2 Intensity transformations and spatial filtering

2.1 Preliminaries

In this chapter, we will discuss about some basic image processing techniques that include the intensity transforms and spatial filtering. All the techniques introduced here will be performed in the spatial domain.

## 2.2 Basic Intensity Transformation Functions

Photographic Negative:

This is perhaps the simplest intensity transform. Supposing that we have a grey level image in the range [0,1], it is expected to transform the black points (0s) into the white ones (1s), and the white pixels (1s) into the black ones (0s). This simple transform can be denoted by (assume that $f(x, y)$ is normalized into the range [0,1])

$$\tilde{f}(x, y) = 1 - f(x, y) \tag{2.2.1}$$

For a 256 gray level image, the transform can be accomplished by

$$\tilde{f}(x, y) = 1 - f(x, y) / 256 \tag{2.2.2}$$

An example of the Photographic Negative transform is shown in **Figure 14**.

(a)   (b)
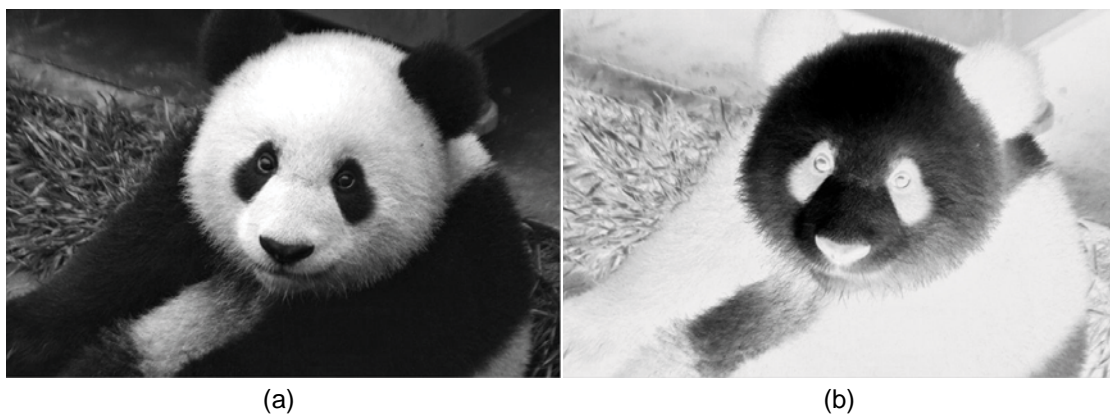
Figure 14 A panda Image (a), and its photographic negative transformed image (b).

Gamma Transform

Gamma transform is also called power-law transform. It is mathematically denoted as follows:

$$\tilde{f}(x,y) = c * f(x,y)^{\gamma} \tag{2.2.3}$$

where c and γ are two constants. The gamma transform can make pixels look brighter or darker depending on the value of γ. When $f(x,y)$ is within the range [0,1] and γ is larger than one, it makes the image darker. When γ is smaller than one, it makes the image look brighter. **Figure 15** shows the output of the Gamma transform against different inputs with the parameters set as 0.5, 1 and 2 ($c = 1$). From this plot, we can see that the curve with $\gamma = 2$ is below the curve with $\gamma = 1$. This indicates that the output is smaller than the input, which explains why an image in the Gamma transform, when $\gamma = 2$, will become darker.
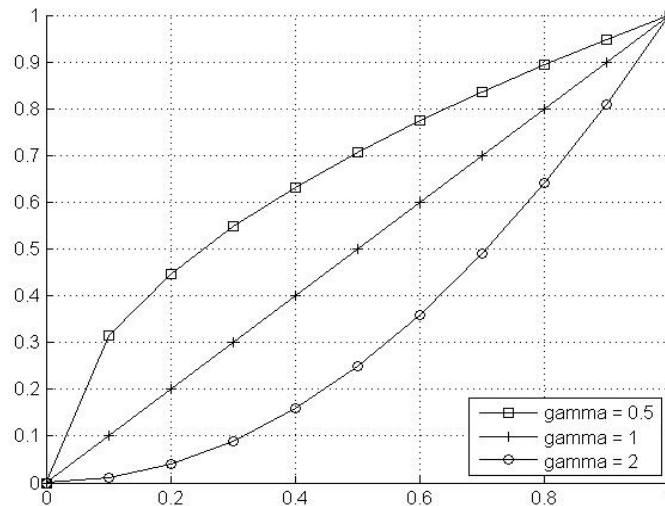
Figure 15 The plot of the Gamma transform against different parameters. Horizontal axis is the input, and vertical is the output.

