Chapter 15.
Fundamental Principles of Digital Image Processing

15-1. Introduction

- Digital image processing in general involves the use of computers for manipulating digital images in order to improve their quality and/or modify their appearance

- In digital image processing, the digital number of each pixel in an original image is input to a computer, with its inherent row and column location

- The computer operates on the digital number according to some preselected mathematical function or functions, and then stores the results in another array which represents the new or modified image

- When all pixels of the original image have been processed in this manner and stored in the new array, the result is a new digital image

15-1. Introduction

- One types of digital image processes falls under the general heading of *preprocessing operations*

- These are generally aimed at correcting for distortions in the images which stem from the image acquisition process, and they include corrections for such conditions as scanner or camera imperfections and atmospheric refraction

- Another type of digital image processing, called **image enhancement**, has as its goal, the <u>improvement of the visual quality of images</u>

- **Image enhancement** makes interpretation and analysis of images easier, faster, and more accurate; and thus <u>it can significantly improve the quality of photogrammetric products developed from digital images, and reduce the cost of producing them</u>

- **Digital orthophotos** in particular benefit significantly from the improved image quality that results from image enhancements

15-1. Introduction

- A third type of digital image processing, called **image classification**, attempts to replace manual human visual analysis with automated procedures for recognizing and identifying objects and features in a scene
- **Image classification** processes have been widely used in a host of different interpretation and analysis applications, as well as in the production of a variety of thematic maps

- A final type of digital image processing, ***data merging***, combines image data for a certain geographic area with other geographically referenced information in the same area
- The procedures may overlay multiple images of the same area taken at different dates—a technique which is very useful in identifying changes over time, such as monitoring a forest fire or following the spread of a disease in a certain tree species
- The procedures can also combine image data with nonimage data such as DEMs, land cover, and soils
- These types of digital image processing are extremely important in the operation of geographic information systems

15-2. The Digital Image Model

- To comprehend the subject of digital image processing, an understanding of the concept of the digital image model is useful
- A digital image model can be considered as a mathematical expression which yields image density as a function of x, y (row, column) position
- Digital images are imperfect renditions of the object scenes they portray, and their imperfections stem from a host of sources, including the imaging system, signal noise, atmospheric scatter, and shadows
- The primary degradation of an image is a combined systematic blurring effect resulting from aberrations of the lens, resolution of the recording medium (e.g., CCD Array/film), and, to some extent, atmospheric scatter
- The effect of these combined factors can be specified in terms of the *point-spread function*, that can be thought of as the blurred image that would result from a perfect point source of light such as a star
- This point spread function can be represented as a mathematical expression which models the systematic image imperfection, and which was applied to the ideal image through a process known as *convolution*

15-2. The Digital Image Model

❖ The cumulative effect of the systematic image degradations can be represented in a general sense by

$$I(x,y) = O(x,y) * P(x,y) + N(x,y) \qquad (15\text{-}1)$$

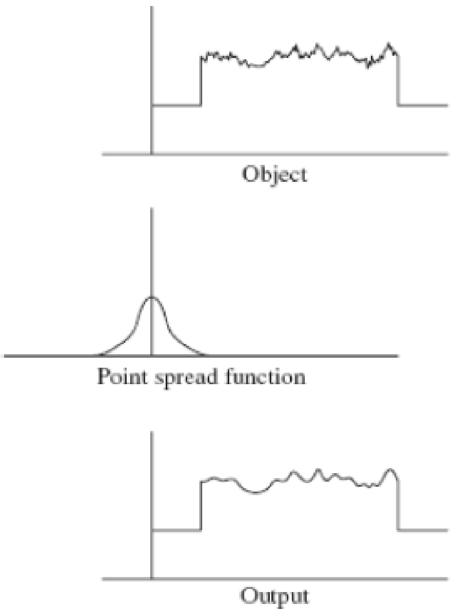


Figure 15-1. The effect of convolution of an ideal image with the point spread function.

❖ In Eq. (15-1), $I(x,y)$ is the actual image model as represented by the digital image, $O(x,y)$ is the theoretical ideal image model of the object, $P(x,y)$ is the point-spread function, $N(x,y)$ is signal noise, and $*$ (asterisk) indicates the convolution operation

❖ Figure 15-1 illustrates convolution of the ideal image model by the point spread function

❖ Notice how the high frequency features in the original object are are smoothed in the output image

❖ Spatial frequency is explained in the next section

❖ One goal of image processing is to negate the effects of image noise and the point-spread function to recover the theoretical ideal image

15-2. The Digital Image Model

$$I(x,y) = O(x,y) * P(x,y) + N(x,y) \qquad (15\text{-}1)$$

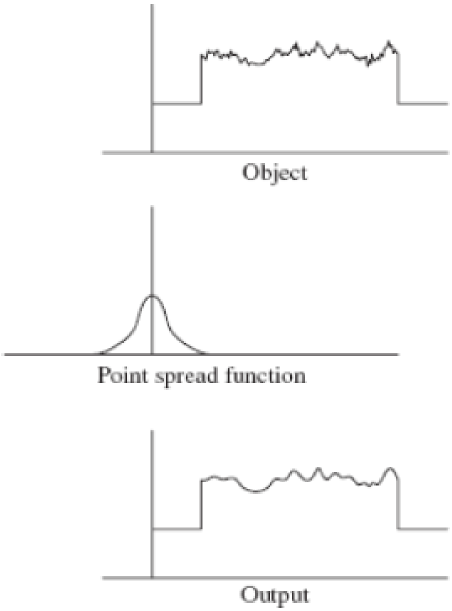


Figure 15–1. The effect of convolution of an ideal image with the point spread function.

- While there are techniques known as *optimal filters* that can negate these detrimental effects to some extent, their implementation is complex and requires an in-depth knowledge of signal processing theory
- On the other hand, simpler methods are also available which can at least partially compensate for these effects
- Some of these methods can reduce image noise, and others can reduce the blurring caused by the point-spread function through edge enhancement

15-3. Spatial Frequency of a Digital Image

- ❖ The concepts of pixels as discrete samples of the image at a specific geometric resolution, as well as the quantization of brightness values are important in understanding digital image processing
- ❖ Another notion that is key to the comprehension of digital image processing is that of *spatial frequency*
- ❖ Objects (e.g., terrain features) have a virtually unlimited range of spatial frequencies
- ❖ Imagine, for example, an ideal image of a cornfield, where the rows are nominally spaced 1 m apart
- ❖ On a small-scale image (e.g., high-altitude air photo), the variations from light to dark to light, etc. (corresponding to corn, soil, corn, etc.) might be barely discernible
  - ➪ The spatial frequency at object scale associated with this variation would be 1 cycle per meter

15-3. Spatial Frequency of a Digital Image

- A medium-scale image (e.g., low-altitude air photo) of the same cornfield might reveal variations from light to dark between individual leaves of the plants
  - If the leaves were nominally spaced 10 cm (0.1 m) apart, the corresponding spatial frequency would be 1 cycle per 0.1 m, or 10 cycles per meter
- A large-scale image (e.g., close-up photo taken with a handheld camera) might reveal variations associated with individual kernels on an ear of corn
  - If the kernels were nominally spaced 5 mm (0.005 m) apart, the corresponding spatial frequency would be 1 cycles per 0.005 m, or 200 cycles per meter
- If an individual kernel were viewed under a microscope, even higher spatial frequencies would be discernible
- The point of this example is that a multitude of different frequencies exists in objects in nature

15-3. Spatial Frequency of a Digital Image

- ❖ Although there are many different frequencies in a natural scene, most of the higher frequencies will be lost when a digital image is acquired
- ❖ This occurs whether the image is exposed directly by a digital sensor or is obtained by scanning a film-based photograph
- ❖ In the process of obtaining digital images, brightness variations that occur within an individual pixel will be averaged together, thus attenuating high frequencies
- ❖ This is a consequence of the physical size of the pixel being sampled
- ❖ The highest frequency that can be represented in a digital image is the *Nyquist frequency*, which is one-half the sampling frequency
- ❖ As an example, in a satellite image with pixels having a 10-m ground resolution, the sampling frequency is 1 sample per 10 m, or 0.1 sample per meter

⇨ The highest spatial frequency (Nyquist frequency) that can be represented in this image is therefore 0.05 cycles per meter, or 1 cycle per 20 m

⇨ Note that 1 cycle corresponds to two samples

15-3. Spatial Frequency of a Digital Image

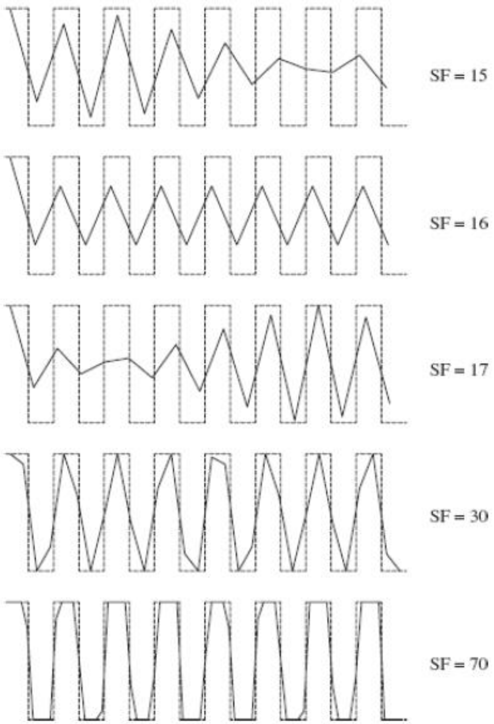


Figure 15–2. The effect of different sampling frequencies in the depiction of a signal with a frequency of eight cycles per second.

- ❖ Figure 15-2 illustrates the effect of using different sample frequencies to represent a given signal
- ❖ The signal frequency is eight cycles per unit
- ❖ Notice that a sampling frequency of 16 samples per unit is the minimum to fully represent the signal
- ❖ **The Nyquist frequency** is the upper limit for the modulation transfer function
  - ➪ This is illustrated in Fig. 15-3

- ❖ Since the sampling frequency in the figure is 160 samples per mm, the modulation transfer function is not applicable above the Nyquist frequency, 80 cycles per mm

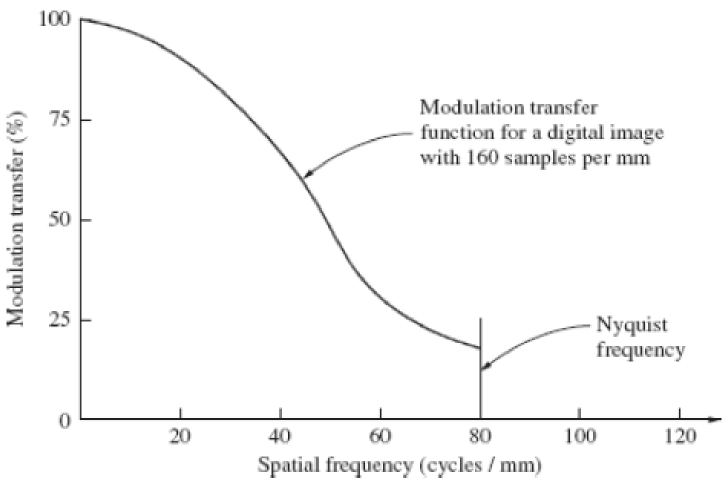


Figure 15–3. Relationship between the Nyquist frequency and the modulation transfer function.

15-3. Spatial Frequency of a Digital Image

- The relationship between the variations of digital numbers in a digital image and the spatial frequencies they represent can be precisely quantified in a mathematical sense
- Often, certain characteristics of a digital image are more logically described in terms of spatial frequency than by digital numbers in an image
- In sections that follow, explanations will be given which should help to clarify this relationship
- However, a full understanding of the concept will require substantial study in the area of signal processing theory, which is beyond the scope of this text

15-4. Contrast Enhancement

❖ Contrast enhancement is a employed type of digital image processing operation

❖ When each pixel of a scene is quantized, a digital number that corresponds to the brightness at its location is assigned

❖ In an area where the overall scene brightness is low (e.g., a shadow region), the majority of the pixels will have low values

❖ In such cases, the contrast between features will be diminished

❖ To get an impression of the overall brightness values in an image, it is useful to produce a *histogram* of the digital numbers

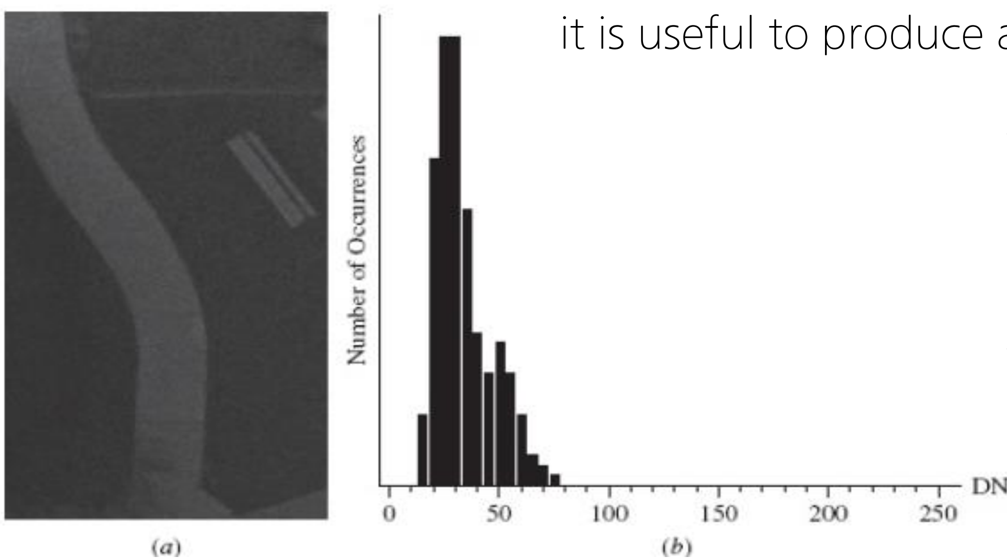


Figure 15-4. (a) Digital image of an area in a shadow. (b) Histogram of digital numbers from the image.

