Image processing :

The range of light intensity levels $\rightarrow 10^{10}$

subjective brightness : intensity as percieved by human visual system is a logarithmic function of light intensity incident on the eye.

brightness adaptation : the visual system is not able to operate over a huge range simultanously, instead it changes its overall sensitivity.

brightness discrimination : we have uniformly illuminated large area with intensity I and we add an increment of illumination ΔI as a circle in the middle and we ~~change~~ vary ΔI and observe the result. The result moves from no perceivable change to perceivable change.

→ another aspect of our visual system is that our visual perception depends on what is surrounding us.

generating Images : Images are generated by the combination of illumination source and the reflection or absorption of energy from the scene being Imaged.

$f(x,y)$ can be charactrized by 2 componants: 1) the amount of source illumination incident on the scene being viewed 2) the amount of illumination reflected by the object in the scene.

$$F(x,y) = r(x,y) \cdot i(x,y)$$

Sampling and quantization:

To create a digital image, we need to convert a continous sensed data to digital form. Images may be continuous with respect to the x- and y- coordinates and also in amplitude. we have to sample the function in both coordinates and ~~amplitude~~ in amplitude. Digitizing the coordinates value is called sampling and ~~for the~~ Digitizing the amplitude value is called quantization.

~~Coast~~ Contouring: If the number of quantization levels is not sufficeint, Contouring can be seen in the image contouring starts to become visible at 6 bits/pixel.

~~quantization~~

False contouring: Due to insufficient ~~n~~number of gray levels, this artifact is more visible and It is called false contouring.

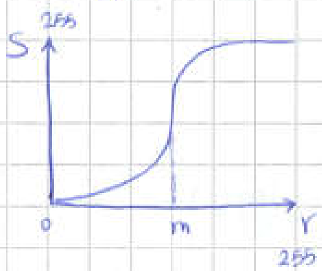
$\Delta I = \frac{256}{2^m}$ $\quad 2^m = \frac{256}{\Delta I}$ $\quad$ ~~then~~ when $m = 2$

$2^m = 2^2 = 4$ we have 4 gray levels. 0 - 63
64 - 127
128 - 191
192 - 256

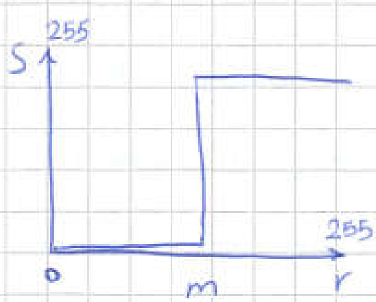
chapter 3:

Image enhancement: It seeks to improve the visual appearance of an image or convert it to a form suited for analysis by a human or machine.

Gray level transformation:



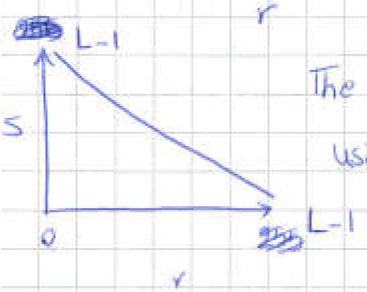
The effect of this transformation would be to produce ~~the~~ an image of higher contrast than the original by darkening The levels below m and brightening The level above m, in the original image. In this transformation, the values of r below m are compressed into a narrow range of s, toward black. the opposite effect takes place for values of r above m.



This transformation produces a two-level (binary) image. A mapping of this form is called a thresholding function. all values of r below m become black and all values of r above m become white.

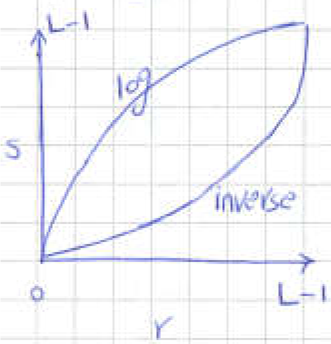
identity transformation:

The output intensities are identical to input intensities.