Figure 16 shows that the Gamma transformed panda image under different parameter values ($c = 1$).

(a)   (b)
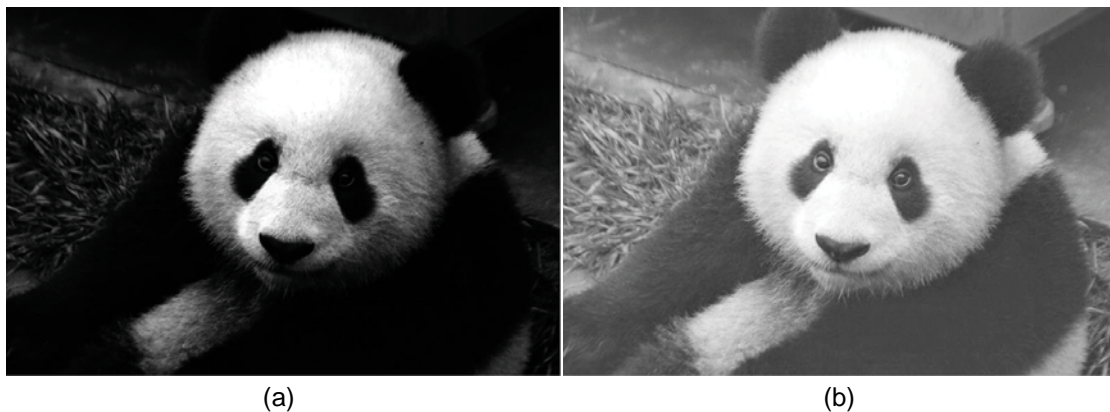
**Figure 16** Gamma transformed images different parameters. (a) Gamma = 2 ; (b) Gamma = 0.5 .

2.3   Histogram Processing

Histogram is one simple but very important statistical feature of an image. It has been commonly used in image processing. Intensity histogram is a distribution of the grey level values of all the pixels within the image. Each bin in the histogram represents the number of pixels whose intensity values fall in that particular bin. A 256 grey level histogram is often used, where each grey level corresponds to one bin. Using bi to represent the ith number of bins, the histogram can be represented as follows:

$$h(i) = \#\{(x, y), f(x, y) \in bi\} \tag{2.3.1}$$

where # represent the Cardinality of the set. Though it is possible to show the histogram of a color image, in this chapter we mainly focus on the gray level histogram. The color histogram will be shown in Chapter 1 (*Digital Image Processing – Part Two*).

Figure 17 shows an image and its grey level histogram of 64 bins.

(a)   (b)
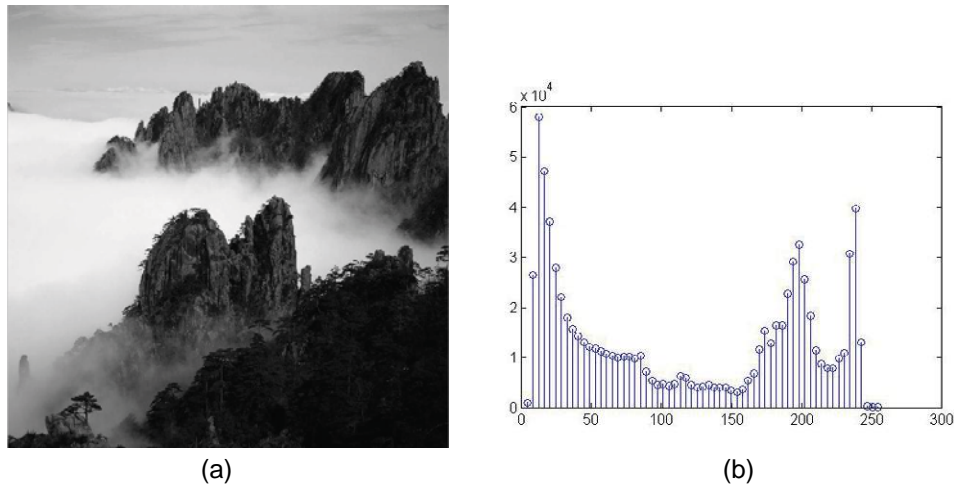
Figure 17 A mountain image (a) and its 64 bins grey level histogram (b).

