Chapter 1

Introduction

Why do we process images?

Image Processing has been developed in response to three major problems concerned with pictures:

- picture digitisation and coding to facilitate transmission, printing and storage of pictures;
- picture enhancement and restoration in order, for example, to interpret more easily pictures of the surface of other planets taken by various probes;
- picture segmentation and description as an early stage to Machine Vision.

Image Processing nowadays refers mainly to the processing of digital images.

What is an image?

A **panchromatic image** is a 2D light intensity function, $f(x, y)$, where x and y are spatial coordinates and the value of f at (x, y) is proportional to the brightness of the scene at that point. If we have a **multispectral image**, $f(x, y)$ is a vector, each component of which indicates the brightness of the scene at point (x, y) at the corresponding spectral **band**.

What is a digital image?

A **digital image** is an image $f(x, y)$ that has been discretised both in spatial coordinates and in brightness. It is represented by a 2D integer array, or a series of 2D arrays, one for each colour band. The digitised brightness value is called **grey level**.

Each element of the array is called **pixel** or **pel**, derived from the term "picture element". Usually, the size of such an array is a few hundred pixels by a few hundred pixels and there are several dozens of possible different grey levels. Thus, a digital image looks like this

$$f(x, y) = \begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \vdots & \vdots & & \vdots \\ f(N,1) & f(N,2) & \dots & f(N,N) \end{bmatrix} \tag{1.1}$$

Image Processing: The Fundamentals, Second Edition Maria Petrou and Costas Petrou

with $0 \leq f(x, y) \leq G-1$, where usually N and G are expressed as positive integer powers of 2 ($N = 2^n$, $G = 2^m$).

What is a spectral band?

A colour band is a range of wavelengths of the electromagnetic spectrum, over which the sensors we use to capture an image have nonzero sensitivity. Typical colour images consist of three colour bands. This means that they have been captured by three different sets of sensors, each set made to have a different sensitivity function. Figure 1.1 shows typical sensitivity curves of a multispectral camera.

All the methods presented in this book, apart from those in Chapter 7, will refer to single band images.

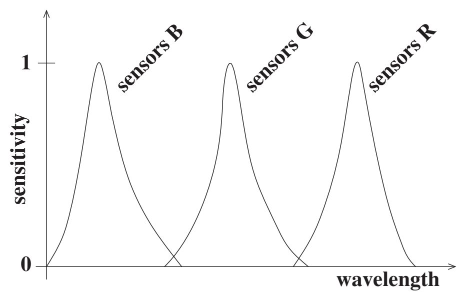


Figure 1.1: The spectrum of the light which reaches a sensor is multiplied with the sensitivity function of the sensor and recorded by the sensor. This recorded value is the brightness of the image in the location of the sensor and in the band of the sensor. This figure shows the sensitivity curves of three different sensor types.

Why do most image processing algorithms refer to grey images, while most images we come across are colour images?

For various reasons.

1. A lot of the processes we apply to a grey image can be easily extended to a colour image by applying them to each band separately.

2. A lot of the information conveyed by an image is expressed in its grey form and so colour is not necessary for its extraction. That is the reason black and white television receivers had been perfectly acceptable to the public for many years and black and white photography is still popular with many photographers.

3. For many years colour digital cameras were expensive and not widely available. A lot of image processing techniques were developed around the type of image that was available. These techniques have been well established in image processing.

Nevertheless, colour is an important property of the natural world, and so we shall examine its role in image processing in a separate chapter in this book.

How is a digital image formed?

Each pixel of an image corresponds to a part of a physical object in the 3D world. This physical object is illuminated by some light which is partly reflected and partly absorbed by it. Part of the reflected light reaches the array of sensors used to image the scene and is responsible for the values recorded by these sensors. One of these sensors makes up a pixel and its field of view corresponds to a small patch in the imaged scene. The recorded value by each sensor depends on its sensitivity curve. When a photon of a certain wavelength reaches the sensor, its energy is multiplied with the value of the sensitivity curve of the sensor at that wavelength and is accumulated. The total energy collected by the sensor (during the exposure time) is eventually used to compute the grey value of the pixel that corresponds to this sensor.

If a sensor corresponds to a patch in the physical world, how come we can have more than one sensor type corresponding to the same patch of the scene?

Indeed, it is not possible to have three different sensors with three different sensitivity curves corresponding exactly to the same patch of the physical world. That is why digital cameras have the three different types of sensor slightly displaced from each other, as shown in figure 1.2, with the sensors that are sensitive to the green wavelengths being twice as many as those sensitive to the blue and the red wavelengths. The recordings of the three sensor types are interpolated and superimposed to create the colour image. Recently, however, cameras have been constructed where the three types of sensor are combined so they exactly exist on top of each other and so they view exactly the same patch in the real world. These cameras produce much sharper colour images than the ordinary cameras.

R	G	R	G	R	G
G	B	G	B	G	B
R	G	R	G	R	G
G	B	G	B	G	B
R	G	R	G	R	G
G	B	G	B	G	B

Figure 1.2: The *RGB* sensors as they are arranged in a typical digital camera.

What is the physical meaning of the brightness of an image at a pixel position?

The brightness values of different pixels have been created by using the energies recorded by the corresponding sensors. They have significance only *relative* to each other and they are meaningless in absolute terms. So, pixel values between different images should only be compared if either care has been taken for the physical processes used to form the two images to be identical, or the brightness values of the two images have somehow been normalised so that the effects of the different physical processes have been removed. In that case, we say that the sensors are **calibrated**.

Example 1.1

You are given a triple array of 3×3 sensors, arranged as shown in figure 1.3, with their sampled sensitivity curves given in table 1.1. The last column of this table gives in some arbitrary units the energy, E_{λ_i}, carried by photons with wavelength λ_i.

Wavelength	Sensors B	Sensors G	Sensors R	Energy
λ_0	0.2	0.0	0.0	1.00
λ_1	0.4	0.2	0.1	0.95
λ_2	0.8	0.3	0.2	0.90
λ_3	1.0	0.4	0.2	0.88
λ_4	0.7	0.6	0.3	0.85
λ_5	0.2	1.0	0.5	0.81
λ_6	0.1	0.8	0.6	0.78
λ_7	0.0	0.6	0.8	0.70
λ_8	0.0	0.3	1.0	0.60
λ_9	0.0	0.0	0.6	0.50

Table 1.1: The sensitivity curves of three types of sensor for wavelengths in the range $[\lambda_0, \lambda_9]$, and the corresponding energy of photons of these particular wavelengths in some arbitrary units.

The shutter of the camera is open long enough for 10 photons to reach the locations of the sensors.

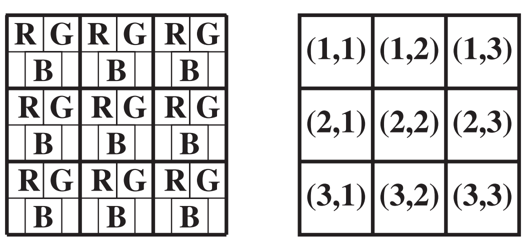


Figure 1.3: Three 3×3 sensor arrays interlaced. On the right, the locations of the pixels they make up. Although the three types of sensor are slightly misplaced with respect to each other, we assume that each triplet is coincident and forms a single pixel.

For simplicity, we consider that exactly the same types of photon reach all sensors that correspond to the same location.

