Digital Image Processing

Lecture Three
Image Enhancement

Sumaiya Fazal Dad

2018

IQRA National University
Peshawar Campus

This Lecture Includes:

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 technologoes

For many of the image processing operations in this lecture grey levels are assumed to be given in the range [0.0, 1.0]

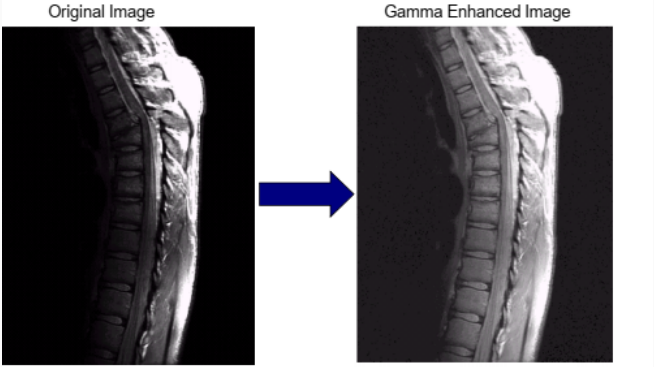
What is Image Enhancement

Image enhancement is the process of making images more useful and better.

The reasons for doing this include:

1. **Highlighting interesting detail in images**
2. **Removing noise from images**
3. **Making images more visually appealing**

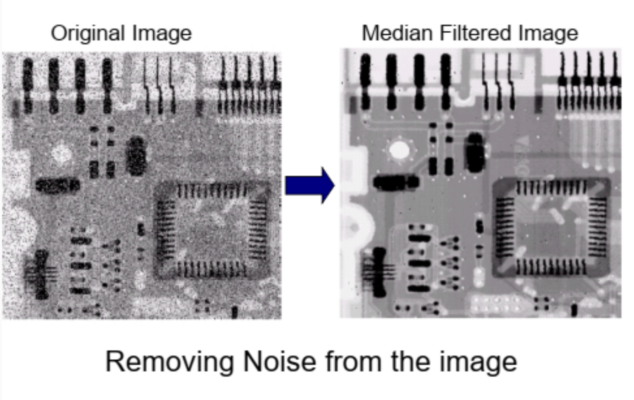
Image Enhancement Example



MRI of a fractured spine.

Image Enhancement Example

Image Enhancement Example



Spatial and Frequency Domains

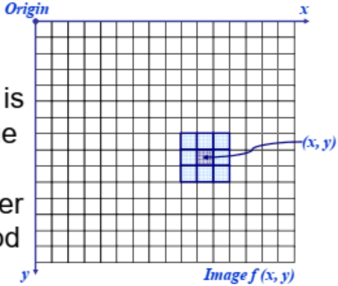
There are **two broad categories** of image enhancement techniques:

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

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 neighbourhood of (x, y)



Point Processing

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

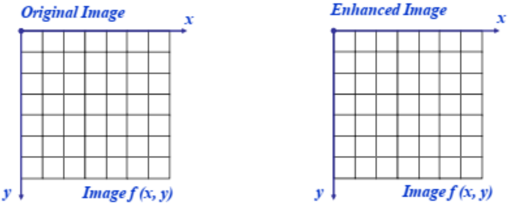
In this case T is referred to as a ***grey level transformation function*** or a ***point processing*** *operation.*

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

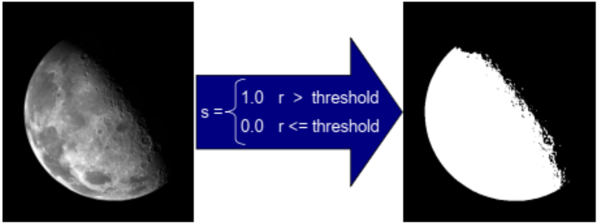
Point Processing Negative Image



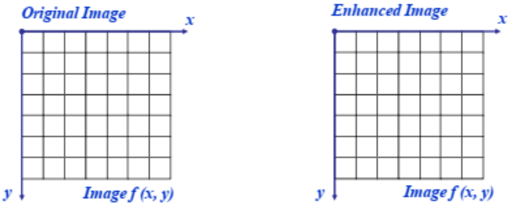
$$s = intensity_{max} - r$$

Point Processing: Thresholding

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



Point Processing: Thresholding



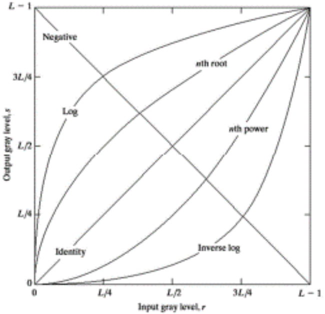
$$s = \begin{cases} 1.0 & r > threshold \\ 0.0 & r <= threshold \end{cases}$$