❖ The histogram is a plot of digital number on the abscissa versus number of occurrences on the ordinate

❖ As an example, if a digital image has 1907 pixels having a digital number (DN) of 29, the histogram will show an ordinate of 1907 at the abscissa of 29

15-4. Contrast Enhancement

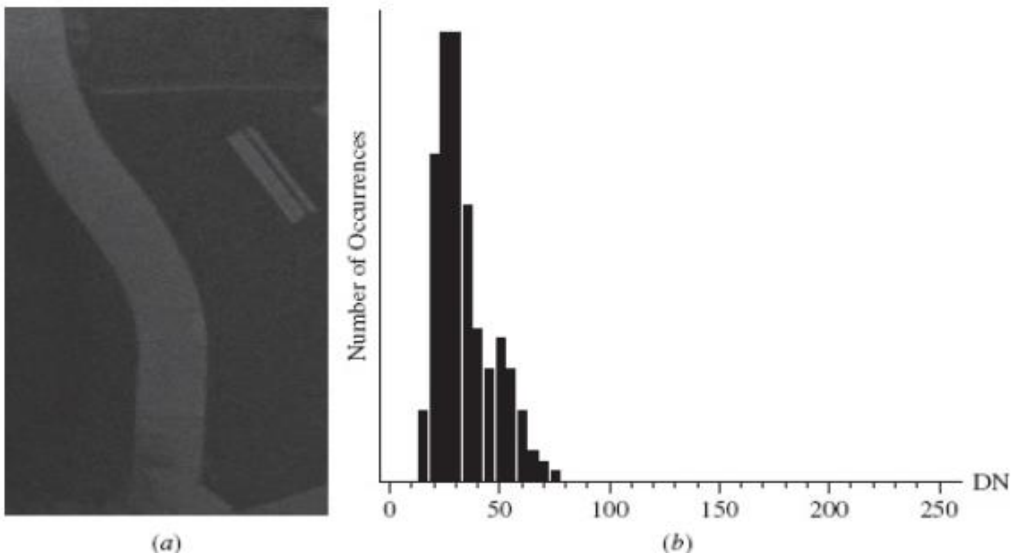


Figure 15–4. (a) Digital image of an area in a shadow. (b) Histogram of digital numbers from the image.

- Figure 15-4a shows a small portion of a digital image which is in the shadow of a building, and Fig. 15-4b shows its corresponding histogram of digital numbers
- The image contains a sidewalk and a bench which are barely noticeable because of the low contrast caused by the shadow
- Notice from the histogram that the majority of digital numbers falls in the range of 20 to 65, even though the available range is 0 to 255
- This indicates that when the image is displayed, the contrast among features in this area will be low, as is certainly evident in Fig. 15-4a

15-4. Contrast Enhancement

- A simple method of increasing the contrast is to apply a linear stretch
- With this operation, the digital numbers within a certain range (e.g., from 20 to 65) are linearly expanded to the full available range of values (e.g., from 0 to 255)
- The effect of a linear stretch on the image of Fig. 15-4a is illustrated in Fig. 15-5a. Its associated histogram is shown in Fig. 15-5b

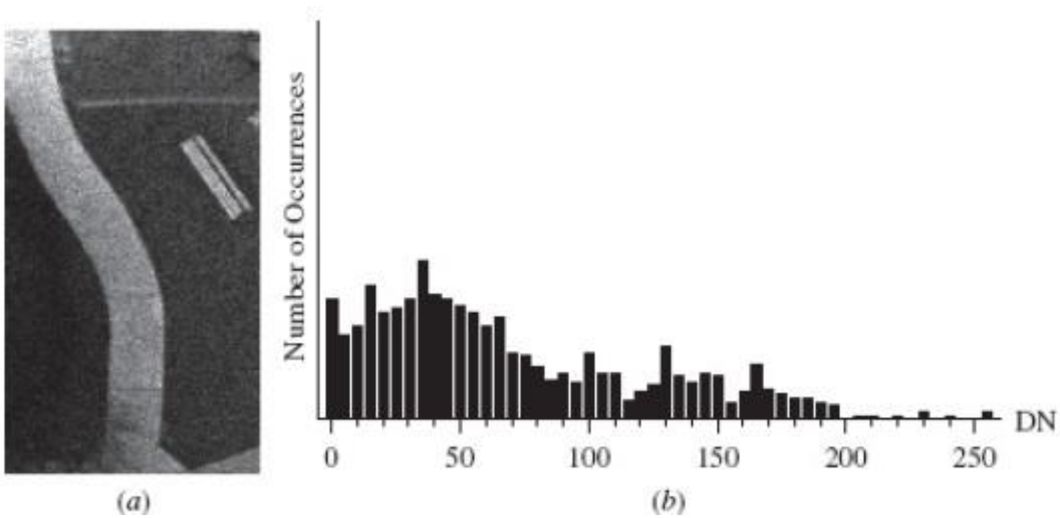
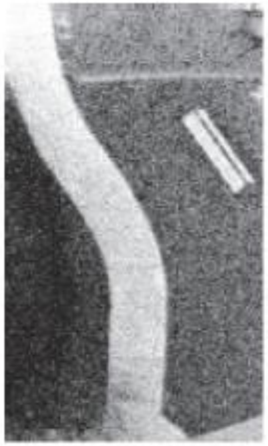


Figure 15-5. (a) Digital image from Fig. 15-4a after linear stretch contrast enhancement. (b) Histogram of digital numbers from the image after linear stretch.

- Notice in this histogram that the digital numbers have been expanded (stretched) so as to encompass the full available range (0 to 255)
- As can be seen in Fig. 15-4a, the effect of this linear stretch is enhanced contrast, making the sidewalk and bench more apparent

15-4. Contrast Enhancement

- A more complex method of increasing the contrast is to apply *histogram equalization*
- The concept behind histogram equalization is to expand the digital numbers in a nonlinear fashion across the available range so that the values are more evenly distributed
- The effect of histogram equalization on the image of Fig. 15-4a is illustrated in Fig. 15-6a
  - ➪ Its associated histogram is shown in Fig. 15-6b



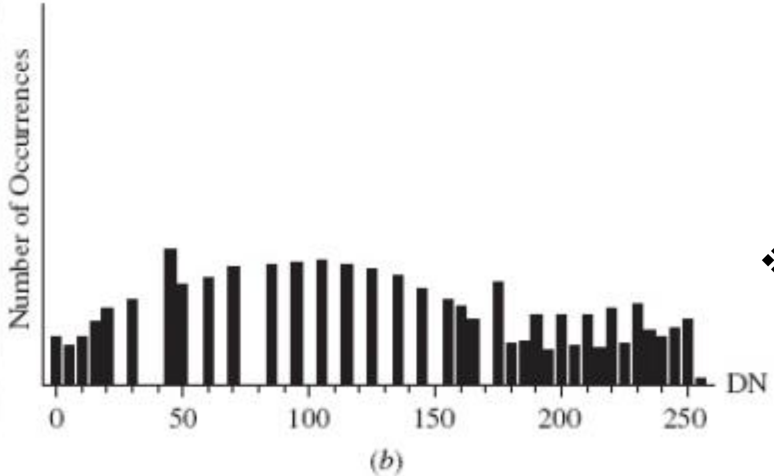


Figure 15-6. (a) Digital image from Fig. 15-4a after histogram equalization contrast enhancement. (b) Histogram of digital numbers from the image after histogram equalization.

- Notice from this histogram that the numbers are more evenly distributed, particularly at the high end of the digital number range
- Note also that the histogram-equalized image of Fig. 15-6a has even greater contrast than that of the linear stretched image, making the sidewalk and bench even more pronounced

15-5. Spectral Transformations

- Digital images capture information at various spatial frequencies and exist in the *spatial domain*
- The term *spatial domain* refers to the concept that the coordinates of pixels in the image relate directly to positions in the two-dimensional space of the image
- Through certain mathematical operations, images can be converted from the spatial domain to the *frequency domain*, and vice versa
- A digital image which has been converted to the frequency domain has pixels whose positions (row and column) relate to frequency instead of spatial locations
- Images in the frequency domain no longer have visibly recognizable features as spatial domain images do
- They are an abstraction of the original spatial image; however, they do contain the entire information content of the original
- In the frequency domain, the spatial frequencies inherent in the native image are represented by measures of amplitude and phase for the various frequencies present in the image

15-5. Spectral Transformations

- The most commonly used method for converting a digital image to the frequency domain (or vice versa) is the *discrete Fourier transform* (DFT)
- This process is roughly analogous to conversion between geodetic and geocentric coordinates
- Converting from geodetic to geocentric coordinates changes the representation of a point's position from latitude, longitude, and height; to $X, Y,$ and Z
- In an analogous fashion, the DFT converts the brightness values from a (spatial domain) digital image to a set of coefficients for sine and cosine functions at frequencies ranging from 0 to the Nyquist frequency
- In an additional step, amplitude and phase information can be obtained from these sine and cosine coefficients
- A more detailed explanation of the DFT requires the use of complex numbers, which is beyond the scope of this text

15-5. Spectral Transformations

- The discrete Fourier transform has become a useful tool for analyzing digitally sampled periodic functions in one, two, or even more dimensions
- A particularly efficient implementation of the discrete Fourier transform is the aptly named *fast Fourier transform* (FFT)
- Since its discovery and implementation in the 1960s, it has enabled discrete Fourier transforms to be computed at a greatly enhanced speed
- This greatly enhanced computational speed has led to wider use of the Fourier transform as an image processing tool

15-5. Spectral Transformations

- Figure 15-7a shows a digital satellite image of a portion of Gainesville, Florida, with its Fourier transform (amplitudes only) shown in Fig. 15-7b
- In Fig. 15-7b, greater amplitudes correspond to brighter tones, and lesser amplitudes correspond to darker tones
- This Fourier transform exhibits a star-shaped pattern which is typical of two-dimensional transforms of urban scenes

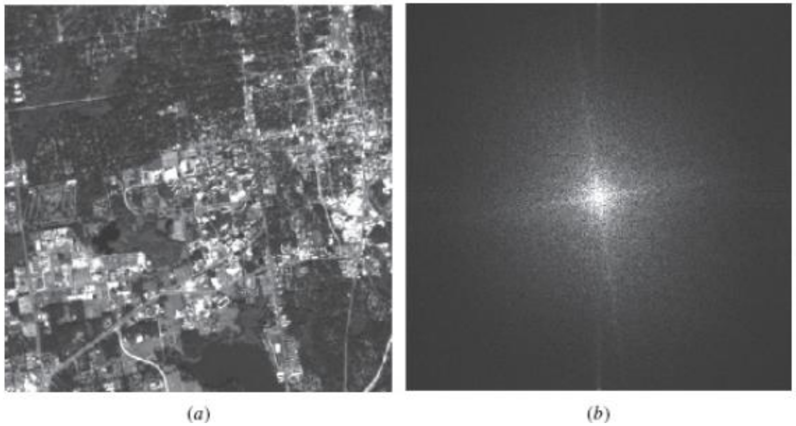


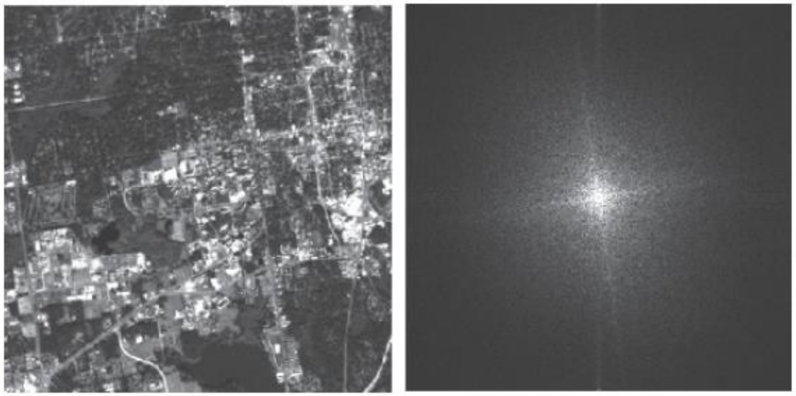
Figure 15-7. (a) Digital satellite image of Gainesville, Florida. (b) Fourier transform. (Courtesy University of Florida.)