The wavelengths of the photons that reach the pixel locations of each triple sensor, as identified in figure 1.3, are:

Location $(1,1)$: $\lambda_0,\lambda_9,\lambda_9,\lambda_8,\lambda_7,\lambda_8,\lambda_1,\lambda_0,\lambda_1,\lambda_1$
Location $(1,2)$: $\lambda_1,\lambda_3,\lambda_3,\lambda_4,\lambda_4,\lambda_5,\lambda_2,\lambda_6,\lambda_4,\lambda_5$
Location $(1,3)$: $\lambda_6,\lambda_7,\lambda_7,\lambda_0,\lambda_5,\lambda_6,\lambda_6,\lambda_1,\lambda_5,\lambda_9$
Location $(2,1)$: $\lambda_0,\lambda_1,\lambda_0,\lambda_2,\lambda_1,\lambda_1,\lambda_4,\lambda_3,\lambda_3,\lambda_1$
Location $(2,2)$: $\lambda_3,\lambda_3,\lambda_4,\lambda_3,\lambda_4,\lambda_4,\lambda_5,\lambda_2,\lambda_9,\lambda_4$
Location $(2,3)$: $\lambda_7,\lambda_7,\lambda_6,\lambda_7,\lambda_6,\lambda_1,\lambda_5,\lambda_9,\lambda_8,\lambda_7$
Location $(3,1)$: $\lambda_6,\lambda_6,\lambda_1,\lambda_8,\lambda_7,\lambda_8,\lambda_9,\lambda_9,\lambda_8,\lambda_7$
Location $(3,2)$: $\lambda_0,\lambda_4,\lambda_3,\lambda_4,\lambda_1,\lambda_5,\lambda_4,\lambda_0,\lambda_2,\lambda_1$
Location $(3,3)$: $\lambda_3,\lambda_4,\lambda_1,\lambda_0,\lambda_0,\lambda_4,\lambda_2,\lambda_5,\lambda_2,\lambda_4$

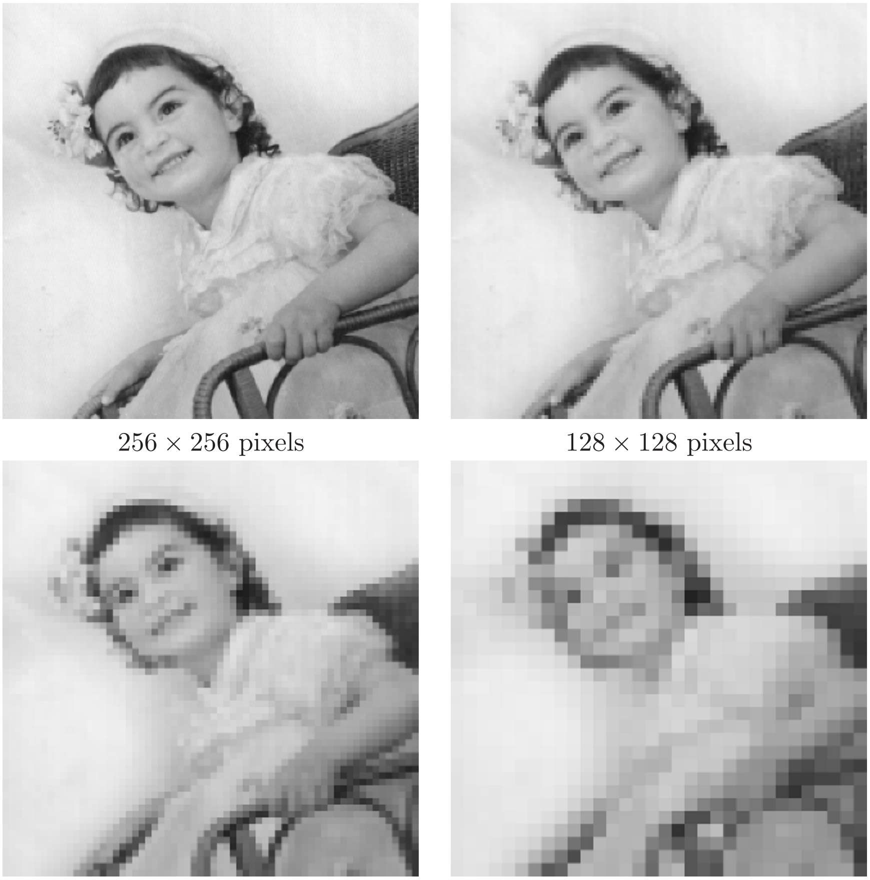
Calculate the values that each sensor array will record and thus produce the three *photon energy* bands recorded.

We denote by $g_X(i,j)$ the value that will be recorded by sensor type X at location (i,j). For sensors R, G and B in location $(1,1)$, the recorded values will be:

$$\begin{aligned}
g_R(1,1) &= 2E_{\lambda_0}\times 0.0 + 2E_{\lambda_9}\times 0.6 + 2E_{\lambda_8}\times 1.0 + 1E_{\lambda_7}\times 0.8 + 3E_{\lambda_1}\times 0.1 \\
&= 1.0\times 0.6 + 1.2\times 1.0 + 0.7\times 0.8 + 2.85\times 0.1 \\
&= 2.645 \qquad (1.2) \\
g_G(1,1) &= 2E_{\lambda_0}\times 0.0 + 2E_{\lambda_9}\times 0.0 + 2E_{\lambda_8}\times 0.3 + 1E_{\lambda_7}\times 0.6 + 3E_{\lambda_1}\times 0.2 \\
&= 1.2\times 0.3 + 0.7\times 0.6 + 2.85\times 0.2 \\
&= 1.35 \qquad (1.3) \\
g_B(1,1) &= 2E_{\lambda_0}\times 0.2 + 2E_{\lambda_9}\times 0.0 + 2E_{\lambda_8}\times 0.0 + 1E_{\lambda_7}\times 0.0 + 3E_{\lambda_1}\times 0.4 \\
&= 2.0\times 0.2 + 2.85\times 0.4 \\
&= 1.54 \qquad (1.4)
\end{aligned}$$

Working in a similar way, we deduce that the energies recorded by the three sensor arrays are:

$$\begin{aligned}
E_R &= \begin{pmatrix} 2.645 & 2.670 & 3.729 \\ 1.167 & 4.053 & 4.576 \\ 4.551 & 1.716 & 1.801 \end{pmatrix} \\
E_G &= \begin{pmatrix} 1.350 & 4.938 & 4.522 \\ 2.244 & 4.176 & 4.108 \\ 2.818 & 2.532 & 2.612 \end{pmatrix} \\
E_B &= \begin{pmatrix} 1.540 & 5.047 & 1.138 \\ 4.995 & 5.902 & 0.698 \\ 0.536 & 4.707 & 5.047 \end{pmatrix} \qquad (1.5)
\end{aligned}$$



256 × 256 pixels   128 × 128 pixels

64 × 64 pixels   32 × 32 pixels

Figure 1.4: Keeping the number of grey levels constant and decreasing the number of pixels with which we digitise the same field of view produces the checkerboard effect.

Why are images often quoted as being 512 × 512, 256 × 256, 128 × 128 etc?

Many calculations with images are simplified when the size of the image is a power of 2. We shall see some examples in Chapter 2.

How many bits do we need to store an image?

The number of bits, b, we need to store an image of size $N \times N$, with 2^m grey levels, is:

$$b = N \times N \times m \tag{1.6}$$

So, for a typical 512×512 image with 256 grey levels ($m = 8$) we need 2,097,152 bits or 262,144 8-bit bytes. That is why we often try to reduce m and N, without significant loss of image quality.

What determines the quality of an image?

The quality of an image is a complicated concept, largely subjective and very much application dependent. Basically, an image is of good quality if it is not noisy and
(1) it is not blurred;
(2) it has high resolution;
(3) it has good contrast.

What makes an image blurred?

Image blurring is caused by incorrect image capturing conditions. For example, out of focus camera, or relative motion of the camera and the imaged object. The amount of image blurring is expressed by the so called **point spread function** of the imaging system.

What is meant by image resolution?

The resolution of an image expresses how much detail we can see in it and clearly depends on the number of pixels we use to represent a scene (parameter N in equation (1.6)) and the number of grey levels used to quantise the brightness values (parameter m in equation (1.6)).

Keeping m constant and decreasing N results in the **checkerboard effect** (figure 1.4). Keeping N constant and reducing m results in **false contouring** (figure 1.5). Experiments have shown that the more detailed a picture is, the less it improves by keeping N constant and increasing m. So, for a detailed picture, like a picture of crowds (figure 1.6), the number of grey levels we use does not matter much.

Example 1.2

Assume that the range of values recorded by the sensors of example 1.1 is from 0 to 10. From the values of the three bands captured by the three sets of sensors, create digital bands with 3 bits each ($m = 3$).

For 3-bit images, the pixels take values in the range $[0, 2^3 - 1]$*, ie in the range* $[0, 7]$*. Therefore, we have to divide the expected range of values to* 8 *equal intervals:* $10/8 = 1.25$*. So, we use the following conversion table:*

All pixels with recorded value in the range $[0.0, 1.25)$ *get grey value* 0
All pixels with recorded value in the range $[1.25, 2.5)$ *get grey value* 1
All pixels with recorded value in the range $[2.5, 3.75)$ *get grey value* 2
All pixels with recorded value in the range $[3.75, 5.0)$ *get grey value* 3
All pixels with recorded value in the range $[5.0, 6.25)$ *get grey value* 4
All pixels with recorded value in the range $[6.25, 7.5)$ *get grey value* 5
All pixels with recorded value in the range $[7.5, 8.75)$ *get grey value* 6
All pixels with recorded value in the range $[8.75, 10.0]$ *get grey value* 7

This mapping leads to the following bands of the recorded image.

$$R = \begin{pmatrix} 2 & 2 & 2 \\ 0 & 3 & 3 \\ 3 & 1 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 1 & 3 & 3 \\ 1 & 3 & 3 \\ 2 & 2 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 4 & 0 \\ 3 & 4 & 0 \\ 0 & 3 & 4 \end{pmatrix} \tag{1.7}$$

$m = 8$ $m = 7$ $m = 6$

$m = 5$ $m = 4$ $m = 3$

$m = 2$ $m = 1$

Figure 1.5: Keeping the size of the image constant (249×199) and reducing the number of grey levels ($= 2^m$) produces false contouring. To display the images, we always map the different grey values to the range $[0, 255]$.

