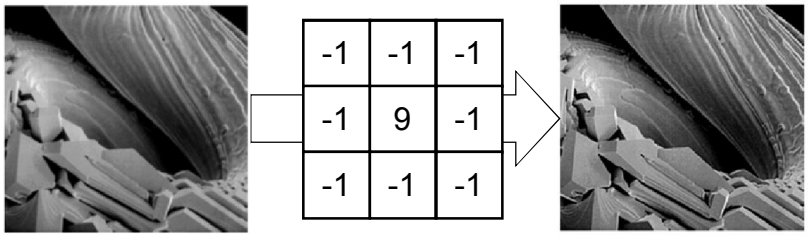
There are lots of slightly different versions of the Laplacian that can be used:

0	1	0
1	-4	1
0	1	0

Simple Laplacian

1	1	1
1	-8	1
1	1	1

Variant of Laplacian



1st derivative filtering

• Implementing 1st derivative filters is difficult in practice.

• For a function *f(x, y)* the gradient of f at coordinates *(x, y)* is given as the column vector:

$$\nabla \mathrm{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

1st derivative filtering (cont…)

The magnitude of this vector is given by:

$$\begin{aligned} \nabla f &= mag\ (\nabla \mathrm{f}) \\ &= \left[G_x^2 + G_y^2\right]^{1/2} \\ &= \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{1/2} \end{aligned}$$

For practical reasons this can be simplified as:

$$\nabla f \approx |G_x| + |G_y|$$

1st derivative filtering (cont…)

There is some debate as to how best to calculate these gradients but we will use:

$$\nabla f \approx \left|(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)\right| + \left|(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)\right|$$

which is based on these coordinates:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sobel operators

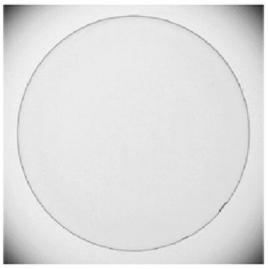
Based on the previous equations we can derive the *Sobel Operators.*

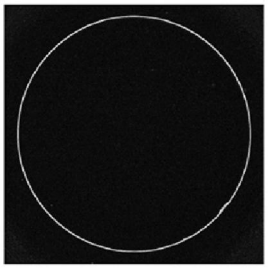
-1	-2	-1
0	0	0
1	2	1

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

To filter an image it is filtered using both operators the results of which are added together.

Sobel example





An image of a contact lens which is enhanced in order to make defects (at four and five o'clock in the image) more obvious

Sobel filters are typically used for edge detection.

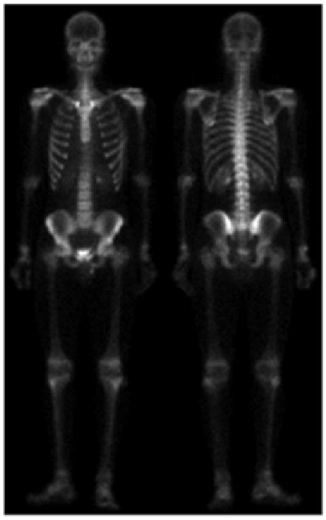
1st & 2nd Derivatives

Comparing the 1st and 2nd derivatives we can conclude the following:

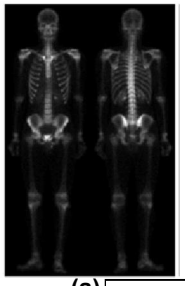
- 1st order derivatives generally produce thicker edges.
- 2nd order derivatives have a stronger response to fine detail e.g. thin lines.
- 1st order derivatives have stronger response to grey level step.
- 2nd order derivatives produce a double response at step changes in grey level.

Combining spatial enhancement methods

- Successful image enhancement is typically not achieved using a single operation.
- Rather we combine a range of techniques in order to achieve a final result.
- This example will focus on enhancing the bone scan to the right.

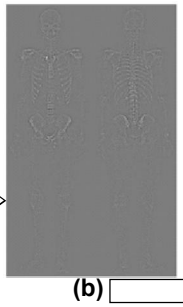


Combining spatial enhancement methods (cont…)



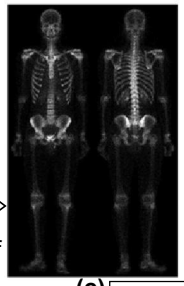
(a)

Laplacian filter of bone scan (a)



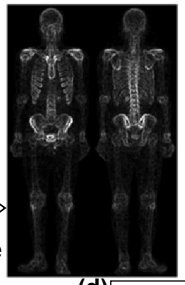
(b)

Sharpened version of bone scan achieved by subtracting (a) and (b)



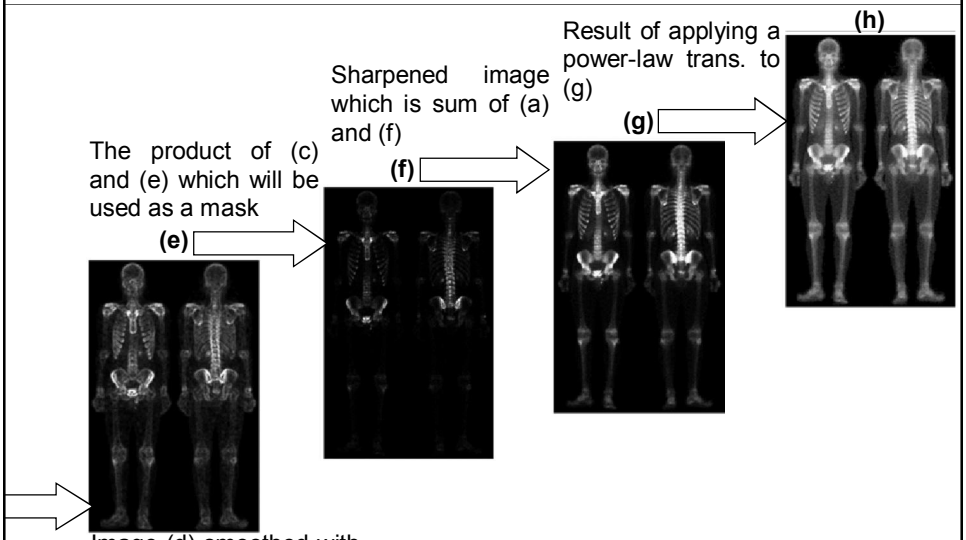
(c)

Sobel filter of bone scan (a)



(d)

Combining spatial enhancement methods (cont…)



Combining spatial enhancement methods (cont…)

Compare the original and final images ….

Summary

- We have looked at:
  - Different kinds of image enhancement
  - Histograms
  - Histogram equalisation
  - Point processing operations
  - Spatial filtering (Neighbourhood operations, Smoothing filters, Correlation and convolution, Sharpening filters, 1st derivative filters, 2nd derivative filters, …).