- The lowest spatial frequencies are represented at the center of the star with frequency increasing with increasing distance from the center

15-5. Spectral Transformations

❖ It can be seen from Fig. 15-7b that lower frequencies have greater magnitudes than higher frequencies

❖ The figure also shows discernible spikes radiating from the center



(a) (b)

Figure 15-7. (a) Digital satellite image of Gainesville, Florida. (b) Fourier transform. (Courtesy University of Florida.)

❖ Those radiating in vertical and horizontal directions are due to discontinuities associated with the edges of the image, whereas the spikes oriented approximately 10° counterclockwise are due to the edges associated with the primary streets and buildings in the image

15-5. Spectral Transformations

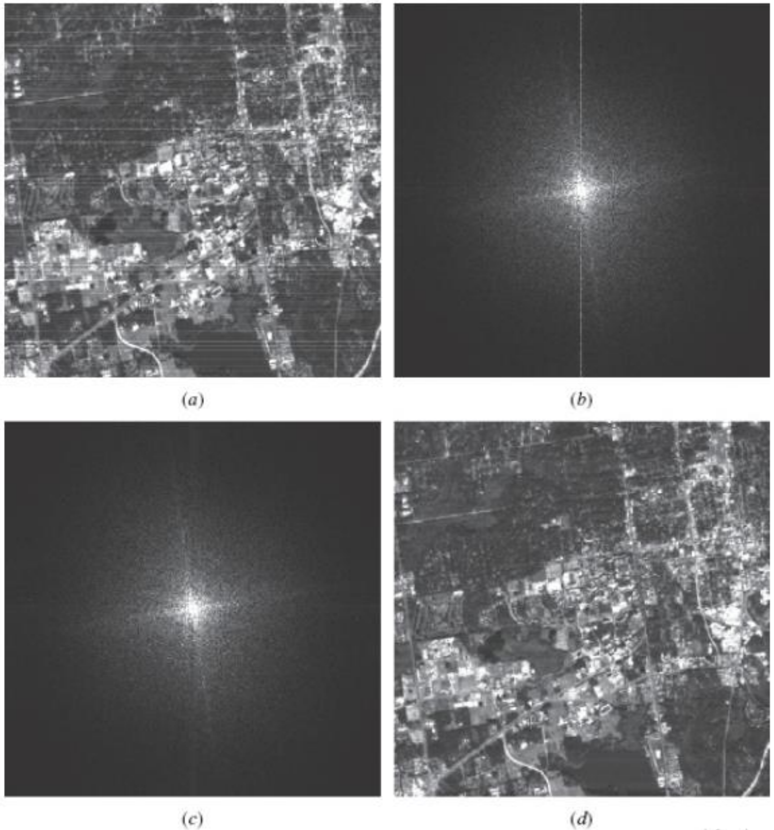


Figure 15-8. (a) Gainesville image with interference noise. (b) Fourier transform. (b) Fourier transform with high-frequency interference eliminated. (d) Cleaned image after inverse Fourier transform of (c). (Courtesy University of Florida.)

- Certain operations are easier to accomplish in the frequency domain than in the spatial domain
- For example, assume a digital sensor had an electrical problem which caused an interference pattern in the image as illustrated in Fig. 15-8a
- Conversion of this contaminated image from the spatial to the frequency domain reveals high amplitudes for the particular frequencies which contributed to the interference, as shown in Fig. 15-8b

15-5. Spectral Transformations

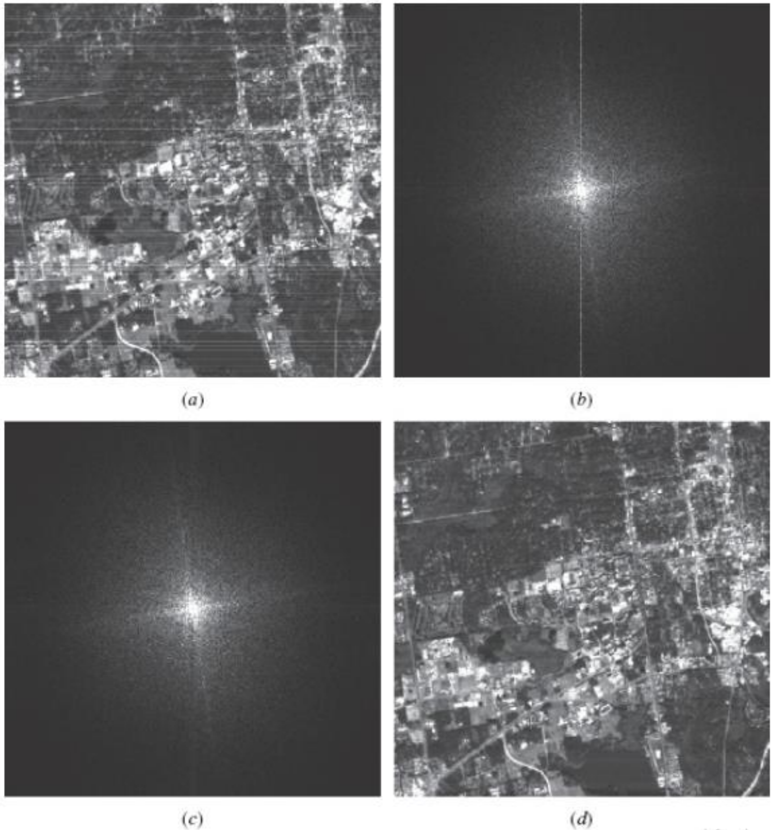


Figure 15–8. (a) Gainesville image with interference noise. (b) Fourier transform. (b) Fourier transform with high–frequency interference eliminated. (d) Cleaned image after inverse Fourier transform of (c). (Courtesy University of Florida.)

- The coefficients corresponding to the sine and cosine terms for those frequencies can be changed from the high values to zero values as illustrated in Fig. 15-8c, thus eliminating the interference in the frequency domain image
- To view the image, an inverse Fourier transform would then be applied to the modified frequency image to convert back to the spatial domain, as illustrated in Fig. 15-8d
- Since the systematic interference was eliminated in the frequency domain, it is no longer present in the spatial domain image
- The interference removal was not perfect, however, as can be seen in the subtle interference patterns which remain in Fig. 15-8d

15-5. Spectral Transformations

- Certain mathematical operations are easier to compute in the frequency domain than the spatial domain
- Another such operation is *convolution* and this process involves two functions: the signal (e.g., digital image) and the response function
- An example of a response function is the mathematical representation of the effect that a real imaging system imparts to an ideal image to produce the actual image
- The two functions are convolved, which will apply the "smearing" effect of the response function to the signal
- Depending on the form of the response function, the convolution will have the effect of filtering out certain high frequencies from the image

15-5. Spectral Transformations

- In the absence of noise, the inverse operation *deconvolution* can be applied to an image that has been affected by a known response function, to reconstruct the high frequencies which were lost
  - Deconvolution may be done to remove the blurring effect of the point-spread function in order to recover the ideal image from an imperfect real image
  - Deconvolution should be done with caution, however, since it can be highly sensitive to image noise

- Another operation which is easy to compute in the frequency domain is *correlation*
  - Correlation also involves two functions, which can both be assumed to be signals (e.g., digital images)
  - Correlation provides a measure of the similarity of two images by comparing them while superimposed, as well as shifting an integral number of pixels left, right, up, or down, all with one simultaneous operation
  - Correlation can be useful in matching patterns from one image that correspond to those of another image
  - Pattern matching provides the backbone of softcopy mapping systems

15-5. Spectral Transformations

- Fourier transforms, while probably the best known, are not the only image transformations available

- Another transformation is the *wavelet transform*
  - Wavelets are types of functions characterized by an oscillation over a fixed interval
  - They can be used to represent an image in a similar manner as the Fourier transform
  - In practice, a "mother wavelet" serves as a template for several "child" wavelets, which are scaled and translated to match the frequencies in the rows and columns of an image
  - Since all that is needed to represent an image are the scale and translation parameters, wavelets can reduce the amount of storage space for an image by eliminating the coefficients of rarely used frequencies
  - Thus, wavelets are useful in image compression

15-5. Spectral Transformations

- Fourier transforms, while probably the best known, are not the only image transformations available

- Another useful transformation is the Hough transform
  - The Hough transform is primarily used for joining discontinuous edges in an image
  - It can be highly useful for reconstructing linear and circular features from raster images, where the edges have breaks in them
  - Details of these transforms are beyond the scope of this text; however, references cited at the end of this chapter can be consulted for further information

15-5. Spectral Transformations

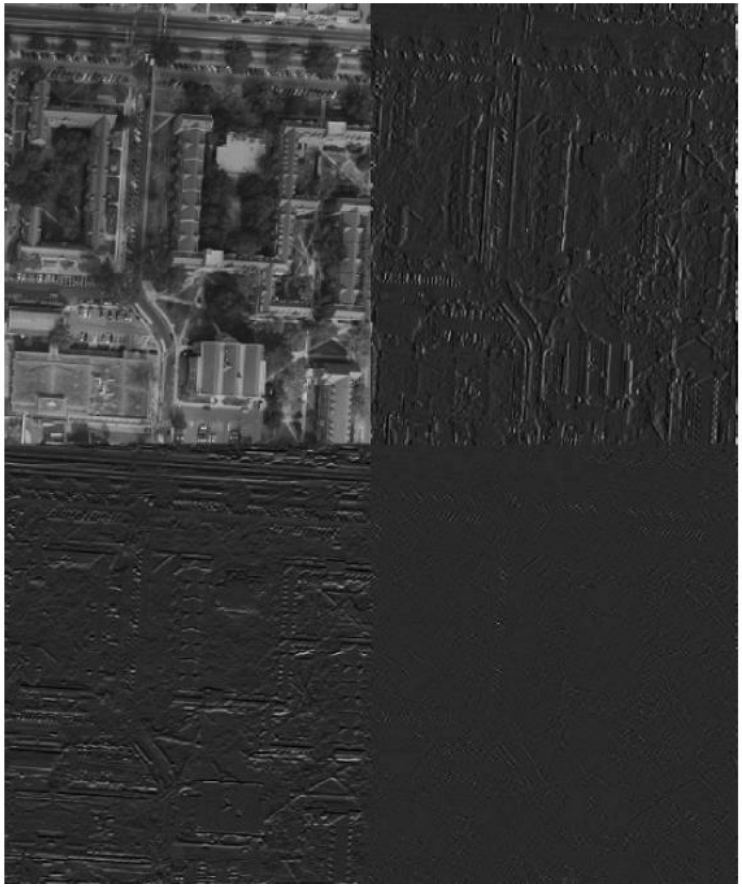


Figure 15-9. The first level of wavelet decomposition for an aerial image.

- Figure 15-9 shows a first-level wavelet decomposition
- In the upper left corner is an image at half its original resolution (low frequency components). The three other images contain the high frequency components removed when decreasing the resolution
- The upper-right hand image contains the high frequencies components in the horizontal direction, the lower-left hand image contains the components associated with high frequencies in the vertical direction, and the lower-right hand image contains the components for diagonal high frequencies
- These four sub-images can be used to perfectly recreate the image in its original resolution

15-5. Spectral Transformations

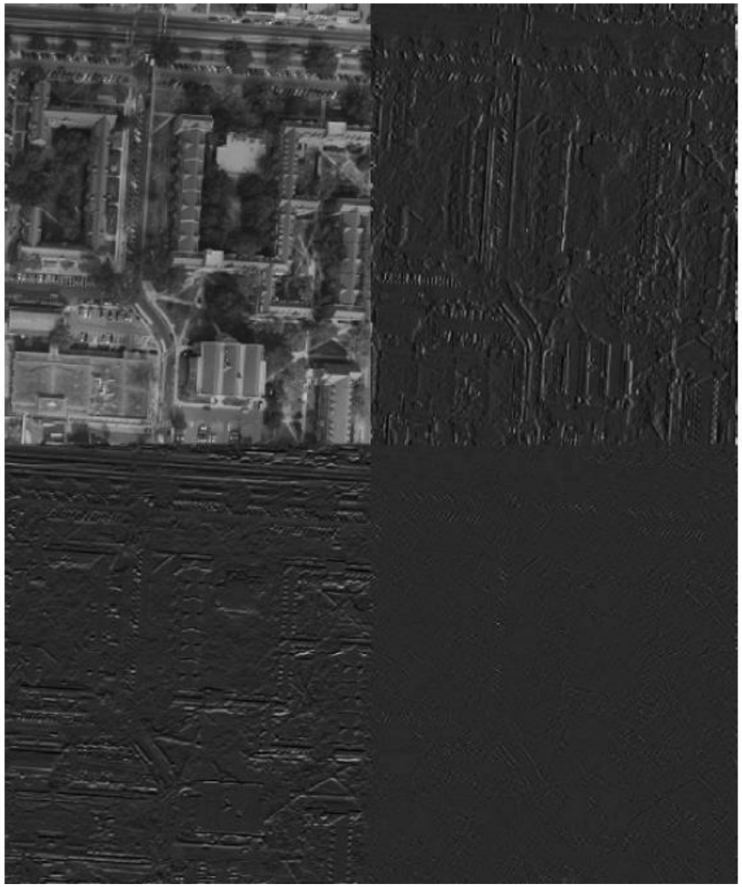


Figure 15-9. The first level of wavelet decomposition for an aerial image.