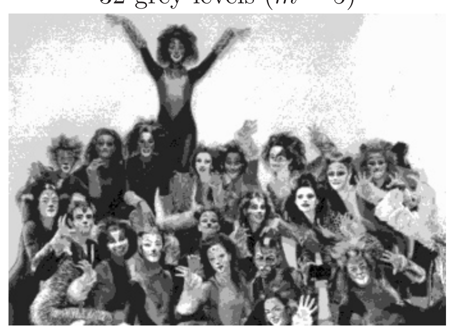
256 grey levels ($m = 8$)

128 grey levels ($m = 7$)

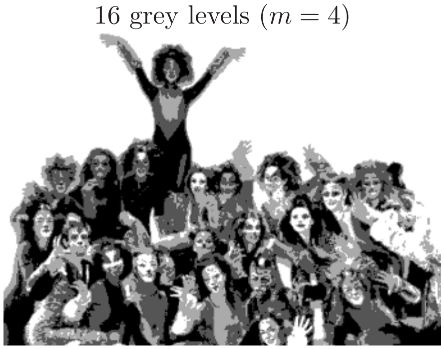
64 grey levels ($m = 6$)

32 grey levels ($m = 5$)

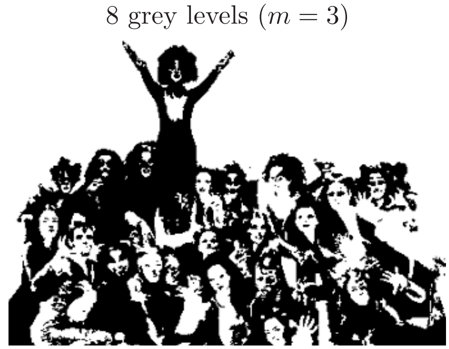
16 grey levels ($m = 4$)



8 grey levels ($m = 3$)



4 grey levels ($m = 2$)



2 grey levels ($m = 1$)

Figure 1.6: Keeping the number of pixels constant and reducing the number of grey levels does not affect much the appearance of an image that contains a lot of details.

What does "good contrast" mean?

Good contrast means that the grey values present in the image range from black to white, making use of the full range of brightness to which the human vision system is sensitive.

Example 1.3

Consider each band created in example 1.2 as a separate grey image. Do these images have good contrast? If not, propose some way by which bands with good contrast could be created from the recorded sensor values.

The images created in example 1.2 do not have good contrast because none of them contains the value 7 (which corresponds to the maximum brightness and which would be displayed as white by an image displaying device).
The reason for this is the way the quantisation was performed: the look up table created to convert the real recorded values to digital values took into consideration the full range of possible values a sensor may record (ie from 0 *to* 10*). To utilise the full range of grey values for each image, we should have considered the minimum and the maximum value of its pixels, and map that range to the* 8 *distinct grey levels. For example, for the image that corresponds to band R, the values are in the range* $[1.167, 4.576]$. *If we divide this range in* 8 *equal sub-ranges, we shall have:*

All pixels with recorded value in the range $[0.536, 1.20675)$	*get grey value* 0
All pixels with recorded value in the range $[1.20675, 1.8775)$	*get grey value* 1
All pixels with recorded value in the range $[1.8775, 2.54825)$	*get grey value* 2
All pixels with recorded value in the range $[2.54825, 3.219)$	*get grey value* 3
All pixels with recorded value in the range $[3.219, 3.88975)$	*get grey value* 4
All pixels with recorded value in the range $[3.88975, 4.5605)$	*get grey value* 5
All pixels with recorded value in the range $[4.5605, 5.23125)$	*get grey value* 6
All pixels with recorded value in the range $[5.23125, 5.902]$	*get grey value* 7

We must create one such look up table for each band. The grey images we create this way are:

$$R = \begin{pmatrix} 3 & 3 & 6 \\ 0 & 6 & 7 \\ 7 & 1 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 0 & 7 & 7 \\ 1 & 6 & 6 \\ 3 & 2 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 6 & 0 \\ 6 & 7 & 0 \\ 0 & 6 & 6 \end{pmatrix} \tag{1.8}$$

Example 1.4

Repeat example 1.3, now treating the three bands as parts of the same colour image.

If we treat all three bands as a single colour image (as they are meant to be), we must *find the minimum and maximum value over all three recorded bands, and create a look up table appropriate for all bands. In this case, the range of the recorded values is* $[0.536, 5.902]$*. The look up table we create by mapping this range to the range* $[0, 7]$ *is*

All pixels with recorded value in the range $[0.536, 1.20675)$	*get grey value* 0
All pixels with recorded value in the range $[1.20675, 1.8775)$	*get grey value* 1
All pixels with recorded value in the range $[1.8775, 2.54825)$	*get grey value* 2
All pixels with recorded value in the range $[2.54825, 3.219)$	*get grey value* 3
All pixels with recorded value in the range $[3.219, 3.88975)$	*get grey value* 4
All pixels with recorded value in the range $[3.88975, 4.5605)$	*get grey value* 5
All pixels with recorded value in the range $[4.5605, 5.23125)$	*get grey value* 6
All pixels with recorded value in the range $[5.23125, 5.902]$	*get grey value* 7

The three image bands we create this way are:

$$R = \begin{pmatrix} 3 & 3 & 4 \\ 1 & 5 & 6 \\ 5 & 1 & 1 \end{pmatrix} \quad G = \begin{pmatrix} 1 & 6 & 5 \\ 2 & 5 & 5 \\ 3 & 2 & 3 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 6 & 0 \\ 6 & 7 & 0 \\ 0 & 6 & 6 \end{pmatrix} \tag{1.9}$$

Note that each of these bands if displayed as a separate grey image may not display the full range of grey values, but it will have grey values that are consistent *with those of the other three bands, and so they can be directly compared. For example, by looking at the three digital bands we can say that pixel* $(1, 1)$ *has the same brightness (within the limits of the digitisation error) in bands G or B. Such a statement was not possible in example 1.3 because the three digital bands were not calibrated.*

What is the purpose of image processing?

Image processing has multiple purposes.

- To improve the quality of an image in a subjective way, usually by increasing its contrast. This is called **image enhancement**.
- To use as few bits as possible to represent the image, with minimum deterioration in its quality. This is called **image compression**.
- To improve an image in an objective way, for example by reducing its blurring. This is called **image restoration**.
- To make explicit certain characteristics of the image which can be used to identify the contents of the image. This is called **feature extraction**.

How do we do image processing?

We perform image processing by using image transformations. Image transformations are performed using **operators**. An operator takes as input an image and produces another image. In this book we shall put emphasis on a particular class of operators, called **linear operators**.

Do we use nonlinear operators in image processing?

Yes. We shall see several examples of them in this book. However, nonlinear operators cannot be collectively characterised. They are usually problem- and application-specific, and they are studied as individual processes used for specific tasks. On the contrary, linear operators can be studied collectively, because they share important common characteristics, irrespective of the task they are expected to perform.

What is a linear operator?

Consider $\mathcal{O}$ to be an operator which takes images into images. If f is an image, $\mathcal{O}(f)$ is the result of applying $\mathcal{O}$ to f. $\mathcal{O}$ is linear if

$$\mathcal{O}[af + bg] = a\mathcal{O}[f] + b\mathcal{O}[g] \tag{1.10}$$

for all images f and g and all scalars a and b.

How are linear operators defined?

Linear operators are defined in terms of their **point spread functions**. The point spread function of an operator is what we get out if we apply the operator on a point source:

$$\mathcal{O}[\text{point source}] \quad \equiv \quad \text{point spread function} \tag{1.11}$$

Or

$$\mathcal{O}[\delta(\alpha - x, \beta - y)] \quad \equiv \quad h(x, \alpha, y, \beta) \tag{1.12}$$

where $\delta(\alpha - x, \beta - y)$ is a point source of brightness 1 centred at point (x, y).