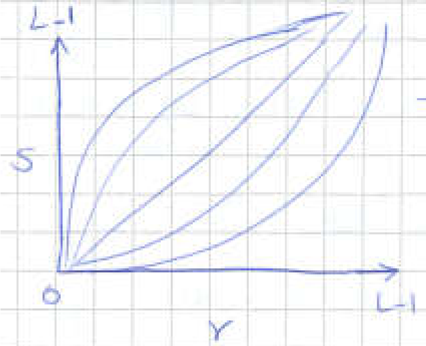
The negative of an image is obtained by using negative transformation which is given by this expression: $s = L - 1 - r$

This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size.



log transformation:

The general form is $s = c\log(1+r)$ where c is constant and $r \geq 0$. This transformation maps a narrow range of low gray-level values in the input image into a wider range of output levels. also the opposite is true of higher values of input levels. we use this transformation to expand the values of dark pixels in an image while compressing the higher-level values. The opposite is true of the inverse log transformation.

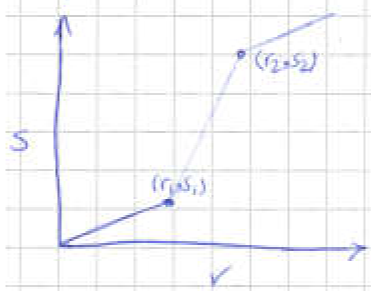


power-law Transformation:

~~The form is~~ power-law transformation have the form $S = Cr^{\gamma}$ where C and γ are positive constant.

power-law curves with fractional values of γ maps a narrow Range of dark input values into a wider Range of output values. The opposite being true for higher values of input levels. by varying γ different transformation curves will be obtained.

piecewise-linear Transformation function:



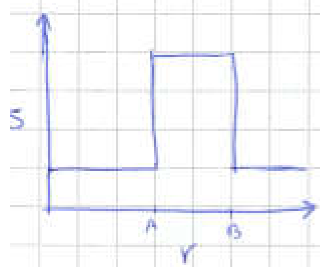
The locations of points (r_1,s_1) and (r_2,s_2) Control the shape of the Transformation function. if $r_1 = s_1$ and $r_2 = s_2$, the Transformation is a linear function that produce no changes in gray levels. if $r_1 = r_2$ and $s_1 = 0$ and $s_2 = L-1$ the transformation becomes a thresholding function that creates a binary image.

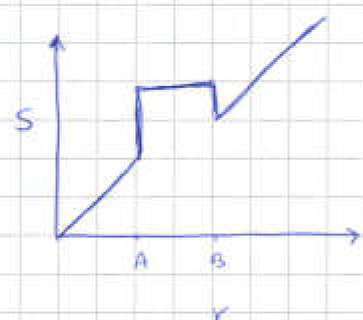
contrast stretching: This transformation has been divided into three transformations. ~~For the first one,~~

The first transformation maps a wider range of gray-level values in the input ~~input~~ into a narrow range of low values in the output.

. the second transformation maps a ~~wider~~ narrow range of middle values in the input into a wider range of values in the output ~~of output values~~. and the third one maps a wider range of higher values in the input into a narrow range of higher values in the output.

The idea behind contrast stretching is to increase the dynamic range of the gray levels in the image being processed.





gray-level slicing : ~~we can use these two~~

highlighting a specific range of gray levels in an image often is desired. In the left, transformation assigns a high value for all gray level values in the range of interest and a low value for all othe gray levels. the second transformation brightens the desired range of gray levels but preserves the background and gray-level tonalities in the image.

Bit-plane slicing: highlighting the contribution made to total image by specific bits might be desired.
if each pixel in the image represented by 8 bits.
The image is composed of eight 1-bit planes, Ranging from bit plane 0 to bit plane 7. in terms of 8-bit bytes, plane 0 contains all the lowest order bits and plane 7 contains all the High-order bits. the higher-order bits (especially the top four) contain the majority of the visually significant data. the other bit planes contribute to more subtle details in the image. separating a digital image into its bit plane is useful for analysing the relative importance played by each bit of the image. in terms of bit-Plane extraction for 8-bit image, ~~for showing a binary image for~~ bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation function that maps all levels in the image between 0 and 127 to 0 and maps all levels between 128 and 255 to 255.