2.3.1   Histogram Equalization

It is a fact that the histogram of an intensity image lies within a limited data range. Those images usually have black or white foreground and background. **Figure 18** shows an image example whose intensity distribution is either black or white. From **Figure 18** (b), we can see that a very large portion of pixels whose intensity rests within the range [0, 50] or [180, 255]. A very small portion of pixel resides in the range of [50, 180]. This makes some details of the image hardly visible, e.g. the trees on the mountains in the image shown in **Figure 18** (a). This problem can be solved by a histogram stretching technique called *histogram equalization*.

The basic idea of histogram equalization is to find the intensity transform such that the histogram of the transformed image is uniform. Of the existing probabilistic theories, there exists such an intensity transform. Suppose that we have an image $f(x, y)$, and its histogram $h(i)$, we have the accumulative function of $h(i)$ as follows:

$$c(i) = \int_0^i h(t)\, dt \tag{2.3.1.1}$$

It can be proved that such a transform makes the variable $y = c(i)$ follow a uniform distribution. Thus, for a 256 grey level image, the histogram equalization can be performed by the following equation:

$$t = \frac{256}{n} * c(f(x, y)) \tag{2.3.1.2}$$

where is the total number of pixels in the image.

(a)   (b)
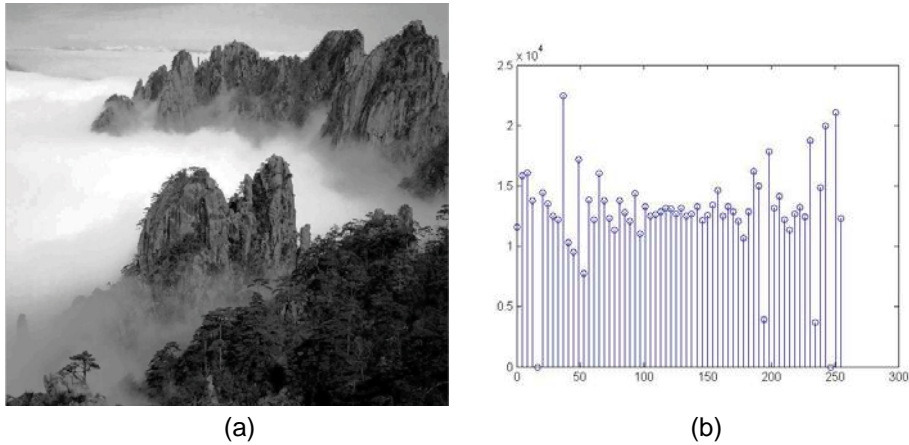
Figure 18 (a) The mountain image after histogram equalization, and (b) histogram.

2.3.2 Histogram Matching

Histogram Equalization is a special case of histogram matching. The general purpose of histogram matching is to find a transform that can adjust an image by converting its histogram to a specified histogram a priori. As we have learnt from the above discussion, the desired histogram in histogram equalization is a uniform distribution. In this case, there exists a transformation as shown in Eq. (2.3.1.2). A straightforward implementation can be achieved. However, in more general case we may want the desired histogram to be a specific shape and to adjust the image based on this histogram. This operation is called histogram matching. It can be achieved by matching the accumulative distributions of the original histogram and desired histogram. Step by step, it can be implemented as follows:

1) Compute the histogram of the original image $f(x, y)$ and its accumulated distribution $c(i)$.
2) Compute the accumulated distribution of the desired histogram $d(i)$.
3) Compute the desired intensity $g(x, y)$ at location (x, y) by matching the $d(i)$ and $c(i)$ using minimum distance criteria (nearest n) as follows:

$$g(x, y) = i^* = \arg\min_i(|d(i) - c(f(x, y))|) \tag{2.3.2.1}$$

Figure 19 shows some internal results of the above steps on the image shown in **Figure 17** (a).