What is the relationship between the point spread function of an imaging device and that of a linear operator?

They both express the effect of either the imaging device or the operator on a point source. In the real world a star is the nearest to a point source. Assume that we capture the image of a star by using a camera. The star will appear in the image like a blob: the camera received the light of the point source and spread it into a blob. The bigger the blob, the more blurred the image of the star will look. So, the point spread function of the camera measures the amount of blurring present in the images captured by this camera. The camera, therefore, acts like a linear operator which accepts as input the ideal brightness function of the continuous real world and produces the recorded digital image. That is why we use the term "point spread function" to characterise both cameras and linear operators.

How does a linear operator transform an image?

If the operator is *linear*, when the point source is a times brighter, the result will be a times higher:

$$\mathcal{O}\left[a\delta(\alpha - x, \beta - y)\right] = ah(x, \alpha, y, \beta) \tag{1.13}$$

An image is a collection of point sources (the *pixels*) each with its own brightness value. For example, assuming that an image f is 3×3 in size, we may write:

$$f = \begin{pmatrix} f(1,1) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & f(1,2) & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \cdots + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & f(3,3) \end{pmatrix} \tag{1.14}$$

We may say that an image is the sum of these point sources. Then the effect of an operator characterised by point spread function $h(x, \alpha, y, \beta)$ on an image $f(x, y)$ can be written as:

$$g(\alpha, \beta) = \sum_{x=1}^{N} \sum_{y=1}^{N} f(x, y) h(x, \alpha, y, \beta) \tag{1.15}$$

where $g(\alpha, \beta)$ is the output "image", $f(x, y)$ is the input image and the size of the image is $N \times N$. Here we treat $f(x, y)$ as the brightness of a point source located at position (x, y). Applying an operator on it produces the point spread function of the operator times the strength of the source, ie times the grey value $f(x, y)$ at that location. Then, as the operator is linear, we sum over all such point sources, ie we sum over all pixels.

What is the meaning of the point spread function?

The point spread function $h(x, \alpha, y, \beta)$ expresses how much the input value at position (x, y) influences the output value at position (α, β). If the influence expressed by the point spread function is independent of the actual positions but depends only on the *relative* position of the influencing and the influenced pixels, we have a **shift invariant** point spread function:

$$h(x, \alpha, y, \beta) = h(\alpha - x, \beta - y) \tag{1.16}$$

Then equation (1.15) is a *convolution*:

$$g(\alpha, \beta) = \sum_{x=1}^{N} \sum_{y=1}^{N} f(x, y) h(\alpha - x, \beta - y) \tag{1.17}$$

If the columns are influenced independently from the rows of the image, then the point spread function is **separable**:

$$h(x, \alpha, y, \beta) \equiv h_c(x, \alpha) h_r(y, \beta) \tag{1.18}$$

The above expression serves also as the definition of functions $h_c(x, \alpha)$ and $h_r(y, \beta)$. Then equation (1.15) may be written as a cascade of two 1D transformations:

$$g(\alpha, \beta) = \sum_{x=1}^{N} h_c(x, \alpha) \sum_{y=1}^{N} f(x, y) h_r(y, \beta) \tag{1.19}$$

If the point spread function is both shift invariant and separable, then equation (1.15) may be written as a cascade of two 1D convolutions:

$$g(\alpha, \beta) = \sum_{x=1}^{N} h_c(\alpha - x) \sum_{y=1}^{N} f(x, y) h_r(\beta - y) \tag{1.20}$$

Box 1.1. The formal definition of a point source in the continuous domain

Let us define an extended source of constant brightness

$$\delta_n(x,y) \equiv n^2 rect(nx,ny) \tag{1.21}$$

where n is a positive constant and

$$rect(nx,ny) \equiv \begin{cases} 1 \text{ inside a rectangle } |nx| \leq \frac{1}{2}, |ny| \leq \frac{1}{2} \\ 0 \text{ elsewhere} \end{cases} \tag{1.22}$$

The total brightness of this source is given by

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \delta_n(x,y)dxdy = n^2 \underbrace{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} rect(nx,ny)dxdy}_{\text{area of rectangle}} = 1 \tag{1.23}$$

and is independent of n.

As $n \to +\infty$, we create a sequence, δ_n, of extended square sources which gradually shrink with their brightness remaining constant. At the limit, δ_n becomes Dirac's delta function

$$\delta(x,y) \begin{cases} \neq 0 & \text{for } x = y = 0 \\ = 0 & \text{elsewhere} \end{cases} \tag{1.24}$$

with the property:

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \delta(x,y)dxdy = 1 \tag{1.25}$$

Integral

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \delta_n(x,y)g(x,y)dxdy \tag{1.26}$$

is the average of image $g(x,y)$ over a square with sides $\frac{1}{n}$ centred at $(0,0)$. At the limit, we have

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \delta(x,y)g(x,y)dxdy = g(0,0) \tag{1.27}$$

which is the value of the image at the origin. Similarly,

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)\delta_n(x-a,y-b)dxdy \tag{1.28}$$

is the average value of g over a square $\frac{1}{n} \times \frac{1}{n}$ centred at $x = a$, $y = b$, since:

$$\begin{aligned} \delta_n(x-a, y-b) &= n^2 rect[n(x-a), n(y-b)] \\ &= \begin{cases} n^2 & |n(x-a)| \le \frac{1}{2} \quad |n(y-b)| \le \frac{1}{2} \\ 0 & \text{elsewhere} \end{cases} \end{aligned} \tag{1.29}$$

We can see that this is a square source centred at (a, b) by considering that $|n(x-a)| \le \frac{1}{2}$ means $-\frac{1}{2} \le n(x-a) \le \frac{1}{2}$, ie $-\frac{1}{2n} \le x - a \le \frac{1}{2n}$, or $a - \frac{1}{2n} \le x \le a + \frac{1}{2n}$. Thus, we have $\delta_n(x-a, y-b) = n^2$ in the region $a - \frac{1}{2n} \le x \le a + \frac{1}{2n}$, $b - \frac{1}{2n} \le y \le b + \frac{1}{2n}$.
At the limit of $n \to +\infty$, integral (1.28) is the value of the image g at $x = a$, $y = b$, ie:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g(x, y)\delta_n(x-a, y-b)dxdy = g(a, b) \tag{1.30}$$

This equation is called the **shifting property** of the delta function. This equation also shows that *any* image $g(a, b)$ can be expressed as a superposition of point sources.

Example 1.5

The following 3×3 image

$$f = \begin{pmatrix} 0 & 2 & 6 \\ 1 & 4 & 7 \\ 3 & 5 & 7 \end{pmatrix} \tag{1.31}$$

is processed by a linear operator $\mathcal{O}$ which has a point spread function $h(x, \alpha, y, \beta)$ defined as:

$h(1,1,1,1) = 1.0$	$h(1,1,1,2) = 0.5$	$h(1,1,1,3) = 0.0$	$h(1,2,1,1) = 0.5$
$h(1,2,1,2) = 0.0$	$h(1,2,1,3) = 0.4$	$h(1,3,1,1) = 0.5$	$h(1,3,1,2) = 1.0$
$h(1,3,1,3) = 0.6$	$h(2,1,1,1) = 0.8$	$h(2,1,1,2) = 0.7$	$h(2,1,1,3) = 0.4$
$h(2,2,1,1) = 0.6$	$h(2,2,1,2) = 0.5$	$h(2,2,1,3) = 0.4$	$h(2,3,1,1) = 0.4$
$h(2,3,1,2) = 0.8$	$h(2,3,1,3) = 1.0$	$h(3,1,1,1) = 0.9$	$h(3,1,1,2) = 0.5$
$h(3,1,1,3) = 0.5$	$h(3,2,1,1) = 0.6$	$h(3,2,1,2) = 0.5$	$h(3,2,1,3) = 0.3$
$h(3,3,1,1) = 0.5$	$h(3,3,1,2) = 0.9$	$h(3,3,1,3) = 1.0$	$h(1,1,2,1) = 1.0$
$h(1,1,2,2) = 0.6$	$h(1,1,2,3) = 0.2$	$h(1,2,2,1) = 0.0$	$h(1,2,2,2) = 0.2$
$h(1,2,2,3) = 0.4$	$h(1,3,2,1) = 0.4$	$h(1,3,2,2) = 1.0$	$h(1,3,2,3) = 0.6$
$h(2,1,2,1) = 0.8$	$h(2,1,2,2) = 0.7$	$h(2,1,2,3) = 0.6$	$h(2,2,2,1) = 0.6$
$h(2,2,2,2) = 0.5$	$h(2,2,2,3) = 0.5$	$h(2,3,2,1) = 0.5$	$h(2,3,2,2) = 1.0$
$h(2,3,2,3) = 1.0$	$h(3,1,2,1) = 0.7$	$h(3,1,2,2) = 0.5$	$h(3,1,2,3) = 0.5$
$h(3,2,2,1) = 0.6$	$h(3,2,2,2) = 0.5$	$h(3,2,2,3) = 0.5$	$h(3,3,2,1) = 0.5$
$h(3,3,2,2) = 1.0$	$h(3,3,2,3) = 1.0$	$h(1,1,3,1) = 1.0$	$h(1,1,3,2) = 0.6$
$h(1,1,3,3) = 1.0$	$h(1,2,3,1) = 0.5$	$h(1,2,3,2) = 0.1$	$h(1,2,3,3) = 0.6$
$h(1,3,3,1) = 0.5$	$h(1,3,3,2) = 1.0$	$h(1,3,3,3) = 0.6$	$h(2,1,3,1) = 0.5$
$h(2,1,3,2) = 0.7$	$h(2,1,3,3) = 0.5$	$h(2,2,3,1) = 0.6$	$h(2,2,3,2) = 0.5$
$h(2,2,3,3) = 0.5$	$h(2,3,3,1) = 0.4$	$h(2,3,3,2) = 0.9$	$h(2,3,3,3) = 1.0$