- The full wavelet transformation entails iteratively reducing the resolution of previous-level images and storing the higher frequency components until the lower resolution image is one pixel
- Similar to the Fourier transform, wavelets also allow the smoothing of imagery by withholding unwanted frequencies when recreating an image
- Wavelets can also be used in many other image processing algorithms such as edge detection

15-6. Moving Window Operations

- The Fourier transform is highly effective for performing convolution of a digital image with a response function
- A simpler method known as the *moving window* can also be used for performing convolution
- The moving window is particularly useful when the response function is highly localized
- For example, a two-dimensional response function may consist of a 3×3 submatrix of nonzero terms contained within a large array which has zero terms everywhere else
- When convolving using the Fourier transform, the image and response function must have the same dimensions, which can lead to inefficiency when the vast majority of terms in the response function are equal to zero
- In cases such as this, the moving window can be used efficiently

15-6. Moving Window Operations

- The moving window operation has two primary inputs: the original digital image and a localized response function known as a *kernel*
- The kernel consists of a set of numbers in a small array, which usually has an odd number of rows and columns
- The kernel is conceptually overlaid with a same-size portion of the input image (image window), and a specific mathematical operation is carried out
- The value resulting from this operation is placed in the center of the corresponding location in the output image, the kernel is shifted by one column (or row), and another value is computed
- This procedure is repeated until the entire output image has been generated

15-6. Moving Window Operations

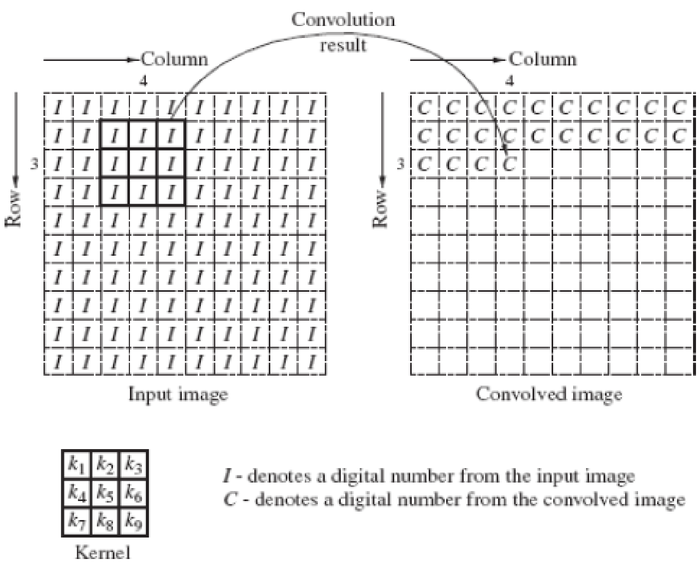


Figure 15-10. Moving-window image convolution.

- Figure 15-10 illustrates one step of the procedure
  - In this figure, the 3 × 3 kernel is centered on pixel (row = 3, column = 4) in the input image
- After the convolution is performed, the result is placed in pixel 3,4 of the output (convolved) image
  - **The kernel** is then shifted so that it is centered on pixel 3,5, and the convolution is repeated
- When convolving pixels at the edge of the original image, the kernel will extend outside the image boundary
- Under the assumption that the image is a periodic function, **the kernel** values that extend beyond the image will "**wrap-around**" so as to overlay with image pixels at the opposite side of the image
- **The wrap-around effect** can be visualized by imagining exact copies of the image to exist above, to the right, below, to the left, and diagonally away from each of the four corners of the original image
- This is the same result that would have occurred if the Fourier transform had been used to perform the convolution

15-6. Moving Window Operations

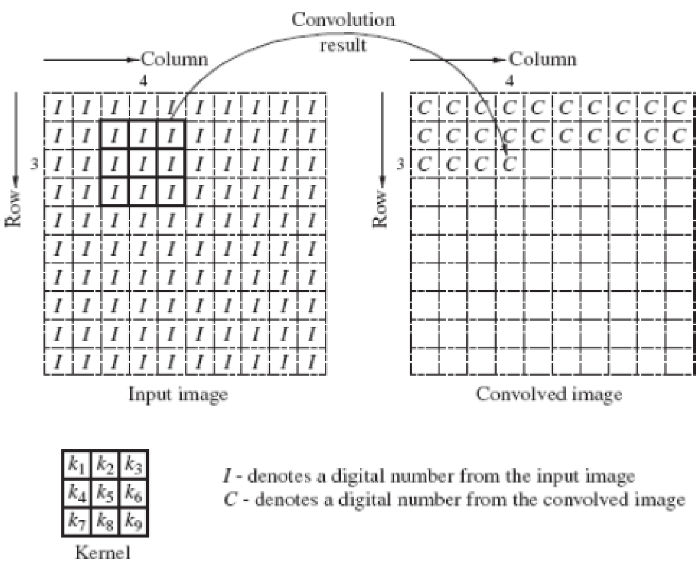


Figure 15-10. Moving-window image convolution.

- The mathematical operation of convolution is accomplished through a series of multiplications and additions
- For example, the convolution result depicted in Fig. 15-10 is computed by

$$C_{34} = k_1 \times I_{23} + k_2 \times I_{24} + k_3 \times I_{25} + k_4 \times I_{33} + k_5 \times I_{34} + k_6 \times I_{35} + k_7 \times I_{43} + k_8 \times I_{44} + k_9 \times I_{45} \quad (15\text{–}2)$$

- In Eq. (15-2), the k values are the individual kernel elements, the I values are digital numbers from the input image with the appropriate row and column indicated by their subscripts, and C_{34} is the convolution result which is placed at row = 3, column = 4 in the output image

15-6. Moving Window Operations

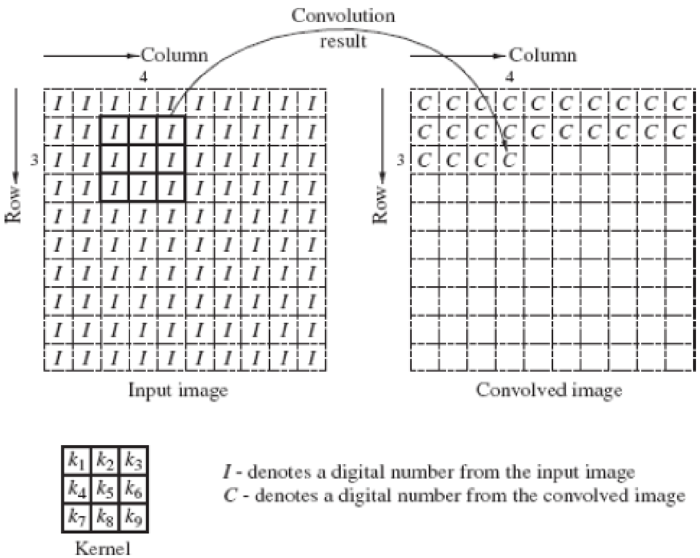


Figure 15-10. Moving-window image convolution.

$$C_{34} = k_1 \times I_{23} + k_2 \times I_{24} + k_3 \times I_{25} + k_4 \times I_{33} + k_5 \times I_{34} + k_6 \times I_{35} + k_7 \times I_{43} + k_8 \times I_{44} + k_9 \times I_{45} \qquad (15\text{-}2)$$

❖ Perhaps the simplest form of convolution kernel is one that computes the average of the nearby pixels

❖ This type of convolution is known as a *low-pass filter*, so called because <u>the averaging operation attenuates high frequencies, allowing low frequencies to be passed on to the convolved image</u>

❖ The kernel values used in a simple 3 × 3 low-pass filter are shown in Eq. (15-3)

$$K = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \qquad (15\text{-}3)$$

❖ In this equation, K is the kernel matrix, which in this case has all elements equal to $\frac{1}{9}$

15-6. Moving Window Operations

$$K = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \quad (15-3)$$

- Convolution of this kernel matrix with a corresponding 3 × 3 image submatrix is equivalent to computing a weighted average of the digital numbers within the submatrix
- Notice that the sum of the elements in this kernel matrix equals 1, which is appropriate when computing a weighted average
- In fact, elements of a kernel can be considered to be the weights used in computing a weighted average
- If the sum of the elements of the kernel matrix does not equal 1, the convolution result can be divided by the kernel's sum before placement into the output image

15-6. Moving Window Operations

- The result of a low-pass filter convolution is illustrated in Figs. 15-11a and b. Figure 15-11a shows the original image of cars in a parking lot
- A low-pass filter was applied using a 5 × 5 convolution kernel which computed a simple average, resulting in the image of Fig. 15-11b
- Notice that the image of Fig. 15-11b now appears somewhat blurred since the highest-frequency detail has been filtered out from the original image
- By applying kernels of larger size (say, 7 × 7 or 9 × 9) a wider range of high-frequency information will be filtered from the image

(a) (b)

Figure 15–11. (a) Parking lot image. (b) Parking lot image after applying a low-pass filter.

- The ultimate limit in kernel size would be the same as that of the original image
- Such a convolution would result in all pixels of the output image having the same value—the overall average of digital numbers from the input image
  - In that case, detail at all frequencies will have been filtered out, leaving a uniformly gray image

15-6. Moving Window Operations

- ❖ The intentional blurring of an image as the result of applying a low-pass filter may seem like a counterproductive operation
- ❖ After all, high-frequency detail assists in identifying features and measuring the positions of their edges
- ❖ The method could serve as a simple high-frequency noise filter, but there are better methods available for this purpose
- ❖ However, another use for low-pass filtering is to employ the process as a precursor to a *high-pass operation*
- ❖ By performing a pixel-by-pixel subtraction of the low-passed image from the original image, the result will give the high-frequency detail which was filtered out in the low-pass operation

15-6. Moving Window Operations

- The intentional blurring of an image as the result of applying a low-pass filter may seem like a counterproductive operation
- After all, high-frequency detail assists in identifying features and measuring the positions of their edges
- The method could serve as a simple high-frequency noise filter, but there are better methods available for this purpose
- However, another use for low-pass filtering is to employ the process as a precursor to a high-pass operation

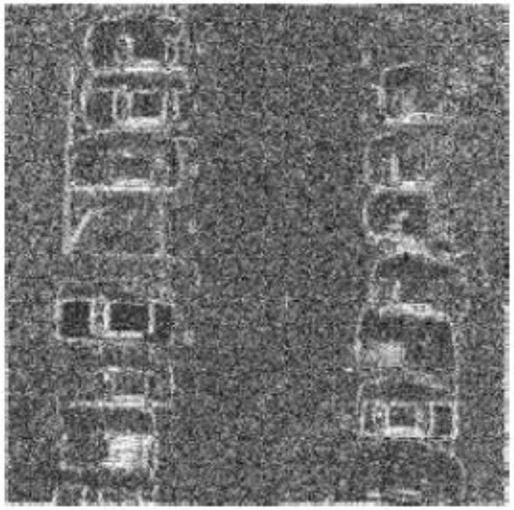


Figure 15-12. Parking lot image after application of a simple high-pass filter.

- Figure 15-12 shows the high-pass filtered result from subtracting the digital numbers of Fig. 15-11b from those of Fig. 15-11a and taking the absolute value
- As is apparent from the figure, this simple high-pass filter can be used to detect edges in the original image

15-6. Moving Window Operations

- Moving window operations can be used to perform other useful operations, such as noise filtering
- One such noise filter method is known as *median filtering*
- In this approach, a convolution is not performed in the normal fashion
- Rather, a 3 × 3 moving window is passed through the input image, and the 9 pixels in the immediate neighborhood are extracted at each step
- The nine digital numbers are then sorted, and the median (middle) value is placed in the corresponding location in the output image
- By using the middle value rather than the average, the median filter will not be sensitive to any extremely high or low value which may be the result of image noise

15-6. Moving Window Operations

(a) (b)

Figure 15–13. (a) Parking lot image with noise. (b) Parking lot image after application of median filter.