Histogram:

Histogram processing re-scales an image so that the enhanced image histogram follow some desired form. also, It is the distribution of pixel values.

histogram equalization: It helps us to see details that we could not see before. when we apply histogram equalization to a dark image the intensity values in the output will be spread to the entire range of intensity values. the contrast of the image will be increased.

Histogram matching:

histogram matching is the process of transforming the histogram of a reference image to ~~target image~~ be as similar as possible to a given target image histogram. histogram matching can be accomplished by applying histogram equalization twice, one to the reference image and one to the target image. Then, use the inverse transform of the reference image histogram equalization to map points from the equalized reference image to the original target image histogram.

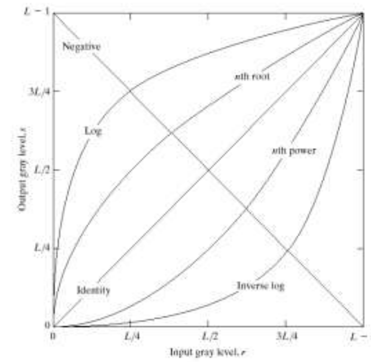


FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.

Answer: Identity: does nothing

Negative: produces the negative of an image. It can be useful, e.g. in viewing mammograms
Log: is a logarithm transformation. It maps a narrow range of low gray-level values in the input image into a wider range of output levels and vice versa for the higher values of the input levels. It expands values of dark pixels in an image while compressing the higher-level values.
Inverse log: is the inverse logarithm transformation. It does the opposite of what the log transformation does.
Nth root: is power-law transformation (with power smaller than 1). It maps a narrow range of dark input values into a wider range of output values.
Nth power: is a power-law transformation (with power greater than 1). It has the opposite effect of the nth root.

Smoothing Spatial filtering:

1-Smoothing linear filtering:

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask. These filters sometimes are called *averaging filters*. For reasons explained in Chapter 4, they also are referred to a *lowpass filters*.

The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced "sharp" transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect that they blur edges. Another application of this type of process includes the smoothing of false contours that result from using an insufficient number of gray levels

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

A spatial averaging filter in which all coefficients are equal is sometimes called a *box filter*.

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

The second mask shown in Fig. 3.34 is a little more interesting. This mask yields a so-called *weighted average*, terminology used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others. In the mask shown in Fig. 3.34(b) the pixel at the center of the mask is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average. The other pixels are inversely weighted as a function of their distance from the center of the mask. The diagonal terms are further away from the center than the orthogonal neighbors (by a factor of $\sqrt{2}$) and, thus, are weighed less than these immediate neighbors of the center pixel. The basic strategy behind weighing the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an attempt to reduce blurring in the smoothing process. We could have picked other weights to accomplish the

As mentioned earlier, an important application of spatial averaging is to blur an image for the purpose getting a gross representation of objects of interest, such that the intensity of smaller objects blends with the background and larger objects become "bloblike" and easy to detect. The size of the mask establishes the relative size of the objects that will be blended with the background.

Problem 2 (15 points):

What is the type of the following filters (lowpass/highpass/bandpass) and why?

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1\\ \hline 1 & 1 & 1\\ \hline 1 & 1 & 1\\ \hline\end{array}\qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1\\ \hline 2 & 4 & 2\\ \hline 1 & 2 & 1\\ \hline\end{array}\qquad \boxed{\begin{array}{ccc} -1 & -1 & -1\\ -1 & 8 & -1\\ -1 & -1 & -1\end{array}}$$

(a) (b) (c)

Ans:

The first is a low-pass filter because it smoothens the output intensity value by using the values of the pixel's neighbors. (5 points)

The second is a low-pass filter because it smoothens the output intensity value by using the values of the pixel's neighbors. (5 points)

The third one is a high-pass filter because it calculates the difference of the intensities in the neighborhood of a pixel (edges). (5 points)

2- order-statistic filters(non-linear):