$h(3,1,3,1) = 0.8$ $h(3,1,3,2) = 0.5$ $h(3,1,3,3) = 0.5$ $h(3,2,3,1) = 0.5$
$h(3,2,3,2) = 0.5$ $h(3,2,3,3) = 0.8$ $h(3,3,3,1) = 0.4$ $h(3,3,3,2) = 0.4$
$h(3,3,3,3) = 1.0$

Work out the output image.

The point spread function of the operator $h(x,\alpha,y,\beta)$ *gives the weight with which the pixel value at input position* (x,y) *contributes to output pixel position* (α,β). *Let us call the output image* $g(\alpha,\beta)$. *We show next how to calculate* $g(1,1)$. *For* $g(1,1)$, *we need to use the values of* $h(x,1,y,1)$ *to weigh the values of pixels* (x,y) *of the input image:*

$$\begin{aligned} g(1,1) &= \sum_{x=1}^{3}\sum_{y=1}^{3} f(x,y)h(x,1,y,1) \\ &= f(1,1)h(1,1,1,1) + f(1,2)h(1,1,2,1) + f(1,3)h(1,1,3,1) \\ &\quad +f(2,1)h(2,1,1,1) + f(2,2)h(2,1,2,1) + f(2,3)h(2,1,3,1) \\ &\quad +f(3,1)h(3,1,1,1) + f(3,2)h(3,1,2,1) + f(3,3)h(3,1,3,1) \\ &= 2\times 1.0 + 6\times 1.0 + 1\times 0.8 + 4\times 0.8 + 7\times 0.5 + 3\times 0.9 \\ &\quad +5\times 0.7 + 7\times 0.8 = 27.3 \end{aligned} \tag{1.32}$$

For $g(1,2)$, *we need to use the values of* $h(x,1,y,2)$:

$$g(1,2) = \sum_{x=1}^{3}\sum_{y=1}^{3} f(x,y)h(x,1,y,2) = 20.1 \tag{1.33}$$

The other values are computed in a similar way. Finally, the output image is:

$$g = \begin{pmatrix} 27.3 & 20.1 & 18.9 \\ 19.4 & 16.0 & 18.4 \\ 16.0 & 29.3 & 33.4 \end{pmatrix} \tag{1.34}$$

Example 1.6

Is the operator of example 1.5 shift invariant?

No, it is not. For example, pixel $(2,2)$ *influences the value of pixel* $(1,2)$ *with weight* $h(2,1,2,2) = 0.7$. *These two pixels are at distance* $2-1=1$ *along the* x *axis and at distance* $2-2=0$ *along the* y *axis. At the same relative distance are also pixels* $(3,3)$ *and* $(2,3)$. *The value of* $h(3,2,3,3)$, *however, is* $0.8 \neq 0.7$.

Example 1.7

The point spread function of an operator that operates on images of size 3×3 is

$h(1,1,1,1) = 1.0$	$h(1,1,1,2) = 0.5$	$h(1,1,1,3) = 0.0$	$h(1,2,1,1) = 0.5$
$h(1,2,1,2) = 0.0$	$h(1,2,1,3) = 0.4$	$h(1,3,1,1) = 0.5$	$h(1,3,1,2) = 1.0$
$h(1,3,1,3) = 0.6$	$h(2,1,1,1) = 0.8$	$h(2,1,1,2) = 0.7$	$h(2,1,1,3) = 0.4$
$h(2,2,1,1) = 1.0$	$h(2,2,1,2) = 0.5$	$h(2,2,1,3) = 0.0$	$h(2,3,1,1) = 0.5$
$h(2,3,1,2) = 0.0$	$h(2,3,1,3) = 0.4$	$h(3,1,1,1) = 0.9$	$h(3,1,1,2) = 0.5$
$h(3,1,1,3) = 0.5$	$h(3,2,1,1) = 0.8$	$h(3,2,1,2) = 0.7$	$h(3,2,1,3) = 0.4$
$h(3,3,1,1) = 1.0$	$h(3,3,1,2) = 0.5$	$h(3,3,1,3) = 0.0$	$h(1,1,2,1) = 1.0$
$h(1,1,2,2) = 1.0$	$h(1,1,2,3) = 0.5$	$h(1,2,2,1) = 0.0$	$h(1,2,2,2) = 0.5$
$h(1,2,2,3) = 0.0$	$h(1,3,2,1) = 0.4$	$h(1,3,2,2) = 0.5$	$h(1,3,2,3) = 1.0$
$h(2,1,2,1) = 0.8$	$h(2,1,2,2) = 0.8$	$h(2,1,2,3) = 0.7$	$h(2,2,2,1) = 1.0$
$h(2,2,2,2) = 1.0$	$h(2,2,2,3) = 0.5$	$h(2,3,2,1) = 0.0$	$h(2,3,2,2) = 0.5$
$h(2,3,2,3) = 0.0$	$h(3,1,2,1) = 0.7$	$h(3,1,2,2) = 0.9$	$h(3,1,2,3) = 0.5$
$h(3,2,2,1) = 0.8$	$h(3,2,2,2) = 0.8$	$h(3,2,2,3) = 0.7$	$h(3,3,2,1) = 1.0$
$h(3,3,2,2) = 1.0$	$h(3,3,2,3) = 0.5$	$h(1,1,3,1) = 1.0$	$h(1,1,3,2) = 1.0$
$h(1,1,3,3) = 1.0$	$h(1,2,3,1) = 0.5$	$h(1,2,3,2) = 0.0$	$h(1,2,3,3) = 0.5$
$h(1,3,3,1) = 0.5$	$h(1,3,3,2) = 0.4$	$h(1,3,3,3) = 0.5$	$h(2,1,3,1) = 0.5$
$h(2,1,3,2) = 0.8$	$h(2,1,3,3) = 0.8$	$h(2,2,3,1) = 1.0$	$h(2,2,3,2) = 1.0$
$h(2,2,3,3) = 1.0$	$h(2,3,3,1) = 0.5$	$h(2,3,3,2) = 0.0$	$h(2,3,3,3) = 0.5$
$h(3,1,3,1) = 0.8$	$h(3,1,3,2) = 0.7$	$h(3,1,3,3) = 0.9$	$h(3,2,3,1) = 0.5$
$h(3,2,3,2) = 0.8$	$h(3,2,3,3) = 0.8$	$h(3,3,3,1) = 1.0$	$h(3,3,3,2) = 1.0$
$h(3,3,3,3) = 1.0$			

Is it shift variant or shift invariant?