- The result of median filtering is shown in Figs. 15-13a and b
- Figure 15-13a shows the same image of a parking lot shown in Fig. 15-11a, except that for purposes of this example, random "salt and pepper" noise has been added
- After the median filter operation is applied, the image of Fig. 15-13b results
- Note that the noise has indeed been eliminated, at the expense of a subtle loss of high-frequency information
- Median filters are useful <u>for removing many forms of random noise, but are not as effective at removing systematic noise</u>

15-6. Moving Window Operations

- Another class of operations which uses the moving window approach is edge detection
- Earlier in this section, a simple form of edge detection based on high-pass filtering was presented
- Other methods exist for edge detection, and two specific ones are presented here: the laplacian and Sobel operators

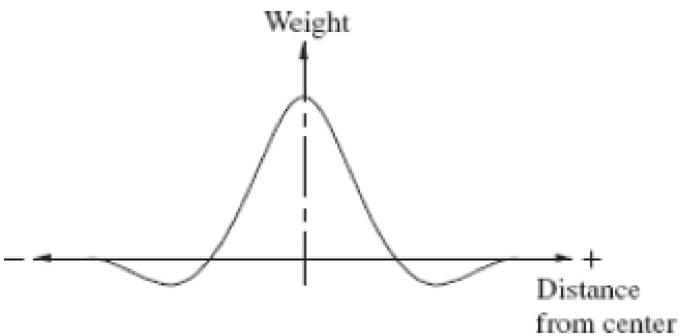


Figure 15-14. Laplacian weighting function.

- The laplacian operator is a convolution using a kernel which has values corresponding to the shape of the function shown in Fig. 15-14, except that it extends in two dimensions
- In Fig. 15-14, note that at the center, the Laplacian starts with a positive value and drops off to negative values at a certain distance from the center

- The effect of this type of kernel is to amplify the differences between neighboring digital numbers which occur at an edge

15-6. Moving Window Operations

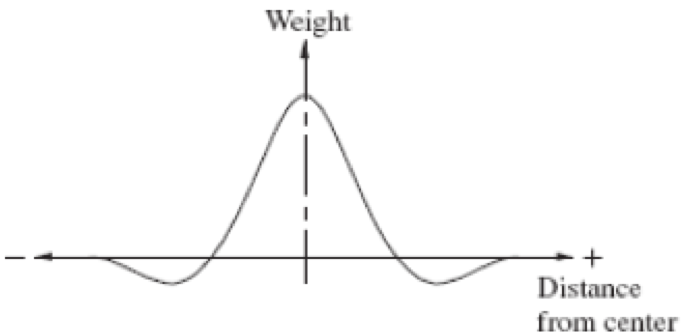


Figure 15−14. Laplacian weighting function.

❖ If the kernel values (weights) are chosen so that their sum is equal to zero, **the laplacian** will result in <u>a high-pass operation, with the lower frequency information filtered out</u>

❖ The result will be an image which contains edge information at the corresponding high frequency

❖ By using a larger laplacian kernel which is flatter and wider, edge information at a particular frequency will be passed to the output image, resulting in a *bandpass* image

❖ **A bandpass image** is <u>an image which contains frequencies within a specific range only</u>

❖ Frequencies outside the range (band) are not present in the bandpass image

❖ Note that since the laplacian kernel contains negative values, the convolution result can be negative, so the absolute value should be taken if the result is to be displayed

15-6. Moving Window Operations

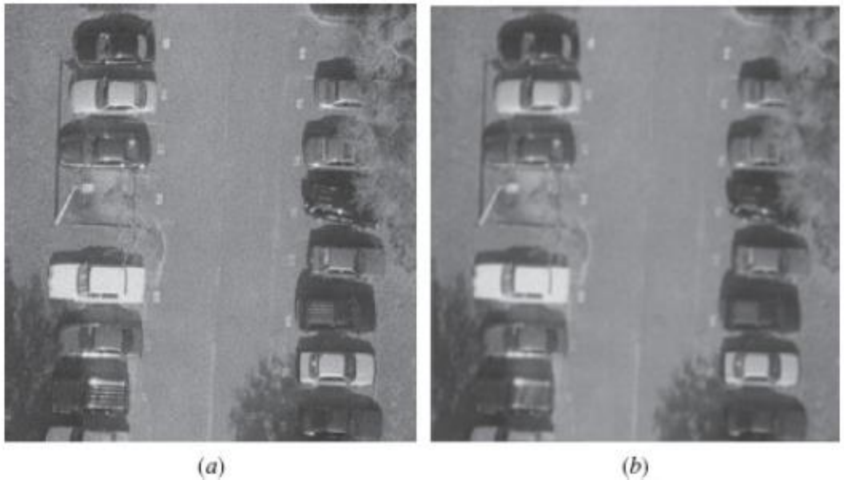


Figure 15-11. (a) Parking lot image. (b) Parking lot image after applying a low-pass filter.

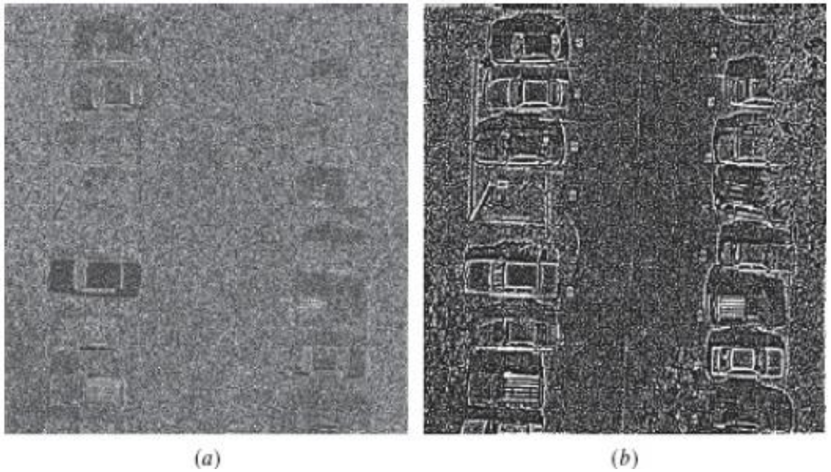


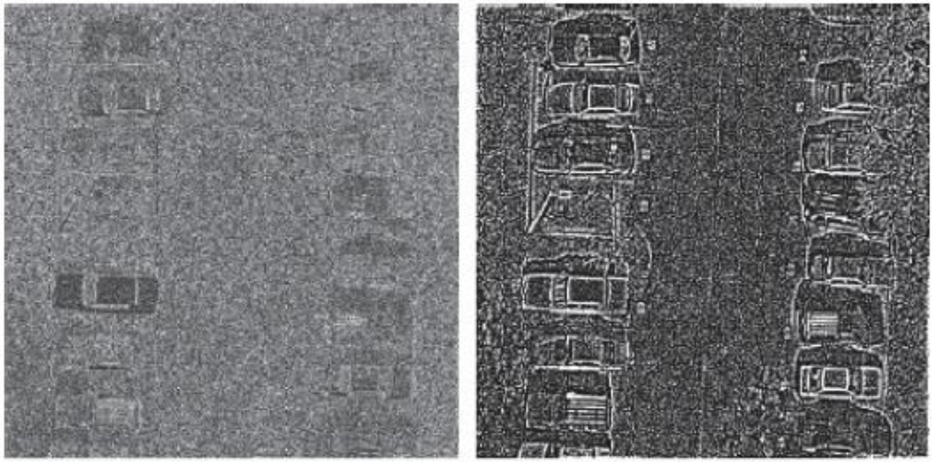
Figure 15-15. (a) Parking lot image after convolution with 3X3 laplacian kernel. (b) Parking lot image after convolution with 5X5 laplacian kernel.

- The results of two laplacian convolutions on the parking lot image of Fig. 15-11a are shown in Figs. 15-15a and b
- The convolution kernel used for Fig. 15-15a has the form shown in Eq. (15-4), and the kernel used for Fig. 15-15b has the form shown in Eq. (15-5)
- Note that the sum of the elements for each of these kernels is equal to zero

$$K = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (15\text{-}4)$$

$$K = \begin{bmatrix} -1 & -2 & -3 & -2 & -1 \\ -2 & 2 & 4 & 2 & -2 \\ -3 & 4 & 8 & 4 & -3 \\ -2 & 2 & 4 & 2 & -2 \\ -1 & -2 & -3 & -2 & -1 \end{bmatrix} \quad (15\text{-}5)$$

15-6. Moving Window Operations



(a) (b)

Figure 15-15. (a) Parking lot image after convolution with 3X3 laplacian kernel. (b) Parking lot image after convolution with 5X5 laplacian kernel.

❖ Notice from Fig. 15-15a that the laplacian operation has enhanced the highest frequencies while filtering out all other frequencies

❖ The resulting edges correspond to the highest frequencies and barely contain recognizable information

❖ Since a wider laplacian kernel was used to produce Fig. 15-15b, the edges that were detected correspond to frequencies that are slightly lower than those of Fig. 15-15a

❖ Notice that the outlines of the cars are readily discernible, as well as the lamppost and its shadow

15-6. Moving Window Operations

- The laplacian edge detector is <u>not sensitive to the orientation of the edge that it detects</u>
- A different edge detection operator, the Sobel, is <u>capable of not only detecting edges, but determining their orientation as well</u>
- This is accomplished by using a pair of kernels (x kernel, y kernel) in the convolution
- The forms of the x and y kernels are given in Eqs. (15-6) and (15-7), respectively

$$K_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \qquad (15\text{–}6)$$

$$K_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \qquad (15\text{–}7)$$

- Convolution by the two kernels at a specific window location gives the x component of the edge, S_x, and the y component of the edge, S_y
- From these two convolution results, the magnitude and direction of the edge can be computed by Eqs. (15-8) and (15-9), respectively

$$\text{Magnitude} = \sqrt{S_x^2 + S_y^2} \qquad (15\text{–}8)$$

$$\text{Direction} = \tan^{-1}\left(\frac{S_y}{S_x}\right) \qquad (15\text{–}9)$$

15-6. Moving Window Operations

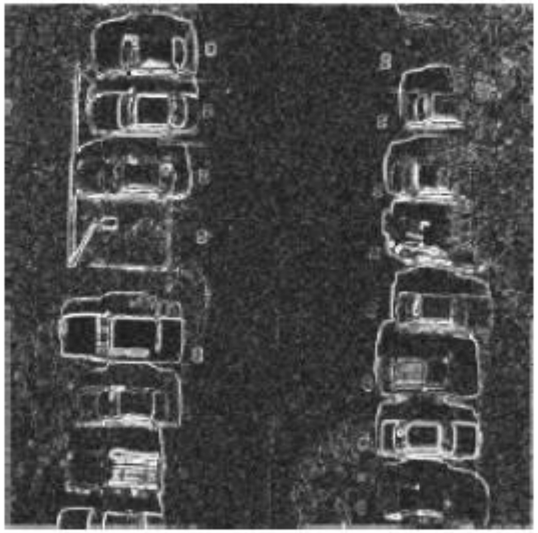


Figure 15-16. Parking lot image after application of Sobel edge detection.

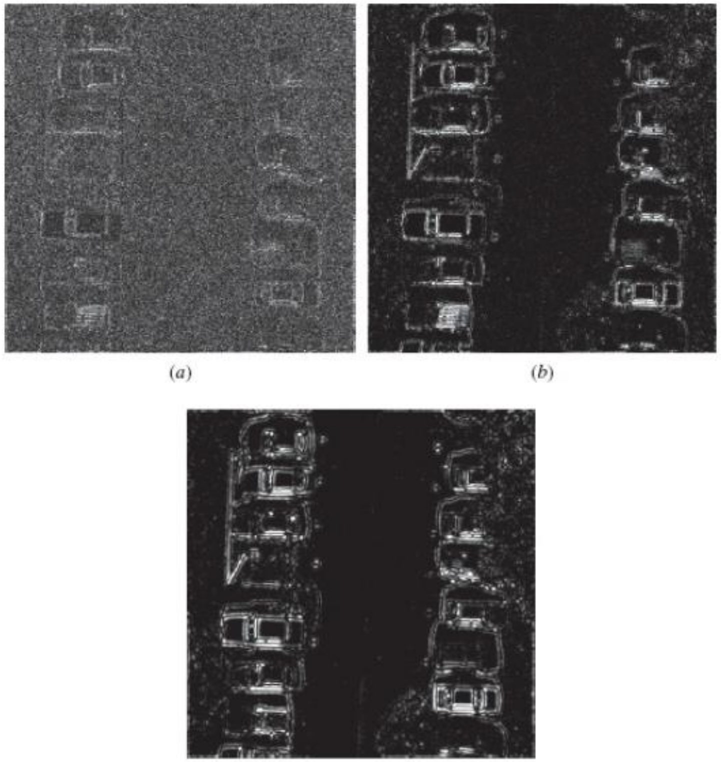
❖ Application of the Sobel edge detection operation to the parking lot image results in the magnitude image shown in Fig. 15-16

❖ Notice that the edges defining the cars and their shadows are well defined, as are the edges defining the lamppost and its shadow

❖ From the examples given on this and the previous page, it can be seen that the size and composition of the kernels applied in digital image processing have a major bearing upon the results achieved, and they also create many interesting possibilities

15-7. Multiscale Representation

- In the preceding section, bandpass filters were discussed. Images that result from a bandpass filter contain edge information that corresponds only to the range of frequencies in the bandpass image
- By using filters which pass lower ranges of frequencies, low-frequency (small-scale) edges can be detected.
- The effect of detecting edges at different scales (frequencies) can be seen in Figs. 15-17a, b, and c
- Figure 15-17a is the result of performing a simple average convolution with a 3 × 3 kernel on the parking lot image, and subtracting the convolved image from the original
- Note that this image contains a great deal of large-scale (high-frequency) edge information



(a) (b) (c)

Figure 15-17. (a) Large-scale edge information from parking lot image. (b) Edge information at a smaller scale. (c) Edge information at a still smaller scale.