(a)   (b)
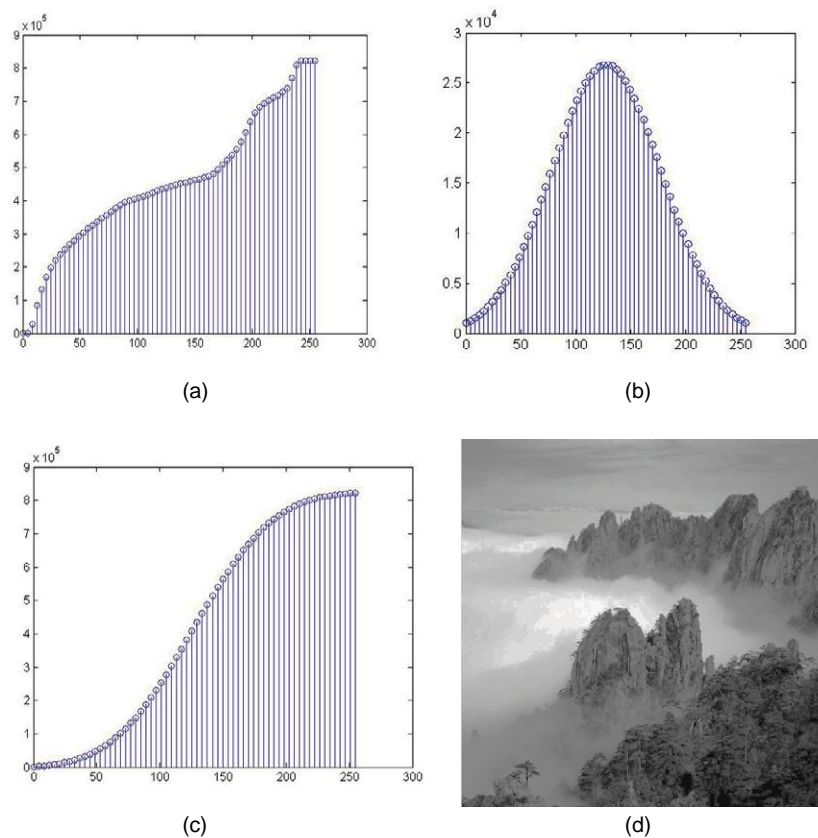

(c)   (d)

Figure 19 (a) The accumulated histogram of the image in Figure 2.3.1 (a); (b) the desired histogram; (c) the desired accumulated histogram; (d) the adjusted image based on image matching according to Eq. (2.3.2.1).

2.4   Fundamentals of Spatial Filtering

Filtering is a fundamental operation in image processing. It can be used for image enhancement, noise reduction, edge detection, and sharpening. The concept of filtering has been applied in the frequency domain, where it rejects some frequency components while accepting others. In the spatial domain, filtering is a pixel neighborhood operation. Commonly used spatial filtering techniques include median filtering, average filtering, Gaussian filtering, etc. The filtering function sometimes is called filter mask, or filter kernel. They can be broadly classified into two different categories: linear filtering and order-statistic filters. In the case of linear filtering, the operation can be accomplished by convolution, i.e., the value of any given pixel in the output image is represented by the weighted sum of the pixel values of its neighborhood (a linear combination) in the input image. For order-statistic filter, the value of a given pixel in the output image is represented by some statistics within its support neighborhood in the original image, such as the median filter. Those filters are normally non-linear and cannot be easily implemented in the frequency domain. However, the common elements of a filter are (1) A neighbourhood (2) an operation on the neighbourhood including the pixel itself. Typically the neighbourhood is a rectangular of different size, for example 3×3, 5×5 … and smaller than the image itself.

2.5   Smoothing Spatial Filters

2.5.1   Smoothing Linear Filters

As stated above, linear filters have observed their output values as the linear combination of the inputs. The commonly used smoothing filters are averaging and Gaussian filters. The smoothing filters usually have the effect of noise reduction (also called low pass filtering, which will be discussed in the next chapter). It can be performed using the convolution operation.

$$s(x,y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} h(m,n)f(x-m,y-n) \qquad (2.5.1.1)$$

$h(m,n)$ is a filtering mask of size MxN. Each element in this filter mask usually represents the weights used in the linear combination. The types of different linear filters correspond to different filter masks. The sum of all the elements in $h(m,n)$ will affect the overall intensities of the output image. Therefore, it is sensible to normalize $h(m,n)$ such that the sum is equal to one. In the case of negative values in $h(m,n)$, the sum is typically set to zero. For smoothing (low pass filters), the values in the mask must be positive; otherwise, this mask will contain some edges (sharpening filters).