Basic Grey Level Transformations

There are many different kinds of grey level transformations

Three of the most common are shown here

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



Logarithmic Transformation

The general form of the log transformation is

$s = c * \log_e(1 + r)$

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

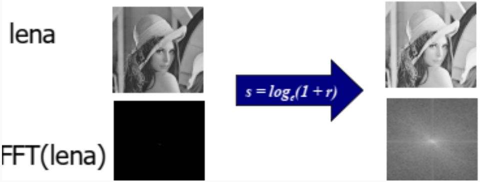
The inverse log transformation performs the opposite transformation

Log transformation increases brightness and contrast of dark regions while reduces contrast of light regions.

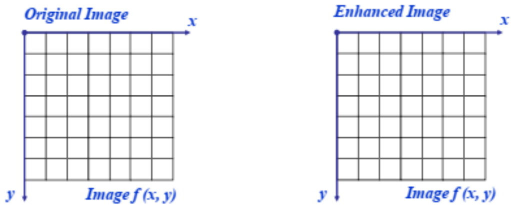
Logarithmic Transformation

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 Transformation



$$s = \log_e(1 + r)$$

We usually set c to 1

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

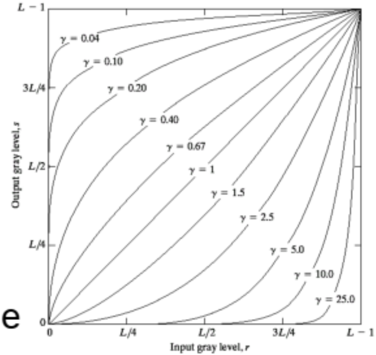
Power Law Transformation

Power law transformations have the following form

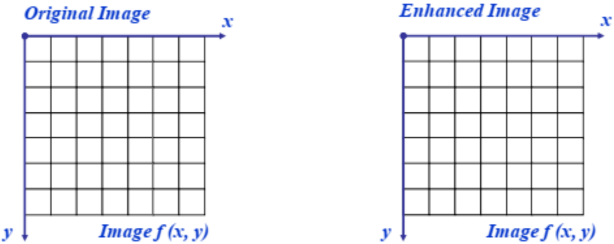
$$s = c * r^{\gamma}$$

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 Transformation

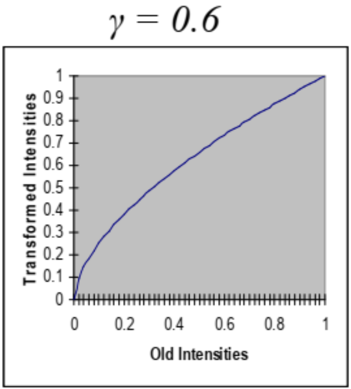


$$s = r^{\gamma}$$

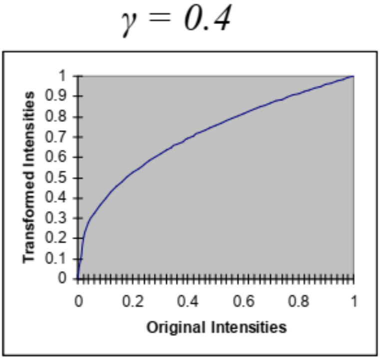
We usually set c to 1

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

Power Law Transformation

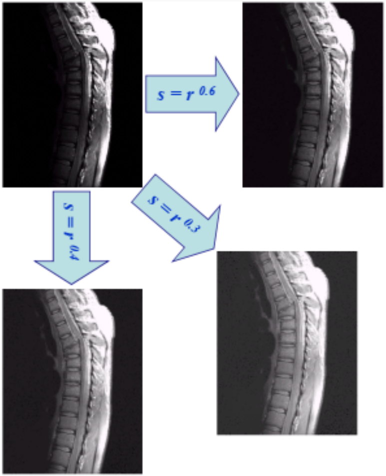


Power Law Transformation



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

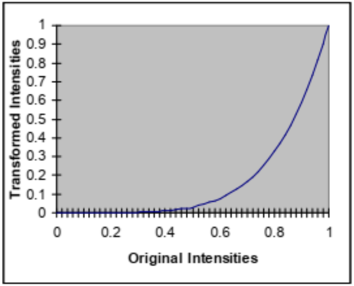
Different curves highlight different detail