15-7. Multiscale Representation

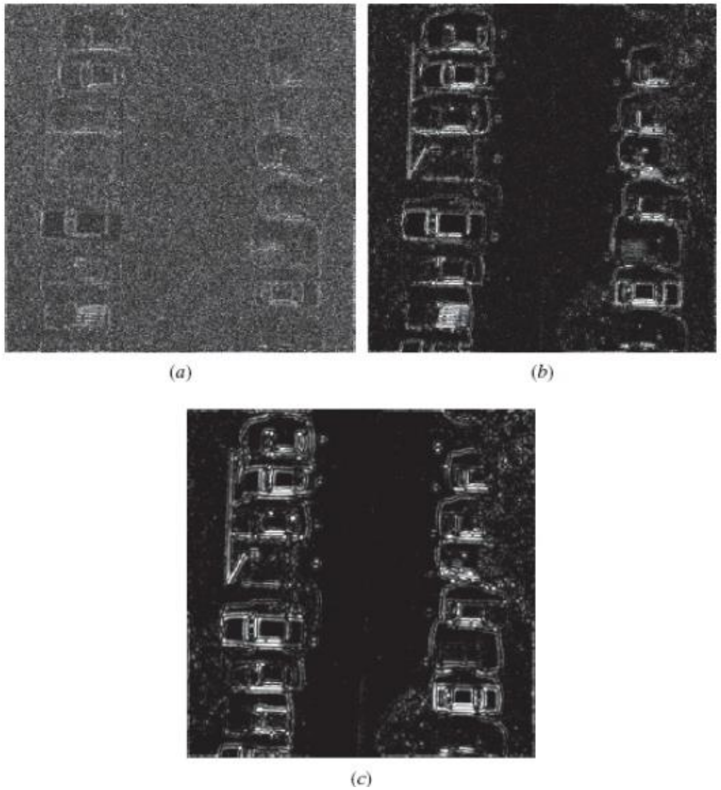


Figure 15-17. (a) Large-scale edge information from parking lot image. (b) Edge information at a smaller scale. (c) Edge information at a still smaller scale.

- Figure 15-17b is the result of two convolutions, followed by a difference
- The first convolution is the same as that used to produce Fig. 15-17a, i.e., a simple average convolution with a 3 × 3 kernel
- Assume that the original parking lot image is image I and the result of the first convolution gives image I
- The second convolution is performed on image I using a simple average convolution with a 5 × 5 kernel, resulting in image I_3

15-7. Multiscale Representation

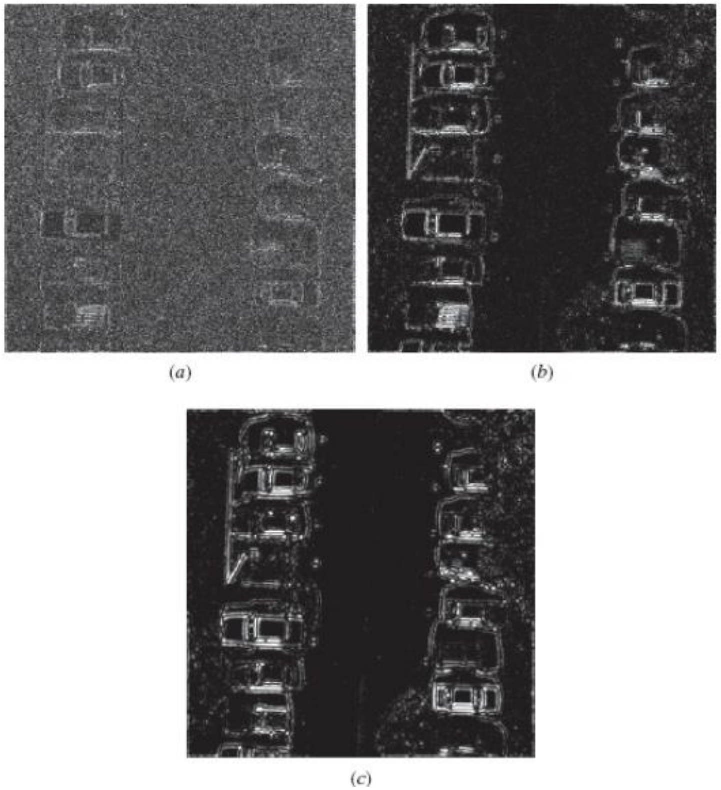


Figure 15-17. (a) Large-scale edge information from parking lot image. (b) Edge information at a smaller scale. (c) Edge information at a still smaller scale.

- Finally, image $I_{3,5}$ is subtracted from image I_3, resulting in Fig. 15-17b
- Notice that this figure contains edge information at a smaller scale (lower frequency) than that of Fig. 15-17a
- Notice also that the edges corresponding to the roof rack on the vehicle at the lower left are still discernible

15-7. Multiscale Representation

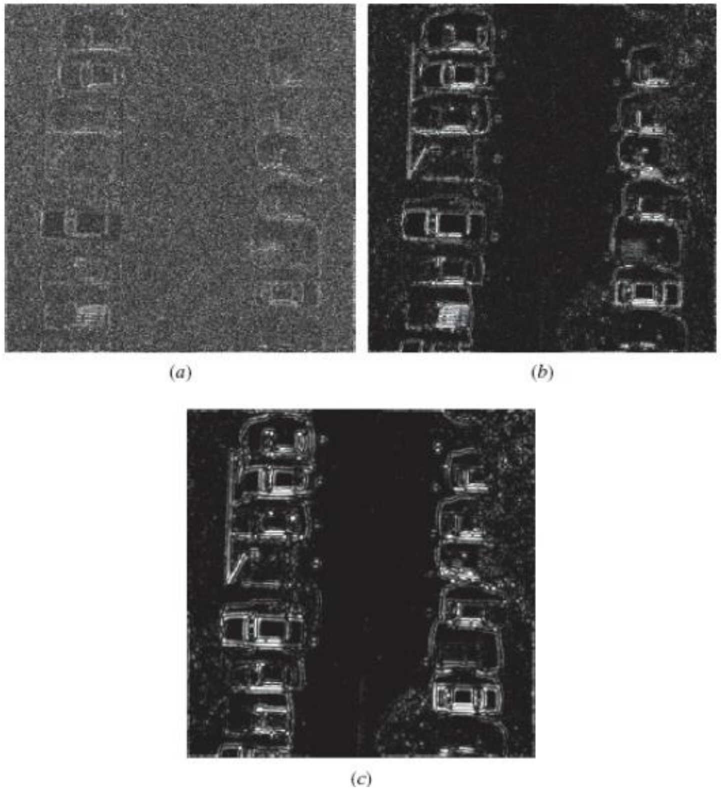


Figure 15-17. (a) Large-scale edge information from parking lot image. (b) Edge information at a smaller scale. (c) Edge information at a still smaller scale.

- The last image, Fig. 15-17c, is the result of three convolutions followed by a difference
- Starting with image $I_{3,5}$ which has been convolved twice, an additional convolution is performed using a 7 × 7 kernel (simple average), resulting in image $I_{3,5,7}$
- Finally, $I_{3,5,7}$ is subtracted from image $I_{3,5}$, resulting in Fig. 15-17c
- In this figure, edges of a still smaller scale (lower frequency) have been detected
- Notice that in this figure, the edges associated with the roof rack have essentially disappeared

15-7. Multiscale Representation

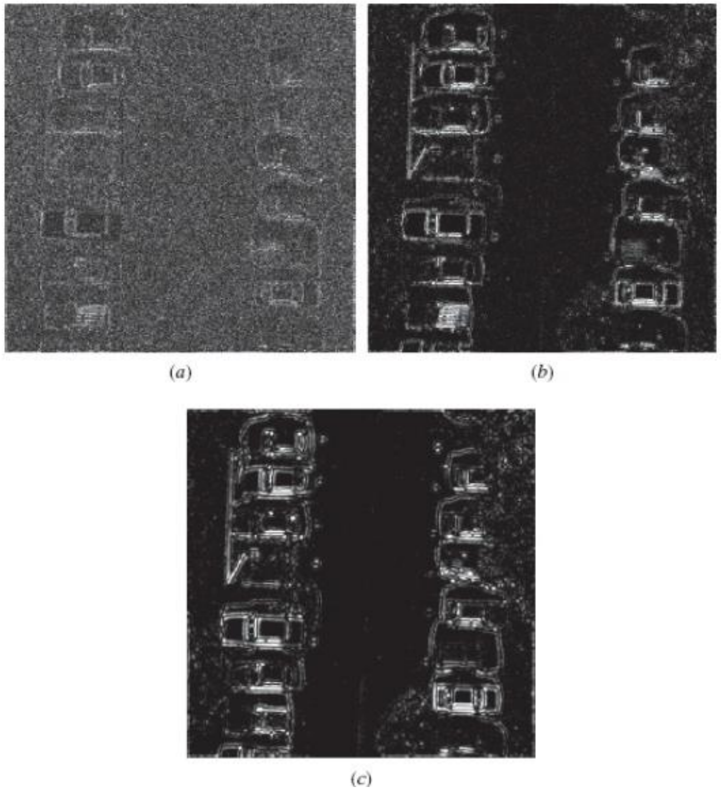


Figure 15-17. (a) Large-scale edge information from parking lot image. (b) Edge information at a smaller scale. (c) Edge information at a still smaller scale.

- The last image, Fig. 15-17c, is the result of three convolutions followed by a difference
- Starting with image I which has been convolved twice, an additional convolution is performed using a 7 × 7 kernel (simple average), resulting in image I
- This method of obtaining multiscale representations of edges is rather simplistic, but serves as an introduction to the concept
- Other approaches such as those that employ the wavelet transform can produce multiscale representations with better resolution

- The wavelet transform is often used to extract multiscale features (edges) for use in feature recognition and pattern matching

15-7. Multiscale Representation

- Another concept relating to image scale is that of the *image pyramid*
- An image pyramid is formed by successively convolving an image with a *gaussian* kernel, with each convolution producing a half-resolution copy of the previous image
- A gaussian kernel is one which has weights that correspond to the shape of a normal (gaussian) distribution or bell-shaped curve
- The series of images thus produced can be visualized as a stack of image layers forming a pyramid, as shown in Fig. 15-18
- This figure shows an image pyramid formed from the parking lot image, with the original image at the bottom of the pyramid and successive half-resolution copies going up the pyramid
- An image is formed by convolving the original image with a gaussian kernel, such as that shown in Eq. (15-10)

$$K=\begin{bmatrix} 0.0025 & 0.0125 & 0.0200 & 0.0125 & 0.0025 \\ 0.0125 & 0.0625 & 0.1000 & 0.0625 & 0.0125 \\ 0.0200 & 0.1000 & 0.1600 & 0.1000 & 0.0200 \\ 0.0125 & 0.0625 & 0.1000 & 0.0625 & 0.0125 \\ 0.0025 & 0.0125 & 0.0200 & 0.0125 & 0.0025 \end{bmatrix} \qquad (15\text{–}10)$$

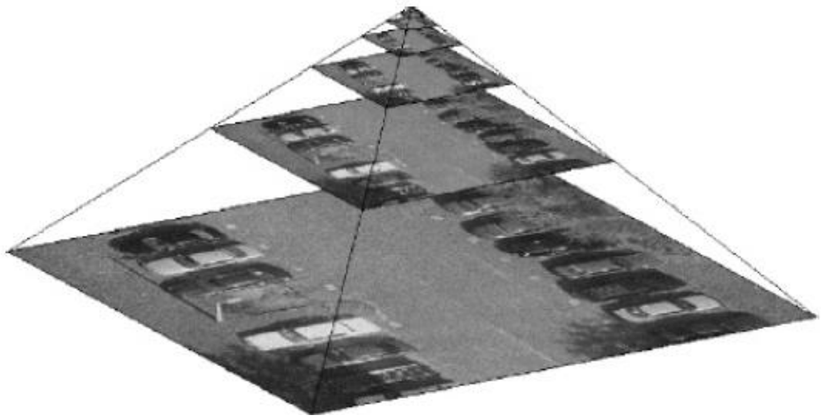


Figure 15-18. Image pyramid formed from parking lot image.