The linear filter can also be performed using correlation as follows:

$$s(x,y) = \sum_{m=-M/2}^{M/2} \sum_{n=-N/2}^{N/2} h(m,n)f(x+m,y+n) \qquad (2.5.1.2)$$

If $h(m,n)$ is symmetric, i.e. $h(m,n) = h(-m,-n)$, the correlation is equivalent to convolution.

Average Filtering

The average filtering is also called mean filtering, where the output pixel value is the mean of its neighborhood. Thus, the filtering mask is as follows (3×3 as an example)

$$h = 1/9 \begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

(2.5.1.2)

In practice, the size of the mask controls the degree of smoothing and loss of the details.

Gaussian Filtering

Gaussian filtering is another important filter. The weights in the filter mask are of a Gaussian function (here we assume it is isotropic):

$$h(m, n) = \frac{1}{Z} \exp(-0.5 * (m^2 + n^2) / \sigma)$$

(2.5.1.3)

where σ is the variance and controls the degree of smoothing. The larger value σ is, the larger degree smoothing can be achieved. For example, the 3×3 Gaussian filter with σ set as 1 is represented by the following equation.

$$h(m, n) = \begin{vmatrix} 0.60 & 0.10 & 0.60 \\ 0.10 & 0.16 & 0.10 \\ 0.60 & 0.10 & 0.60 \end{vmatrix}$$

(2.5.1.4)

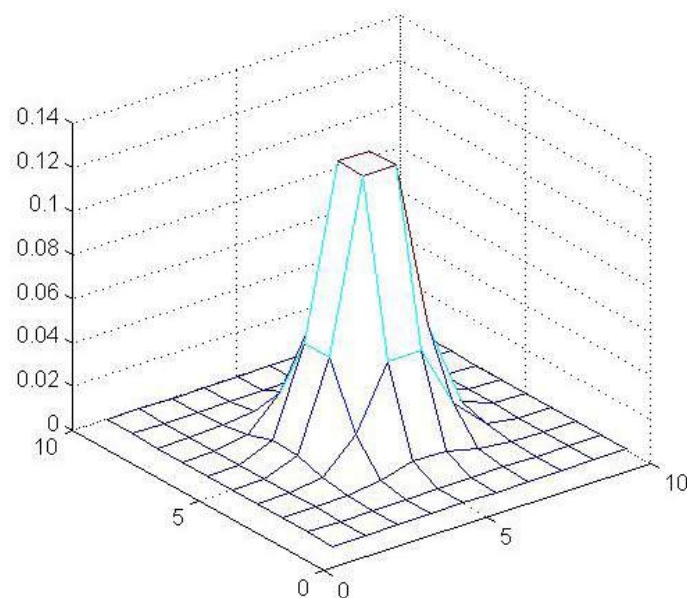**Figure 20** shows the Gaussian mask of size 5×5,

Figure 20 The 3D plot of the Gaussian filter with the variance set as 1.

Figure 21 shows the Lenna image as well as the filtered images using mean filtering with different mask sizes and Gaussian filtering with different. We can see that with the increment of the mask size of the mean filter, the level of smoothing becomes stronger. This also holds for the Gaussian filtering with different.
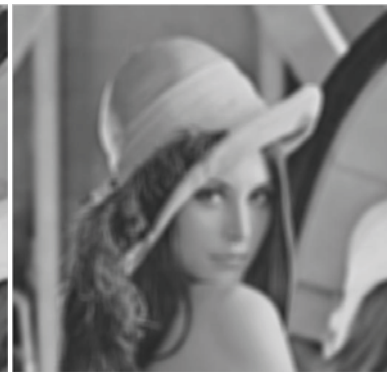

(a)

(b)   (c)   (d)
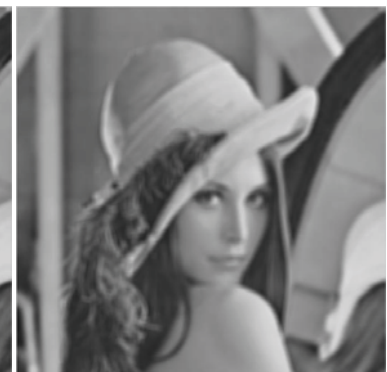

(e)   (f)   (g)

Figure 21 (a) The Lenna image; (b) (c) (d) filtered images using mean filtering with mask size 3, 7, 11; (e) (f) (g) filtered images using Gaussian filtering with different variances at 1, 5, 9.