In order to show that the function is shift variant, it is enough to show that for at least two pairs of pixels, that correspond to input-output pixels in the same relative position, the values of the function are different. This is what we did in example 1.6. If we cannot find such an example, we must then check for shift invariance. The function must have the same value for all pairs of input-output pixels that are in the same relative position in order to be shift invariant. As the range of values each of the arguments of this function takes is $[1,3]$, *the* relative *coordinates of pairs of input-output pixels take values in the range* $[-2,2]$. *We observe the following.*

For $x-\alpha = -2$ *and* $y-\beta = -2$ *we have:* $h(1,3,1,3) = 0.6$
For $x-\alpha = -2$ *and* $y-\beta = -1$ *we have:* $h(1,3,1,2) = h(1,3,2,3) = 1.0$
For $x-\alpha = -2$ *and* $y-\beta = 0$ *we have:* $h(1,3,1,1) = h(1,3,2,2) = h(1,3,3,3) = 0.5$
For $x-\alpha = -2$ *and* $y-\beta = 1$ *we have:* $h(1,3,2,1) = h(1,3,3,2) = 0.4$
For $x-\alpha = -2$ *and* $y-\beta = 2$ *we have:* $h(1,3,3,1) = 0.5$
For $x-\alpha = -1$ *and* $y-\beta = -2$ *we have:* $h(1,2,1,3) = h(2,3,1,3) = 0.4$
For $x-\alpha = -1$ *and* $y-\beta = -1$ *we have:*
$h(1,2,1,2) = h(2,3,2,3) = h(1,2,2,3) = h(2,3,1,2) = 0.0$
For $x-\alpha = -1$ *and* $y-\beta = 0$ *we have:*

$h(1,2,1,1) = h(1,2,2,2) = h(1,2,3,3) = h(2,3,1,1) = h(2,3,2,2) = h(2,3,3,3) = 0.5$
For $x - \alpha = -1$ *and* $y - \beta = 1$ *we have:*
$h(1,2,2,1) = h(1,2,3,2) = h(2,3,2,1) = h(2,3,3,2) = 0.0$
For $x - \alpha = -1$ *and* $y - \beta = 2$ *we have:* $h(1,2,3,1) = 0.5$
For $x - \alpha = 0$ *and* $y - \beta = -2$ *we have:* $h(1,1,1,3) = h(2,2,1,3) = h(3,3,1,3) = 0.0$
For $x - \alpha = 0$ *and* $y - \beta = -1$ *we have:*
$h(1,1,1,2) = h(2,2,1,2) = h(3,3,1,2) = h(1,1,2,3) = h(2,2,2,3) = h(3,3,2,3) = 0.5$
For $x - \alpha = 0$ *and* $y - \beta = 0$ *we have:*
$h(1,1,1,1) = h(2,2,1,1) = h(3,3,1,1) = h(1,1,2,2) = h(2,2,2,2) =$
$h(3,3,2,2) = h(1,1,3,3) = h(2,2,3,3) = h(3,3,3,3) = 1.0$
For $x - \alpha = 0$ *and* $y - \beta = 1$ *we have:*
$h(1,1,2,1) = h(2,2,2,1) = h(3,3,2,1) = h(1,1,3,2) = h(2,2,3,2) = h(3,3,3,2) = 1.0$
For $x - \alpha = 0$ *and* $y - \beta = 2$ *we have:* $h(1,1,3,1) = h(2,2,3,1) = h(3,3,3,1) = 1.0$
For $x - \alpha = 1$ *and* $y - \beta = -2$ *we have:* $h(2,1,1,3) = h(3,2,1,3) = 0.4$
For $x - \alpha = 1$ *and* $y - \beta = -1$ *we have:*
$h(2,1,1,2) = h(2,1,2,3) = h(3,2,1,2) = h(3,2,2,3) = 0.7$
For $x - \alpha = 1$ *and* $y - \beta = 0$ *we have:*
$h(2,1,1,1) = h(2,1,2,2) = h(2,1,3,3) = h(3,2,1,1) = h(3,2,2,2) = h(3,2,3,3) = 0.8$
For $x - \alpha = 1$ *and* $y - \beta = 1$ *we have:*
$h(2,1,2,1) = h(2,1,3,2) = h(3,2,2,1) = h(3,2,3,2) = 0.8$
For $x - \alpha = 1$ *and* $y - \beta = 2$ *we have:* $h(2,1,3,1) = h(3,2,3,1) = 0.5$
For $x - \alpha = 2$ *and* $y - \beta = -2$ *we have:* $h(3,1,1,3) = 0.5$
For $x - \alpha = 2$ *and* $y - \beta = -1$ *we have:* $h(3,1,1,2) = h(3,1,2,3) = 0.5$
For $x - \alpha = 2$ *and* $y - \beta = 0$ *we have:* $h(3,1,1,1) = h(3,1,2,2) = h(3,1,3,3) = 0.9$
For $x - \alpha = 2$ *and* $y - \beta = 1$ *we have;* $h(3,1,2,1) = h(3,1,3,2) = 0.7$
For $x - \alpha = 2$ *and* $y - \beta = 2$ *we have:* $h(3,1,3,1) = 0.8$

So, this is a shift invariant point spread function.

How can we express in practice the effect of a linear operator on an image?

This is done with the help of matrices. We can rewrite equation (1.15) as follows:

$$\begin{aligned} & g(\alpha,\beta) = \\ & f(1,1)h(1,\alpha,1,\beta) + f(2,1)h(2,\alpha,1,\beta) + \ldots + f(N,1)h(N,\alpha,1,\beta) \\ & + f(1,2)h(1,\alpha,2,\beta) + f(2,2)h(2,\alpha,2,\beta) + \ldots + f(N,2)h(N,\alpha,2,\beta) \\ & + \ldots + f(1,N)h(1,\alpha,N,\beta) + f(2,N)h(2,\alpha,N,\beta) + \ldots \\ & + f(N,N)h(N,\alpha,N,\beta) \end{aligned} \tag{1.35}$$

The right-hand side of this expression can be thought of as the dot product of vector

$$\begin{aligned} & \mathbf{h}_{\alpha\beta}^T \equiv \\ & [h(1,\alpha,1,\beta), h(2,\alpha,1,\beta), \ldots, h(N,\alpha,1,\beta), h(1,\alpha,2,\beta), h(2,\alpha,2,\beta), \ldots, \\ & h(N,\alpha,2,\beta), \ldots, h(2,\alpha,N,\beta), \ldots, h(N,\alpha,N,\beta)] \end{aligned} \tag{1.36}$$

with vector:

$$
\begin{aligned}
\mathbf{f}^T \equiv\ & [f(1,1), f(2,1), \ldots, f(N,1), f(1,2), f(2,2), \ldots, f(N,2), \\
& \ldots, f(1,N), f(2,N), \ldots, f(N,N)]
\end{aligned}
\tag{1.37}
$$

This last vector is actually the image $f(x,y)$ written as a vector by stacking its columns one under the other. If we imagine writing $g(\alpha,\beta)$ in the same way, then vectors $\mathbf{h}^T_{\alpha\beta}$ will arrange themselves as the rows of a matrix H, where for $\beta = 1$, α will run from 1 to N to give the first N rows of the matrix, then for $\beta = 2$, β will run again from 1 to N to give the second N rows of the matrix, and so on. Thus, equation (1.15) may be written in a more compact way as:

$$
\mathbf{g} = H\mathbf{f} \tag{1.38}
$$

This is the fundamental equation of linear image processing. H here is a square $N^2 \times N^2$ matrix that is made up of $N \times N$ submatrices of size $N \times N$ each, arranged in the following way:

$$
H = \begin{pmatrix}
\alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=1 \\ \beta=1 \end{pmatrix}} & \alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=2 \\ \beta=1 \end{pmatrix}} & \cdots & \alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=N \\ \beta=1 \end{pmatrix}} \\
\alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=1 \\ \beta=2 \end{pmatrix}} & \alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=2 \\ \beta=2 \end{pmatrix}} & \cdots & \alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=N \\ \beta=1 \end{pmatrix}} \\
\vdots & \vdots & & \vdots \\
\alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=1 \\ \beta=N \end{pmatrix}} & \alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=2 \\ \beta=N \end{pmatrix}} & \cdots & \alpha\!\downarrow\!\overset{x\rightarrow}{\begin{pmatrix} y=N \\ \beta=N \end{pmatrix}}
\end{pmatrix}
\tag{1.39}
$$

In this representation each bracketed expression represents an $N \times N$ submatrix made up from function $h(x,\alpha,y,\beta)$ for fixed values of y and β and with variables x and α taking up all their possible values in the directions indicated by the arrows. This schematic structure of matrix H is said to correspond to a **partition** of this matrix into N^2 square submatrices.

Example 1.8

A linear operator is such that it replaces the value of each pixel by the average of its four nearest neighbours. Apply this operator to a 3×3 image g. Assume that the image is repeated ad infinitum in all directions, so that all its pixels have neighbours. Work out the 9×9 matrix H that corresponds to this operator by using equation (1.39).