Power Law Transformation

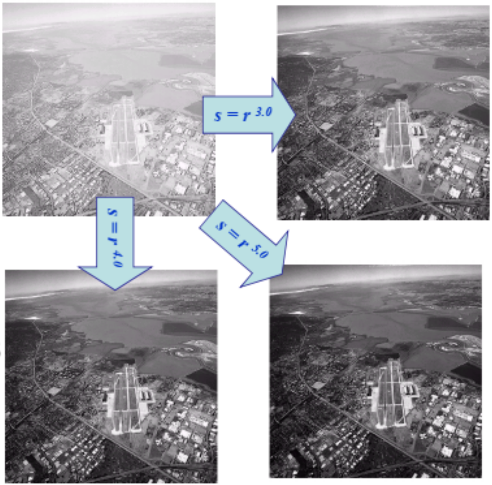
Power Law Transformation

$\gamma = 5.0$



Power Law Transformation

- An aerial photo of a runway is shown
- This time power law transforms are used to darken the image
- Different curves highlight different detail

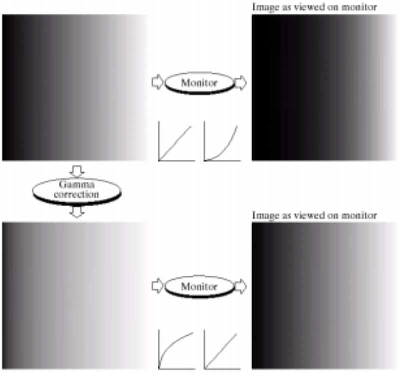


Gamma Correction

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

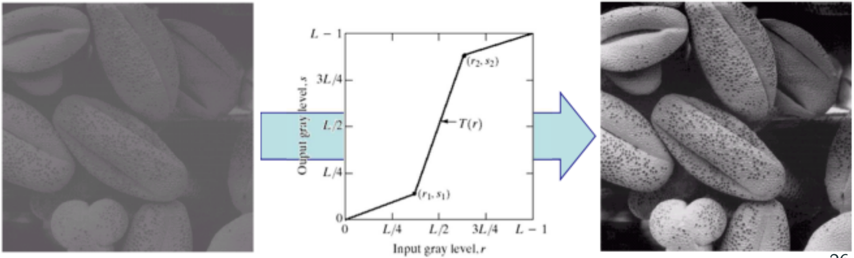
Problem is that display devices do not respond linearly to different intensities

Can be corrected using a log transform



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



Contrast Stretching

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

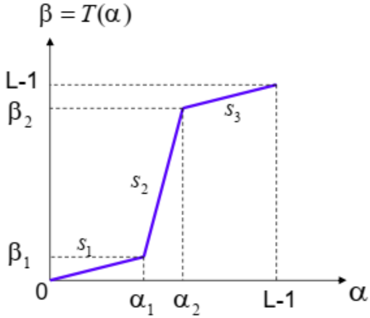
The interesting part is how do we decide on this transformation function?

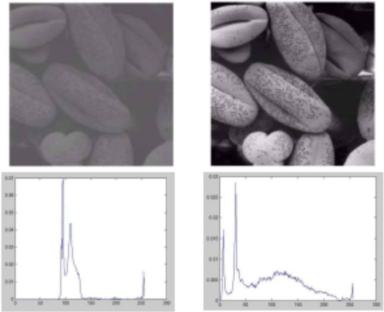
$$s = (r - c)\left(\frac{b - a}{d - c}\right) + a$$

Where s is the output intensity, **r** is the input intensity, **a** and **b** are the lower and upper limit of intensities in gray level image (usually 0 and 255 for 8 bit gray images), **c** and **d** are the stretching points.

Stretch the over-concentrated gray-levels

Piece-wise linear function, where the slope in the stretching region is greater than 1.