2.5.2   Order-Statistic Filters

The order-statistical filters are usually non linear filters, which are hardly represented by convolution. Commonly used filters include median filter. There are some other filters as well such as max/min filter. The median filter simply replaces the value of the pixel with the median value within its neighborhood. Similarly, the max/min filter replaces the value of the pixel with the maximum/minimum value within its neighborhood.

Figure 22 shows the filtered image using a median filter of different mask sizes.

(a)   (b)   (c)

Figure 22 Filtered Lenna image using median filtering of different masks 3, 7, and 11.

Download free eBooks at bookboon.com

2.6 Sharpening filters

The objective of sharpening is to draw the attention to the fine details of an image. This is also related to the situation where an image that has been blurred and now needs to be de-blurred. In contrast to the process of image smoothing that normally uses pixel averaging techniques, sharpening can be conducted using spatial differentiation. The image differentiation actually enhances edges and other discontinuities and depresses the areas of slowly changing gray-level values.

The derivative of a function is determined via differencing. A basic first-order derivative of a one-dimensional function $f(x)$ is

$$\frac{\partial f}{\partial x} = f(x+1) - f(x) \tag{2.6.1}$$

Similarly, a second-order derivative can be defined as follows:

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x) \tag{2.6.2}$$

In this section, we mainly talk about the use of two-dimensional and second-order derivatives for image sharpening. It is inevitable to mention the concept of *isotropic filters*, whose response is independent of the discontinuity direction in the image.

One of the simple isotropic filters is the Laplacian, which is defined as follows with a function $f(x, y)$:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{2.6.3}$$

Where

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \tag{2.6.4}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \tag{2.6.5}$$

Then the two-dimensional Laplacian is obtained summing the two components:

$$\nabla^2 I = [I([+1, \backslash) + I([-1, \backslash) + I([, \backslash +1) + I([, \backslash -1)] - 4I([, \backslash) \tag{2.6.6}$$

To implement Eq. (2.6.6) we illustrate four filter masks used for the Laplacian in **Figure 23**.

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

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

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

Figure 23 Filter masks used in the Laplacian enhancement.

Such a Laplacian enhancement can be described as

$$F(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) \\ f(x, y) + \nabla^2 f(x, y) \end{cases} \tag{2.6.7}$$

To simplify the Laplacian we have

$$) ([,\backslash) = I([,\backslash) - [I([+1,\backslash) + I([-1,\backslash) + I([,\backslash+1) + I([,\backslash-1)] \qquad (2.6.8)$$