Order-statistics filters are nonlinear spatial filters whose response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result. The best-known example in this category is the *median filter*, which, as its name implies, replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median). Median filters are quite popular because, for certain types of random noise, they provide excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size. Median filters are particularly effective in the presence of *impulse noise*, also called *salt-and-pepper noise* because of its median of 20. Thus, the principal function of median filters is to force points with distinct gray levels to be more like their neighbors. In fact, isolated clusters

Sharpening spatial filter:

The principal objective of sharpening is to highlight fine detail in an image or to enhance detail that has been blurred, either in error or as a natural effect of

In the last section, we saw that image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood. Since averaging is analogous to integration, it is logical to conclude that sharpening could be accomplished by spatial differentiation. This, in fact, is the case, and the discussion in

1-foundation:

first-order derivative $\frac{\partial f}{\partial x} = f(x + 1) - f(x).$

we define a second-order derivative as the difference

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

In summary, comparing the response between first- and second-order derivatives, we arrive at the following conclusions. (1) First-order derivatives generally produce thicker edges in an image. (2) Second-order derivatives have a stronger response to fine detail, such as thin lines and isolated points. (3) First-order derivatives generally have a stronger response to a gray-level step. (4) Second-order derivatives produce a double response at step changes in gray level.

Answer:

- **First order derivative**
  - produces thicker edges (2 points)
  - has stronger response to grey-level steps (2 points)
- **Second order derivative**
  - has a much stronger response to details (1 point)
  - produces a double response at step changes in grey level (1 point)
  - has a stronger response to a line than to a step and to a point than to a line. (1 point)
- **Conclusion**
  - Second derivative is more useful to enhance image details (3 points)

Second derivatives (Laplacian): it is linear

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

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

$$\nabla^2 f = \left[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)\right] - 4f(x, y). \tag{3.7-4}$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

Because the Laplacian is a derivative operator, its use highlights gray-level discontinuities in an image and deemphasizes regions with slowly varying gray levels. This will tend to produce images that have grayish edge lines and other discontinuities, all superimposed on a dark, featureless background. Background

Unsharp masking and high-boost filtering:

A process used for many years in the publishing industry to sharpen images consists of subtracting a blurred version of an image from the image itself. This process, called *unsharp masking*, is expressed as

$$f_s(x, y) = f(x, y) - \bar{f}(x, y) \tag{3.7-7}$$

where $f_s(x, y)$ denotes the sharpened image obtained by unsharp masking, and $\bar{f}(x, y)$ is a blurred version of $f(x, y)$. The origin of unsharp masking is in dark-

A slight further generalization of unsharp masking is called *high-boost filtering*. A high-boost filtered image, f_{hb}, is defined at any point (x, y) as

$$f_{\text{hb}}(x, y) = Af(x, y) - \bar{f}(x, y) \tag{3.7-8}$$

First derivatives(gradient): non linear

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

First derivatives in image processing are implemented using the magnitude of the gradient. For a function $f(x, y)$, the gradient of f at coordinates (x, y) is de-

mated inspection. Note also that the gradient process highlighted small specs that are not readily visible in the gray-scale image (specs like these can be for-

the lens). The ability to enhance small discontinuities in an otherwise flat gray field is another important feature of the gradient. ■

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

called the *Sobel operators*,

plement Eq. (3.7-18) via the mechanics given in Eq. (3.5-1). The idea behind using a weight value of 2 is to achieve some smoothing by giving more importance to the center point (we discuss this in more detail in Chapter 10). Note that

Combining spatial enhancement methods:

detect diseases such as bone infection and tumors. Our objective is to enhance this image by sharpening it and by bringing out more of the skeletal detail. The narrow dynamic range of the gray levels and high noise content make this image difficult to enhance. The strategy we will follow is to utilize the Laplacian to highlight fine detail, and the gradient to enhance prominent edges. For reasons that will be explained shortly, a smoothed version of the gradient image will be used to mask the Laplacian image (see Section 3.4 regarding masking). Finally, we will attempt to increase the dynamic range of the gray levels by using a gray-level transformation.