15-7. Multiscale Representation

- Notice that the weights in this kernel are largest in the center and fall off gradually away from the center
- This pattern of weights mimics the behavior of a gaussian distribution in two dimensions
- The convolution is performed in the usual fashion, with the exception that instead of the moving window shifting one row or column at a time, it is shifted two rows or columns
- This results in a convolved image having one-half as many rows and columns as the original
- This reduction in number of pixels is offset by the fact that each pixel in the convolved image represents an area twice the width and height of the original pixels
- Each successive convolution is performed on the previously convolved image, resulting in the series of half-resolution copies

15-7. Multiscale Representation

- Image pyramids can be used for many purposes

- One particularly important use is for multiresolution image matching
- By matching images in upper layers of the pyramid, the location of the match can be predicted in lower layers within a couple of pixels, which avoids searching through the entire full-resolution image to find a matching feature

- Another use is for quick display of an image while zooming in or out
- By first constructing an image pyramid, an image zoom operation can be accomplished by accessing different layers of the pyramid

- These are just two possible uses for image pyramids, and there are many others

15-8. Digital Image Matching

❖ A common task associated with the use of stereoplotters is to place the floating mark at a three-dimensional model position of an object point

❖ This requires the ability to recognize similar image characteristics (texture, shapes, etc.) in small regions from the left and right images of a stereopair

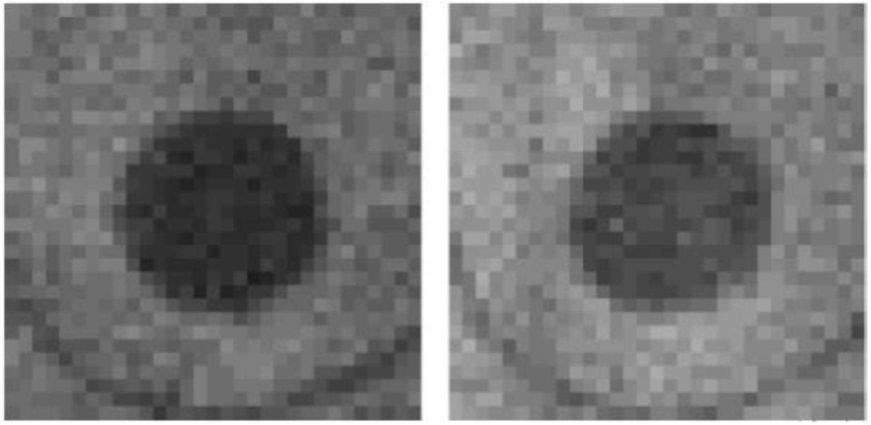


Figure 15-19. A manhole in two different aerial photographs.

❖ Figure 15-19 shows the same object (in this case a manhole) as it appears in two different aerial photos

❖ The task is to find, using the patch of pixels shown in the left image, the associated patch of pixels in the right image using some measure of similarity between the two

❖ The human visual system is able to perform this task with little conscious effort

- When our eyes fixate on an object, the two images merge and the three-dimensional nature of the object is revealed

❖ When one is working with digital images in a softcopy stereoplotter, placing of the floating mark can be performed either manually or by computer processing

- Computer software which accomplishes this uses techniques of digital image matching

15-8. Digital Image Matching

❖ Digital image-matching techniques fall into three general categories: *area-based, feature-based*, and *hybrid methods*

❖ Area-based methods perform the image match by a numerical comparison of digital numbers in small subarrays from each image
- This approach is straightforward and commonly used in softcopy systems

❖ Feature-based methods are more complicated and involve extraction of features, which are comprised of edges at different scales, with subsequent comparison based on feature characteristics such as size and shape
- Feature-based image matching requires techniques from the realm of artificial intelligence in computer science

❖ Hybrid methods involve some combination of the first two approaches
- Typically, hybrid methods involve preprocessing of the left and right images to highlight features (edges)
- After the features have been located, they are matched by areabased methods
- While all three approaches have particular advantages and disadvantages, this section focuses on area-based image-matching techniques

15-8. Digital Image Matching

- The simplest area-based digital image-matching method is a technique known as *normalized cross-correlation*
- In this approach, a statistical comparison is computed from digital numbers taken from same-size subarrays in the left and right images
- A correlation coefficient is computed by the following equation, using digital numbers from subarrays A and B

$$C = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[(A_{ij}-\bar{A})(B_{ij}-\bar{B})]}{\sqrt{\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij}-\bar{A})^2\right]\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(B_{ij}-\bar{B})^2\right]}} \qquad (15-11)$$

15-8. Digital Image Matching

- In Eq. (15-11), c is the correlation coefficient

$$C=\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[(A_{ij}-\bar{A})(B_{ij}-\bar{B})]}{\sqrt{\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij}-\bar{A})^2\right]\left[\sum_{i=1}^{m}\sum_{j=1}^{n}(B_{ij}-\bar{B})^2\right]}} \quad (15\text{-}11)$$

  - m and n are the numbers of rows and columns in the subarrays
  - A_{ij} is the digital number from subarray A at row i, column j
  - $\bar{A}$ is the average of all digital numbers in subarray A
  - B_{ij} is the digital number from subarray B at row i, column j
  - $\bar{B}$ is the average of all digital numbers in subarray B

- The correlation coefficient can range from -1 to +1, with +1 indicating perfect correlation (an exact match)

- A coefficient of -1 indicates negative correlation, which would occur if identical images from a photographic negative and positive were being compared

- Coefficient values near zero indicate a nonmatch, and could result from a comparison of any two sets of random numbers

- Due to factors such as image noise, perfect (+1) correlation is extremely rare

- Generally a threshold value, such as 0.7, is chosen and if the correlation coefficient exceeds that value, the subarrays are assumed to match

15-8. Digital Image Matching

- Normalized cross-correlation is essentially the same operation as *linear regression* in statistics
- Details of linear regression can be found in most elementary statistics texts, and only general concepts are discussed here
- In linear regression, a set of ordered pairs (abscissas and ordinates) is statistically analyzed to determine how well the numbers correspond to a straight-line relationship
- In the process, most-probable values are determined for the parameters (slope and intercept) of a best-fit line through the data points
- For example, assume the following pair of 3 × 3 arrays of digital numbers are to be analyzed using linear regression.

$$A = \begin{bmatrix} 25 & 48 & 89 \\ 43 & 94 & 47 \\ 76 & 21 & 57 \end{bmatrix} \qquad B = \begin{bmatrix} 33 & 56 & 81 \\ 40 & 98 & 54 \\ 84 & 16 & 49 \end{bmatrix}$$

15-8. Digital Image Matching

Table 15-1. Tabular Form for Computing Linear Regression

$a(x_i)$	$b(y_i)$	$a^2(x_i^2)$	$b^2(y_i^2)$	$a \times b\ (x_i\ y_i)$
25	33	625	1089	825
48	56	2304	3136	2688
89	81	7921	6561	7209
43	40	1849	1600	1720
94	98	8836	9604	9212
47	54	2209	2916	2538
76	84	5776	7056	6384
21	16	441	256	336
57	49	3249	2401	2793
∑ = 500	∑ = 511	∑ = 33,210	∑ = 34,619	∑ = 33,705

❖ To compute the linear regression, a tabular solution can be used, as shown in Table 15-1

❖ In this table, the abscissas and ordinates used for the regression are listed in the columns labeled a and b
- a^2 and b^2 are their corresponding squares
- $a \times b$ are the products

❖ Note: In typical notation used for linear regression, x_i and y_i are used for abscissas and ordinates, respectively, and this notation is also shown in the table

15-8. Digital Image Matching

❖ In order to compute the regression, the following terms are computed

$$S_x^2 = \sum (x_i - \bar{x})^2 = \sum x_i^2 - \frac{\left(\sum x_i\right)^2}{n} = 33{,}210 - \frac{500^2}{9} = 5432.2 \qquad (15\text{-}12)$$

$$S_y^2 = \sum (y_i - \bar{y})^2 = \sum y_i^2 - \frac{\left(\sum y_i\right)^2}{n} = 34{,}619 - \frac{511^2}{9} = 5605.6 \qquad (15\text{-}13)$$

$$S_{xy} = \sum [(x_i - \bar{x})(y_1 - \bar{y})] = \sum (x_i y_i) - \frac{\left(\sum x_i\right)\left(\sum y_i\right)}{n} = 33{,}705 - \frac{500 \times 511}{9} = 5316.1 \qquad (15\text{-}14)$$

$$\beta = \frac{S_{xy}}{S_x^2} = \frac{5316.1}{5432.2} = 0.979 \qquad (15\text{-}15)$$

$$\alpha = \frac{\sum y_i}{n} - \beta \frac{\sum x_i}{n} = \frac{511}{9} - 0.979\left(\frac{500}{9}\right) = 2.41 \qquad (15\text{-}16)$$

$$r = \frac{S_{xy}}{\sqrt{S_x^2 S_y^2}} = \frac{33{,}705}{\sqrt{33{,}210 \times 34{,}619}} = 0.963 \qquad (15\text{-}17)$$

- In Eqs. (15-12) through (15-17), n is the number of data points (9 in this case), and the other terms are as indicated in the table
- Parameter β in Eq. (15-15) is the slope of the regression line; parameter α in Eq. (15-16) is the y intercept of the regression line; and parameter r in Eq. (15-17) is the sample correlation coefficient
- Note that r is the same as the normalized cross-correlation coefficient c from Eq. (15-11)

15-8. Digital Image Matching

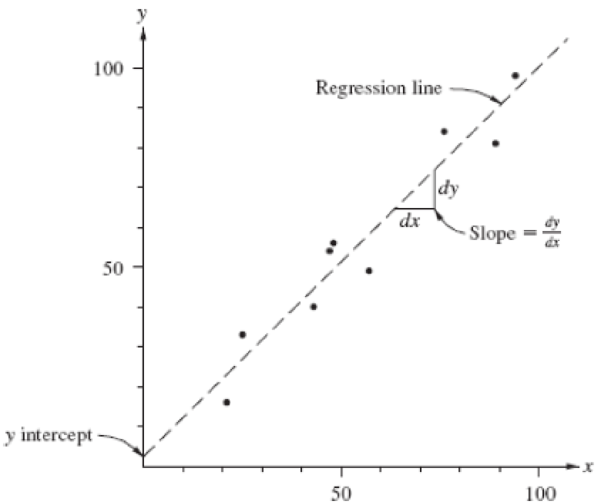


Figure 15−20. Linear regression.

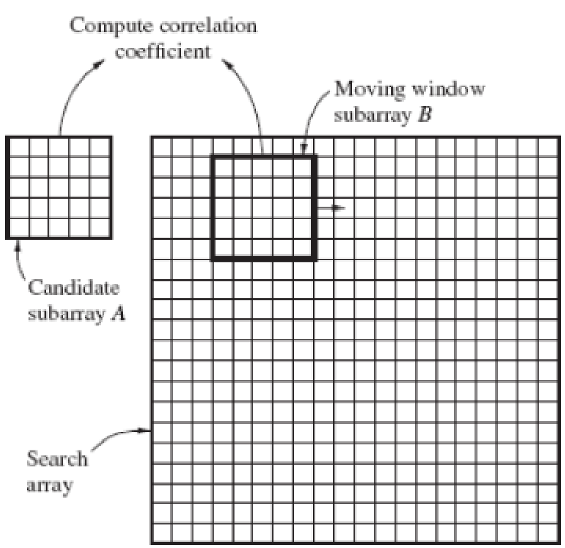


Figure 15−21. Computing correlation coefficients using a moving window within a search array.

- ❖ Figure 15-20 shows a plot of the nine data points, along with the regression line
- ❖ In this figure, the nine data points lie nearly along the regression line, which is also indicated by the correlation coefficient r being nearly equal to 1
- ❖ Digital image matching by correlation can be performed in the following manner
- ❖ A candidate subarray from the left photo is chosen, and a search will be performed for its corresponding subarray in the right image
- ❖ Since the exact position of the image in the right image is not initially known, a search array is selected with dimensions much larger than those of the candidate subarray

15-8. Digital Image Matching

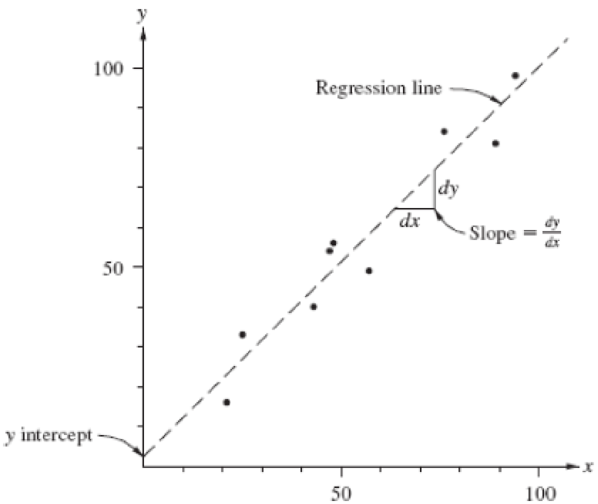


Figure 15−20. Linear regression.

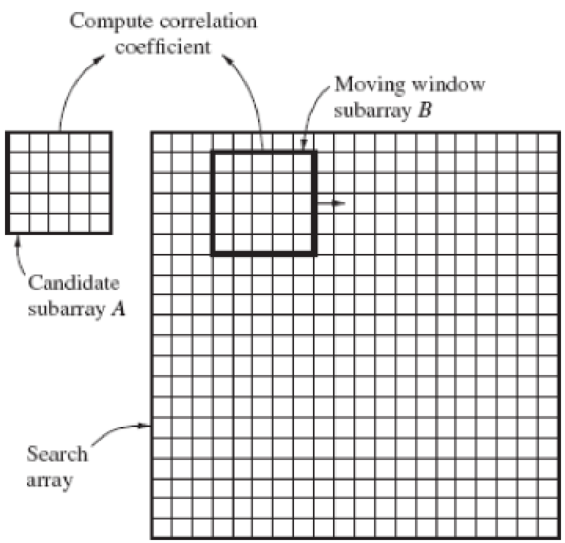


Figure 15−21. Computing correlation coefficients using a moving window within a search array.