One of the examples using the Laplacian is shown in **Figure 24**.
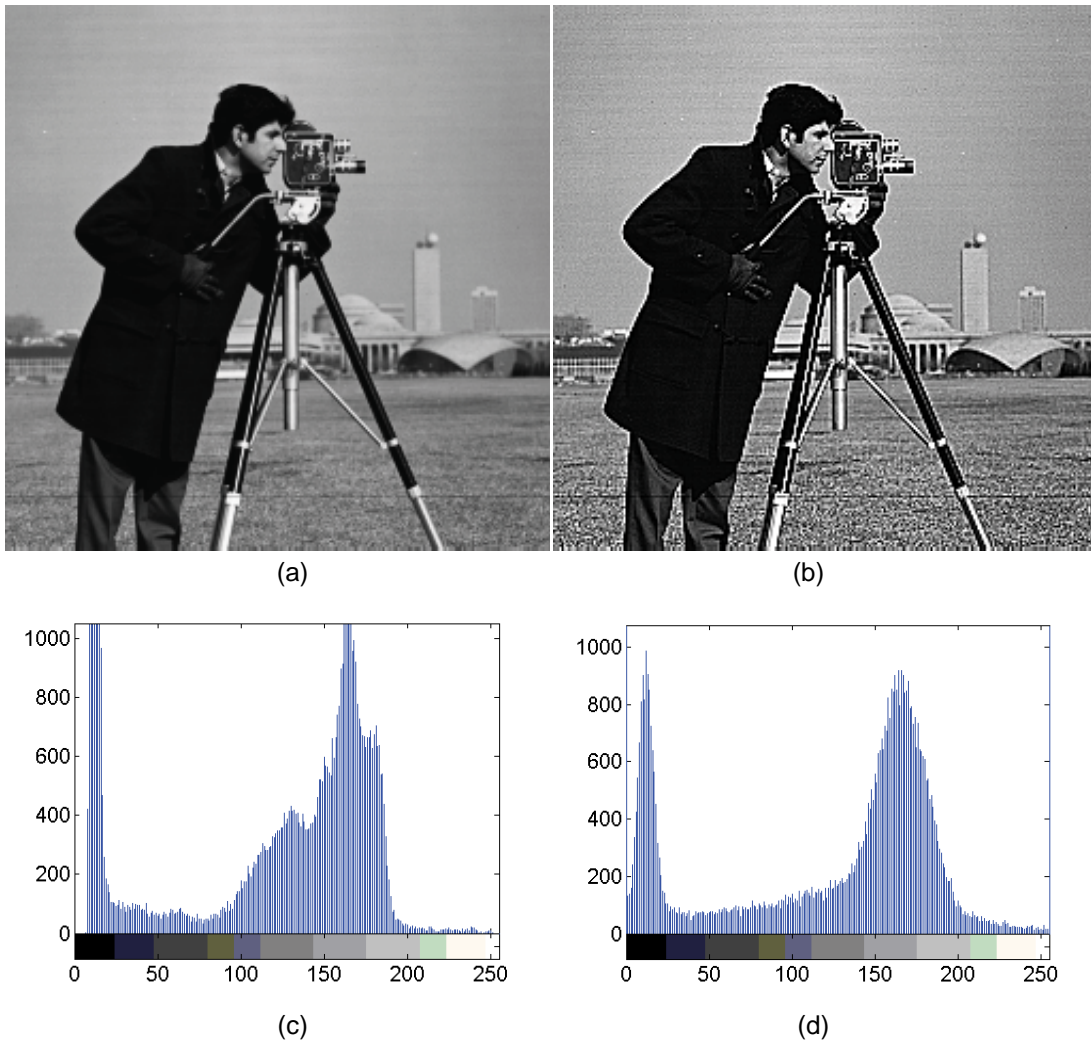
Figure 24 Image sharpening and histograms: (a) original image and its histogram (c), (b) shapened image and its histogram (d).

For a function $f(x, y)$, the gradient of f is defined as

$$\nabla f = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \qquad (2.6.9)$$

In the implementation, this equation can be changed to

$$\nabla f \approx | (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) | + | (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) | \qquad (2.6.10)$$

Where $(z_1, \dots, z_9)$ are the elements of the 3×3 mask. This mask can be one of the following patterns:

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

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

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

Figure 25 Image mask that is used to generate gradients.

2.7   Combining image enhancement methods

In real applications, it is hard to know what kind of noise has been added to an image. Therefore, it is difficult to find a unique filter that can appropriately enhance this noisy image. However, it is possible if several de-blurring methods can be combined in a framework in order to pursue a maximum denoising outcome. In the section, we explain one of the combinatorial techniques using an X-ray example [6].