If the image is repeated in all directions, the image and the neighbours of its border pixels will look like this:

$$
\begin{array}{ccccccc}
g_{33} & | & g_{31} & g_{32} & g_{33} & | & g_{11} \\
- & - & - & - & - & - & - \\
g_{13} & | & g_{11} & g_{12} & g_{13} & | & g_{11} \\
g_{23} & | & g_{21} & g_{22} & g_{23} & | & g_{21} \\
g_{33} & | & g_{31} & g_{32} & g_{33} & | & g_{31} \\
- & - & - & - & - & - & - \\
g_{13} & | & g_{11} & g_{12} & g_{13} & | & g_{11}
\end{array}
\tag{1.40}
$$

The result then of replacing every pixel by the average of its four nearest neighbours is:

$$\begin{array}{ccc} \frac{g_{31}+g_{12}+g_{21}+g_{13}}{4} & \frac{g_{32}+g_{13}+g_{22}+g_{11}}{4} & \frac{g_{33}+g_{11}+g_{23}+g_{12}}{4} \\ \frac{g_{11}+g_{22}+g_{31}+g_{23}}{4} & \frac{g_{12}+g_{23}+g_{32}+g_{21}}{4} & \frac{g_{13}+g_{21}+g_{33}+g_{22}}{4} \\ \frac{g_{21}+g_{32}+g_{11}+g_{33}}{4} & \frac{g_{22}+g_{33}+g_{12}+g_{31}}{4} & \frac{g_{23}+g_{31}+g_{13}+g_{32}}{4} \end{array} \tag{1.41}$$

In order to construct matrix H we must deduce from the above result the weight by which a pixel at position (x, y) of the input image contributes to the value of a pixel at position (α, β) of the output image. Such value will be denoted by $h(x, \alpha, y, \beta)$. Matrix H will be made up from these values arranged as follows:

$$H = \left[\begin{array}{ccccc} h(1,1,1,1) & h(2,1,1,1) & h(3,1,1,1) & h(1,1,2,1) & h(2,1,2,1) \\ h(1,2,1,1) & h(2,2,1,1) & h(3,2,1,1) & h(1,2,2,1) & h(2,2,2,1) \\ h(1,3,1,1) & h(2,3,1,1) & h(3,3,1,1) & h(1,3,2,1) & h(2,3,2,1) \\ h(1,1,1,2) & h(2,1,1,2) & h(3,1,1,2) & h(1,1,2,2) & h(2,1,2,2) \\ h(1,2,1,2) & h(2,2,1,2) & h(3,2,1,2) & h(1,2,2,2) & h(2,2,2,2) \\ h(1,3,1,2) & h(2,3,1,2) & h(3,3,1,2) & h(1,3,2,2) & h(2,3,2,2) \\ h(1,1,1,3) & h(2,1,1,3) & h(3,1,1,3) & h(1,1,2,3) & h(2,1,2,3) \\ h(1,2,1,3) & h(2,2,1,3) & h(3,2,1,3) & h(1,2,2,3) & h(2,2,2,3) \\ h(1,3,1,3) & h(2,3,1,3) & h(3,3,1,3) & h(1,3,2,3) & h(2,3,2,3) \end{array} \right.$$

$$\left. \begin{array}{cccc} h(3,1,2,1) & h(1,1,3,1) & h(2,1,3,1) & h(3,1,3,1) \\ h(3,2,2,1) & h(1,2,3,1) & h(2,2,3,1) & h(3,2,3,1) \\ h(3,3,2,1) & h(1,3,3,1) & h(2,3,3,1) & h(3,3,3,1) \\ h(3,1,2,2) & h(1,1,3,2) & h(2,1,3,2) & h(3,1,3,2) \\ h(3,2,2,2) & h(1,2,3,2) & h(2,2,3,2) & h(3,2,3,2) \\ h(3,3,2,2) & h(1,3,3,2) & h(2,3,3,2) & h(3,3,3,2) \\ h(3,1,2,3) & h(1,1,3,3) & h(2,1,3,3) & h(3,1,3,3) \\ h(3,2,2,3) & h(1,2,3,3) & h(2,2,3,3) & h(3,2,3,3) \\ h(3,3,2,3) & h(1,3,3,3) & h(2,3,3,3) & h(3,3,3,3) \end{array} \right] \tag{1.42}$$

By inspection we deduce that:

$$H = \begin{bmatrix} 0 & 1/4 & 1/4 & 1/4 & 0 & 0 & 1/4 & 0 & 0 \\ 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 0 & 1/4 & 0 \\ 1/4 & 1/4 & 0 & 0 & 0 & 1/4 & 0 & 0 & 1/4 \\ 1/4 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 \\ 0 & 0 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 1/4 \\ 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 \\ 0 & 1/4 & 0 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \\ 0 & 0 & 1/4 & 0 & 0 & 1/4 & 1/4 & 1/4 & 0 \end{bmatrix} \tag{1.43}$$

Example 1.9

The effect of a linear operator is to subtract from every pixel its right neighbour. This operator is applied to image (1.40). Work out the output image and the H matrix that corresponds to this operator.

The result of this operator will be:

$$\begin{array}{ccc} g_{11} - g_{12} & g_{12} - g_{13} & g_{13} - g_{11} \\ g_{21} - g_{22} & g_{22} - g_{23} & g_{23} - g_{21} \\ g_{31} - g_{32} & g_{32} - g_{33} & g_{33} - g_{31} \end{array} \tag{1.44}$$

By following the procedure we followed in example 1.8, we can work out matrix H, which here, for convenience, we call $\hat{H}$:

$$\hat{H} = \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.45}$$

Example 1.10

The effect of a linear operator is to subtract from every pixel its bottom right neighbour. This operator is applied to image (1.40). Work out the output image and the H matrix that corresponds to this operator.

The result of this operator will be:

$$\begin{array}{ccc} g_{11} - g_{22} & g_{12} - g_{23} & g_{13} - g_{21} \\ g_{21} - g_{32} & g_{22} - g_{33} & g_{23} - g_{31} \\ g_{31} - g_{12} & g_{32} - g_{13} & g_{33} - g_{11} \end{array} \tag{1.46}$$

By following the procedure we followed in example 1.8, we can work out matrix H, *which here, for convenience, we call $\tilde{H}$:*

$$\tilde{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.47}$$

Can we apply more than one linear operators to an image?

We can apply as many operators as we like.

Does the order by which we apply the linear operators make any difference to the result?

No, if the operators are shift invariant. This is a very important and convenient property of the commonly used linear operators.

Box 1.2. Since matrix multiplication is not commutative, how come we can change the order by which we apply shift invariant linear operators?

In general, if A and B are two matrices, $AB \neq BA$. However, matrix H, which expresses the effect of a linear operator, has a particular structure. We can see that from equation (1.39): the $N^2 \times N^2$ matrix H can be divided into N^2 submatrices of size $N \times N$ each. There are at most N distinct such submatrices and they have a particular structure: the second column of their elements can be produced from the first column, by shifting all elements by one position down. The element that sticks out at the bottom is put at the empty position at the top. The third column is produced by applying the same procedure to the second column as so on (see example 1.11). A matrix that has this property is called **circulant matrix**. At the same time, these submatrices are also arranged in a circulant way: the second column of them is produced from the first column by shifting all of them by one position down. The submatrix that sticks out at the bottom is put at the empty position at the top. The third column of matrices is created from the second by applying the same procedure and so on (see example 1.12). A matrix that has this property is called **block circulant**. It is this particular structure that allows one to exchange the order by which two operators with matrices H and $\tilde{H}$, respectively, are applied to an image $\mathbf{g}$ written as a vector:

$$H\tilde{H}\mathbf{g} = \tilde{H}H\mathbf{g} \tag{1.48}$$

Example B1.11

Write down the form you expect two 3×3 circulant matrices to have and show that their product commutes.