- A moving window approach is then used, comparing the candidate subarray from the left image with all possible window locations within the search array from the right image, as illustrated in Fig. 15-21
- At each window location in the search array, the correlation coefficient is computed in a manner similar to moving window convolution, resulting in a correlation matrix C
- After all coefficients have been calculated, the largest correlation value in C is tested to see if it is above the threshold
- If it exceeds the threshold, the corresponding location within the search array is considered to be the match

15-8. Digital Image Matching

- A second area-based digital image-matching method is the least squares *matching* technique
- Conceptually, least squares matching is closely related to the correlation method, with the added advantage of being able to obtain the match location to a fraction of a pixel
- Least squares matching can also account for some distortions caused by perspective differences and rotation between images
- Figure 15-22 shows:
  - (a) a calibration target, (b) a tilted image of a target
  - (b) the resampled result from image(b), which is a least squares match of the template(a)
- Different implementations of least squares matching have been devised, with the following form being commonly used

$$A(x, y) = h_0 + h_1 B(x', y') \quad (15\text{-}18)$$

$$x' = a_0 + a_1 x + a_2 y \quad (15\text{-}19)$$

$$y' = b_0 + b_1 x + b_2 y \quad (15\text{-}20)$$

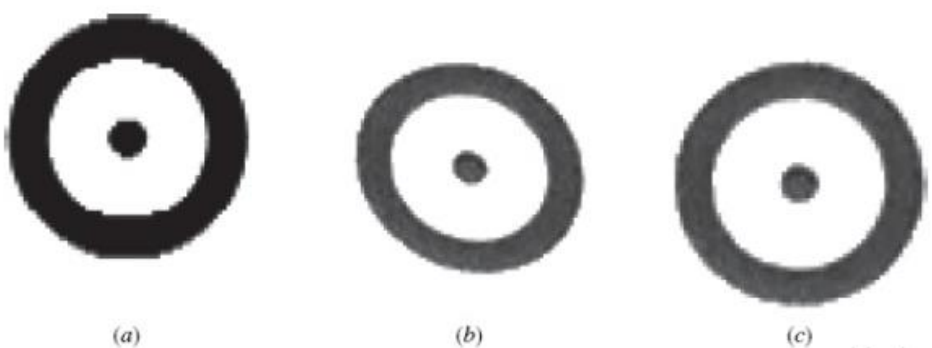


Figure 15-22. A calibration target template, the image of the target, and its resampled image.

15-8. Digital Image Matching

$$A(x, y) = h_0 + h_1 B(x', y') \quad (15-18)$$

❖ In Eq. (15-18), $A(x, y)$ is the digital number from the candidate subarray of the left image at location x, y; $B(x', y')$ is the digital number from a subarray in the search area of the right image at location x', y' ; h_0 is the radiometric shift; and h_1 is the radiometric scale

❖ Note that parameter h_0 is the same as the y intercept α of Eq. (15-16), and h_1 is the same as the slope β of Eq. (15-15)

$$\beta = \frac{S_{xy}}{S_x^2} = \frac{5316.1}{5432.2} = 0.979 \quad (15-15)$$

$$\alpha = \frac{\sum y_i}{n} - \beta \frac{\sum x_i}{n} = \frac{511}{9} - 0.979\left(\frac{500}{9}\right) = 2.41 \quad (15-16)$$

❖ Equations (15-19) and (15-20) specify an affine relationship between the coordinates of the pixel on the left photo and the coordinates of the corresponding pixel on the right photo

$$x' = a_0 + a_1 x + a_2 y \quad (15-19)$$

$$y' = b_0 + b_1 x + b_2 y \quad (15-20)$$

15-8. Digital Image Matching

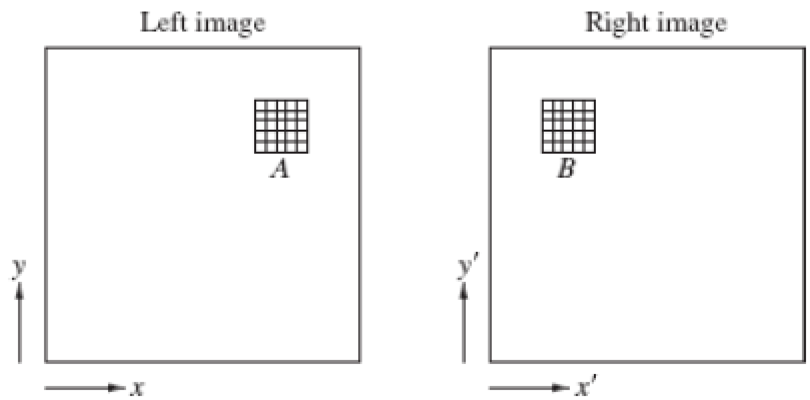


Figure 15-23. Positions of subarrays for least squares matching.

- Figure 15-23 illustrates the positions of subarrays A and B in the left and right images
- In this figure, the x and y axes are the basis for coordinates on the left image, and the x' and y' axes are the basis for coordinates on the right image
- Coordinates in both images are expressed in units of pixels

15-8. Digital Image Matching

$$A(x, y) = h_0 + h_1 B(x', y') \quad (15-18)$$

$$x' = a_0 + a_1 x + a_2 y \quad (15-19)$$

$$y' = b_0 + b_1 x + b_2 y \quad (15-20)$$

❖ Combining Eqs. (15-18), (15-19), and (15-20) and expressing the result in the form of **a least squares observation equation** gives the following:

$$f = h_0 + h_1 B(a_0 + a_1 x + a_2 y,\ b_0 + b_1 x + b_2 y) = A(x, y) + V_A \quad (15-21)$$

❖ In Eq. (15-21), f is the function relating the digital numbers from the two images, V_A is the residual, and the other variables are as previously defined

❖ Equation (15-21) is nonlinear, and application of Taylor's series gives the following linear form

$$f = f_0 + f'_{h_0} dh_0 + f'_{h_1} dh_1 + f'_{a_0} da_0 + f'_{a_1} da_1 + f'_{a_2} da_2 + f'_{b_0} db_0 + f'_{b_1} db_1 + f'_{b_2} db_2 = A(x, y) + V_A \quad (15-22)$$

where

$$f'_{h_0} = 1 \qquad f'_{a_2} = y h_1 B'_x$$

$$f'_{h_1} = B(x' = , y' =) \qquad f'_{b_0} = h_1 B'_y$$

$$f'_{a_0} = h_1 B'_x \qquad f'_{b_1} = x h_1 B'_y$$

$$f'_{a_1} = x h_1 B'_x \qquad f'_{b_2} = y h_1 B'_y$$

and

$$B'_x = \frac{B(x'+1, y') - B(x'-1, y')}{2} \quad (15-23)$$

$$B'_y = \frac{B(x', y'+1) - B(x'-, y'-1)}{2} \quad (15-24)$$

15-8. Digital Image Matching

$$f = h_0 + h_1 B(a_0 + a_1 x + a_2 y,\ b_0 + b_1 x + b_2 y) = A(x, y) + V_A \qquad (15\text{−}21)$$

$$B'_x = \frac{B(x'+1, y') - B(x'-1, y')}{2} \qquad (15\text{−}23)$$

$$B'_y = \frac{B(x', y'+1) - B(x'-, y'-1)}{2} \qquad (15\text{−}24)$$

- Since the function f of Eq. (15-21) includes digital numbers from subarray B, partial derivative terms must be obtained using discrete values to estimate the slope of B in both the x and y directions
- Equation (15-23) computes the estimate for slope in the x direction by taking the difference between the digital numbers of pixels to the right and left divided by 2, and Eq. (15-24) computes the estimate for slope in the y direction in a corresponding manner
- Use of these discrete slope estimates, in conjunction with the chain rule from calculus, allows the partial derivatives (the f' terms) to be determined, as listed above

15-8. Digital Image Matching

- Least squares matching is an iterative process which requires an accurate estimate for the position of B within the right image
- Initial approximations must be obtained for the unknown parameters h_0, h_1, a_0, a_1, a_2, b_0, b_1, and b_2
- Estimates for h_0 and h_1 can be obtained by linear regression as illustrated earlier in this section
- If the coordinates of the lower left pixels of A and B are x_0, y_0 and x'_0, y'_0 , respectively, the following initial approximations can be used for the affine parameters

$$a_o = x'_0 - x_0 \qquad a_1 = 1 \qquad a_2 = 0$$

$$b_o = y'_0 - y_0 \qquad b_1 = 0 \qquad b_2 = 1$$

15-8. Digital Image Matching

- Each iteration of the solution involves forming the linearized equations, solving the equations by least squares to obtain corrections to the approximations, and adding the corrections to the approximations

- At the beginning of an iteration, the pixels of subarray B (along with a 1-pixel-wide border around B which is needed for derivative estimates) are resampled (see App. E) from the right image

- This is done by stepping through the pixels of subarray A, taking the x and y coordinates of each pixel, and transforming them to the right image x' and y' by using Eqs. (15-19) and (15-20)

$$x' = a_0 + a_1 x + a_2 y \qquad (15\text{–}19)$$

$$y' = b_0 + b_1 x + b_2 y \qquad (15\text{–}20)$$

15-8. Digital Image Matching

- A corresponding digital number is then resampled from the right image at position x', y'
- Once subarray B has been filled, the least squares equations can be formed and solved
- The solution is then iterated until the corrections become negligible

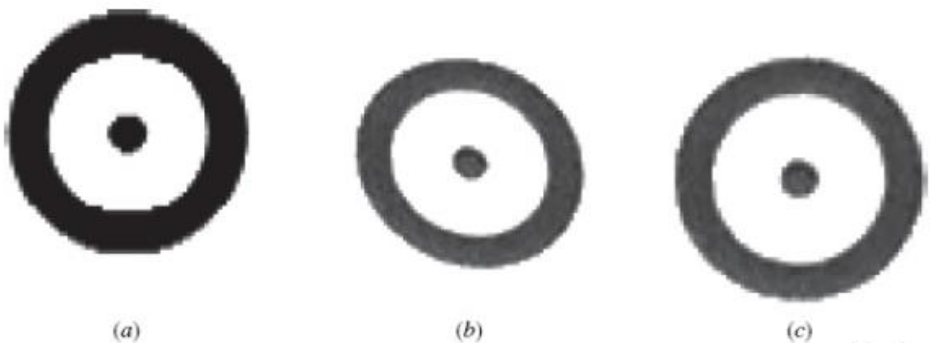


Figure 15-22. A calibration target template, the image of the target, and its resampled image.

- On the final iteration, the resampled subarray should be very similar to the template array as Fig. 15-22c is to Fig. 15-22a

15-8. Digital Image Matching

- ❖ Some final comments are appropriate at this point
- ❖ First, the estimated position of subarray B should be within a couple of pixels in order for the solution to converge properly
  - ➩ This can be achieved efficiently through the use of an image pyramid
- ❖ Corresponding points can be matched at an upper level of the pyramid where the search area contains fewer pixels
- ❖ Once the point is matched at a particular level of the pyramid, the position on the next-lower level will be known to within 2 pixels
- ❖ By progressively matching from upper levels down to the bottom level, accurate position estimates can be obtained at each subsequent level

15-8. Digital Image Matching

- Another concern is the size of the subarrays to be matched
- Generally, a subarray size of 20 × 20 to 30 × 30 gives satisfactory results
- If the subarray is much smaller, the low redundancy can result in a weak solution
- Larger subarrays can lead to problems due to terrain variations within the image area causing distortions that are not affine
- Finally, the transformation equation for y' can be simplified by performing *epipolar resampling* on the images prior to matching
- In epipolar resampling, the images are resampled so that the rows line up with epipolar lines
- When this is done, Eqs. (15-19) and (15-20) can be simplified to

$$x' = a_0 + a_1 x + a_2 y \quad (15\text{–}19)$$
$$y' = b_0 + b_1 x + b_2 y \quad (15\text{–}20)$$

$$\Rightarrow$$

$$x' = a_0 + a_1 x \quad (15\text{–}25)$$
$$y' = b_0 \quad (15\text{–}26)$$

15-9. Summary

- The discussion of digital image processing methods presented in this chapter is only a brief treatment of the subject
- These methods dealt primarily with edge detection, contrast enhancement, noise removal, multiscale representations, and image matching
- A wealth of other digital image processing methods are available for dealing with these problems as well as many others
- An in-depth understanding of digital image processing requires a great deal of study and experimentation
- Those who are interested in learning more about this subject are directed to the references listed below
- Further information can be gathered by consulting the bibliographies contained in these documents