Figure 26 (a) illustrates a male chest's X-ray image. The purpose of the process is to highlight the middle cross section of the image using the combination of sharpening and edge detection. **Figure 26** (b) shows the result of applying the median filter, (c) is the outcome of using Sobel edge detection, and finally (d) demonstrates the combination of the Laplacian and Sobel process. **Figure 26** (d) shows the edge details in the graph, which highlights the structure of the central cross section of the image.
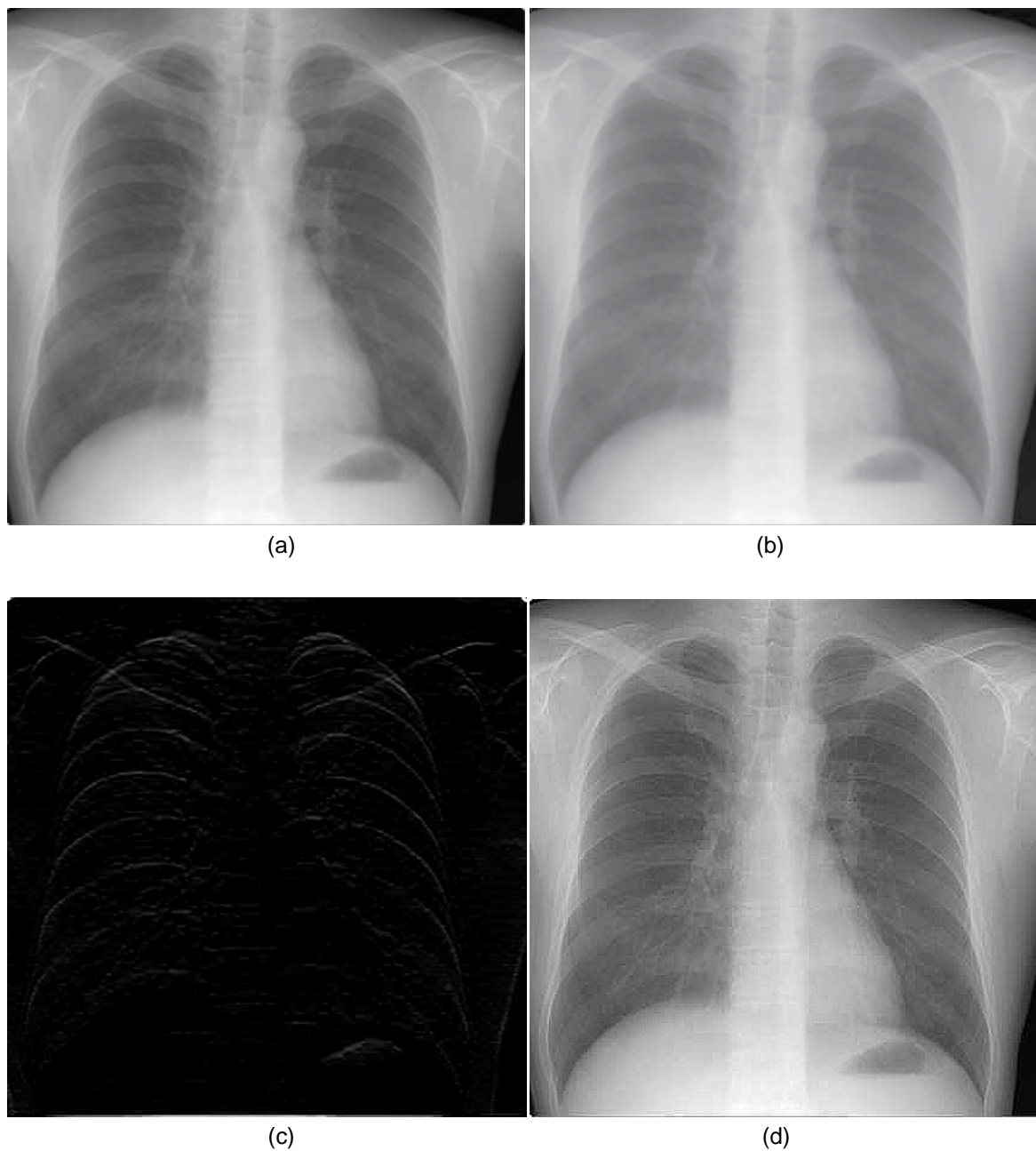
Figure 26 Illustration of using the combination of different segmentation algorithm: (a) original, (b) median filtering, (c) Sobel edge detection and (d) combination of Laplacian and Sobel process.

2.8   Summary

In this chapter, some basic image intensity transformation functions have been provided. This consists of negative and Gamma transforms. Afterwards, histogram based processing techniques were introduced. Especially, histogram equalisation and matching techniques were summarised due to their importance in image processing applications.

To further enhance the image quality we then introduced a number of spatial filters. Particularly, we discussed about averaging filtering, Gaussian filtering, and statistic filters. Due to the needs of enhancing image contrasts, we then focused on the technical details of sharpening filters, which have been commonly used in the image processing applications. As an extension, we discussed about the applications of the combination of image enhancement methods. One of the examples is to integrate a Laplacian filter with the Sobel edge detection for better image details.

2.9   References

[6]   http://www.radiologyinfo.org/en/photocat/gallery3.cfm?pid=1&Image=chest-xray.jpg&pg=chestrad, accessed on 15 October, 2009.

2.10 Problems

(8)   Please perform the image negative and Gamma transform for the cameraman image as follows:

(9)   Please use the Matlab tools to generate codes to add Gaussian noise (mean = 0, variance = 2) to the above cameraman image, followed by averaging and Gaussian filtering respectively.

(10)   Can you find a proper sharpening filter for the following motion blurred image?

(11)   Why do we need to combine different filtering approaches for image enhancement?