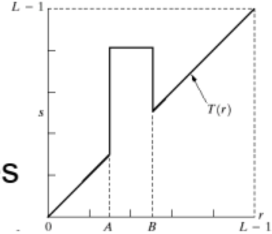


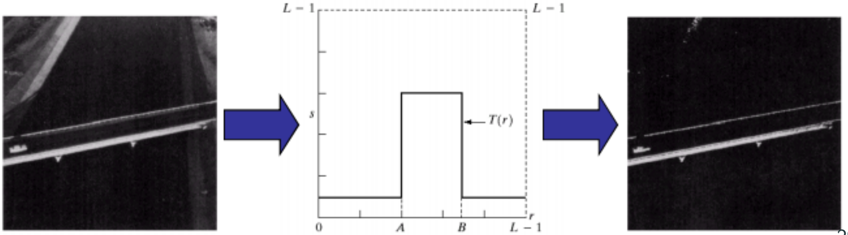


Gray Level Slicing

Highlights a specific range of grey levels

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

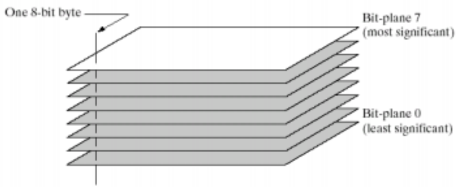


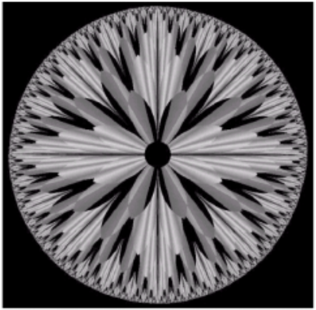


Bit Plane Slicing

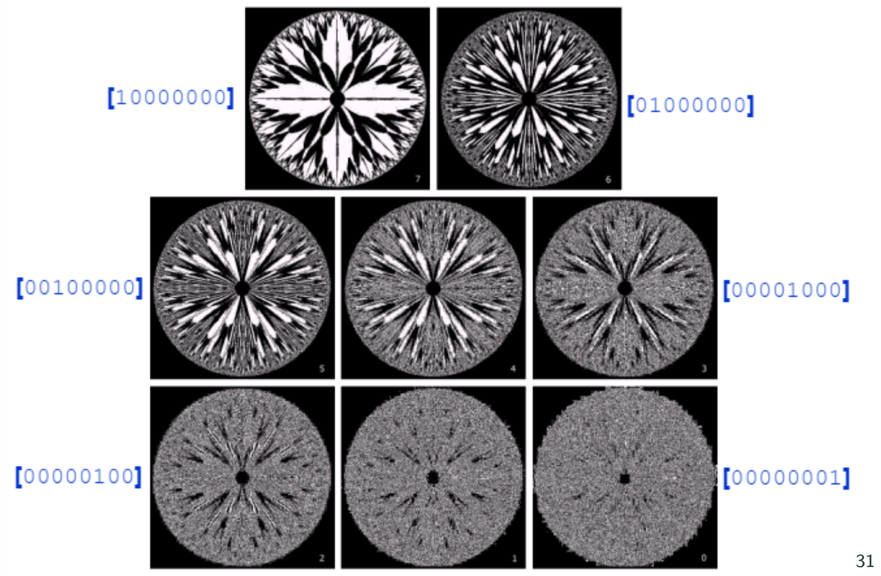
Often by isolating particular bits of the pixel values in an image we can highlight interesting aspects of that image

- Higher-order bits usually contain most of the significant visual information
- Lower-order bits contain subtle details





Bit Plane Slicing



Histogram Equalization

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

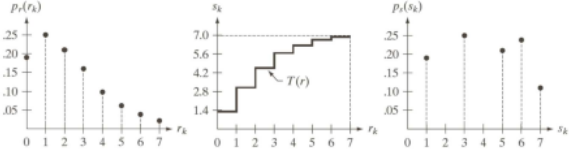
The formula for histogram equalisation is given where

- r_k: input intensity
- s_k: processed intensity
- k: the intensity range (e.g 0.0 – 1.0)
- n_j: the frequency of intensity j
- n: the sum of all frequencies

$$
\begin{aligned}
s_k &= T(r_k) \\
&= \sum_{j=1}^{k} p_r(r_j) \\
&= \sum_{j=1}^{k} \frac{n_j}{n}
\end{aligned}
$$

Histogram Equalization

r_k	n_k	$p_r(r_k) = n_k/MN$	$7 * \sum_u p(u)$	v
$r_0 = 0$	790	0.19	1.33	1
$r_1 = 1$	1023	0.25	3.08	3
$r_2 = 2$	850	0.21	4.55	5
$r_3 = 3$	656	0.16	5.67	6
$r_4 = 4$	329	0.08	6.23	6
$r_5 = 5$	245	0.06	6.65	7
$r_6 = 6$	122	0.03	6.86	7
$r_7 = 7$	81	0.02	7.00	7



a b c

FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

Difference between CS and HE

- **Contrast stretching** is all about increasing the contrast i.e., increasing the difference between the maximum intensity value in an image and the minimum one. All the rest of the intensity values are spread out between this range.
- **Histogram equalization** is about modifying the intensity values of all the pixels in the image such that the histogram is "**flattened**" (in reality, the histogram can't be exactly flattened, there would be some peaks and some valleys, but that's a practical problem).

Difference between CS and HE

- In **contrast stretching** there exists a one-to-one relationship of the intensity values between the source image and the target image i.e., the original image can be restored from the contrast-stretched image. Hence, the process is reversible.
- Once **histogram equalization** is performed, there is no way of getting back the original image i.e., histogram equalization is an **irreversible** operation.