A 3×3 circulant matrix will contain at most 3 distinct elements. Let us call them α, β and γ for matrix A, and $\tilde{\alpha}$, $\tilde{\beta}$ and $\tilde{\gamma}$ for matrix $\tilde{A}$. Since these matrices are circulant, they must have the following structure:

$$A = \begin{pmatrix} \alpha & \gamma & \beta \\ \beta & \alpha & \gamma \\ \gamma & \beta & \alpha \end{pmatrix} \qquad \tilde{A} = \begin{pmatrix} \tilde{\alpha} & \tilde{\gamma} & \tilde{\beta} \\ \tilde{\beta} & \tilde{\alpha} & \tilde{\gamma} \\ \tilde{\gamma} & \tilde{\beta} & \tilde{\alpha} \end{pmatrix} \tag{1.49}$$

We can work out the product $A\tilde{A}$:

$$A\tilde{A} = \begin{pmatrix} \alpha & \gamma & \beta \\ \beta & \alpha & \gamma \\ \gamma & \beta & \alpha \end{pmatrix} \begin{pmatrix} \tilde{\alpha} & \tilde{\gamma} & \tilde{\beta} \\ \tilde{\beta} & \tilde{\alpha} & \tilde{\gamma} \\ \tilde{\gamma} & \tilde{\beta} & \tilde{\alpha} \end{pmatrix} = \begin{pmatrix} \alpha\tilde{\alpha}+\gamma\tilde{\beta}+\beta\tilde{\gamma} & \alpha\tilde{\gamma}+\gamma\tilde{\alpha}+\beta\tilde{\beta} & \alpha\tilde{\beta}+\gamma\tilde{\gamma}+\beta\tilde{\alpha} \\ \beta\tilde{\alpha}+\alpha\tilde{\beta}+\gamma\tilde{\gamma} & \beta\tilde{\gamma}+\alpha\tilde{\alpha}+\gamma\tilde{\beta} & \beta\tilde{\beta}+\alpha\tilde{\gamma}+\gamma\tilde{\alpha} \\ \gamma\tilde{\alpha}+\beta\tilde{\beta}+\alpha\tilde{\gamma} & \gamma\tilde{\gamma}+\beta\tilde{\alpha}+\alpha\tilde{\beta} & \gamma\tilde{\alpha}+\beta\tilde{\gamma}+\alpha\tilde{\alpha} \end{pmatrix} \tag{1.50}$$

We get the same result by working out $\tilde{A}A$:

$$\tilde{A}A = \begin{pmatrix} \tilde{\alpha} & \tilde{\gamma} & \tilde{\beta} \\ \tilde{\beta} & \tilde{\alpha} & \tilde{\gamma} \\ \tilde{\gamma} & \tilde{\beta} & \tilde{\alpha} \end{pmatrix} \begin{pmatrix} \alpha & \gamma & \beta \\ \beta & \alpha & \gamma \\ \gamma & \beta & \alpha \end{pmatrix} = \begin{pmatrix} \tilde{\alpha}\alpha+\tilde{\gamma}\beta+\tilde{\beta}\gamma & \tilde{\alpha}\gamma+\tilde{\gamma}\alpha+\tilde{\beta}\beta & \tilde{\alpha}\beta+\tilde{\gamma}\gamma+\tilde{\beta}\alpha \\ \tilde{\beta}\alpha+\tilde{\alpha}\beta+\tilde{\gamma}\gamma & \tilde{\beta}\gamma+\tilde{\alpha}\alpha+\tilde{\gamma}\beta & \tilde{\beta}\beta+\tilde{\alpha}\gamma+\tilde{\gamma}\alpha \\ \tilde{\gamma}\alpha+\tilde{\beta}\beta+\tilde{\alpha}\gamma & \tilde{\gamma}\gamma+\tilde{\beta}\alpha+\tilde{\alpha}\beta & \tilde{\gamma}\beta+\tilde{\beta}\gamma+\tilde{\alpha}\alpha \end{pmatrix} \tag{1.51}$$

We can see that $A\tilde{A} = \tilde{A}A$, ie that these two matrices commute.

Example B1.12

Identify the 3×3 submatrices from which the H matrices of examples 1.8, 1.9 and 1.10 are made.

We can easily identify that matrix H of example 1.8 is made up from submatrices:

$$H_{11} \equiv \begin{pmatrix} 0 & 1/4 & 1/4 \\ 1/4 & 0 & 1/4 \\ 1/4 & 1/4 & 0 \end{pmatrix} \quad H_{21} = H_{31} \equiv \begin{pmatrix} 1/4 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 1/4 \end{pmatrix} \tag{1.52}$$

Matrix $\hat{H}$ of example 1.9 is made up from submatrices:

$$\hat{H}_{11} \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \hat{H}_{21} \equiv \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \hat{H}_{31} \equiv \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{1.53}$$

Matrix $\tilde{H}$ of example 1.10 is made up from submatrices:

$$\tilde{H}_{11} \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \tilde{H}_{21} \equiv \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \tilde{H}_{31} \equiv \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{pmatrix} \tag{1.54}$$

Thus, matrices H, $\hat{H}$ and $\tilde{H}$ of examples 1.8, 1.9 and 1.10, respectively, may be written as:

$$H = \begin{bmatrix} H_{11} & H_{31} & H_{21} \\ H_{21} & H_{11} & H_{31} \\ H_{31} & H_{21} & H_{11} \end{bmatrix} \quad \hat{H} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{31} & \hat{H}_{21} \\ \hat{H}_{21} & \hat{H}_{11} & \hat{H}_{31} \\ \hat{H}_{31} & \hat{H}_{21} & \hat{H}_{11} \end{bmatrix} \quad \tilde{H} = \begin{bmatrix} \tilde{H}_{11} & \tilde{H}_{31} & \tilde{H}_{21} \\ \tilde{H}_{21} & \tilde{H}_{11} & \tilde{H}_{31} \\ \tilde{H}_{31} & \tilde{H}_{21} & \tilde{H}_{11} \end{bmatrix} \tag{1.55}$$

Each one of the submatrices is circulant and they are arranged in a circulant manner, so matrices H, $\hat{H}$ and $\tilde{H}$ are block circulant.

Example 1.13

Apply the operator of example 1.9 to the output image (1.41) of example 1.8 by working directly on the output image. Then apply the operator of example 1.8 to the output image (1.44) of example 1.9. Compare the two answers.

The operator of example 1.9 subtracts from every pixel the value of its right neighbour. We remember that the image is assumed to be wrapped round so that all pixels have neighbours in all directions. We perform this operation on image (1.41) and obtain:

$$\begin{bmatrix} \frac{g_{31}+g_{12}+g_{21}-g_{32}-g_{22}-g_{11}}{4} & \frac{g_{32}+g_{13}+g_{22}-g_{33}-g_{23}-g_{12}}{4} & \frac{g_{33}+g_{11}+g_{23}-g_{31}-g_{21}-g_{13}}{4} \\ \frac{g_{11}+g_{22}+g_{31}-g_{12}-g_{32}-g_{21}}{4} & \frac{g_{12}+g_{23}+g_{32}-g_{13}-g_{33}-g_{22}}{4} & \frac{g_{13}+g_{21}+g_{33}-g_{11}-g_{31}-g_{23}}{4} \\ \frac{g_{21}+g_{32}+g_{11}-g_{22}-g_{12}-g_{31}}{4} & \frac{g_{22}+g_{33}+g_{12}-g_{23}-g_{13}-g_{32}}{4} & \frac{g_{23}+g_{31}+g_{13}-g_{21}-g_{11}-g_{33}}{4} \end{bmatrix} \tag{1.56}$$

The operator of example 1.8 replaces every pixel with the average of its four neighbours.

We apply it to image (1.44) and obtain:

$$
\begin{bmatrix}
\frac{g_{31}-g_{32}+g_{12}-g_{13}+g_{21}-g_{22}+g_{13}-g_{11}}{4} & \frac{g_{32}-g_{33}+g_{13}-g_{11}+g_{22}-g_{23}+g_{11}-g_{12}}{4} & \frac{g_{33}-g_{31}+g_{11}-g_{12}+g_{23}-g_{21}+g_{12}-g_{13}}{4} \\
\frac{g_{11}-g_{12}+g_{22}-g_{23}+g_{31}-g_{32}+g_{23}-g_{21}}{4} & \frac{g_{12}-g_{13}+g_{23}-g_{21}+g_{32}-g_{33}+g_{21}-g_{22}}{4} & \frac{g_{13}-g_{11}+g_{21}-g_{22}+g_{33}-g_{31}+g_{22}-g_{23}}{4} \\
\frac{g_{21}-g_{22}+g_{32}-g_{33}+g_{11}-g_{12}+g_{33}-g_{31}}{4} & \frac{g_{22}-g_{23}+g_{33}-g_{31}+g_{12}-g_{13}+g_{31}-g_{32}}{4} & \frac{g_{23}-g_{21}+g_{31}-g_{32}+g_{13}-g_{11}+g_{32}-g_{33}}{4}
\end{bmatrix}
\tag{1.57}
$$

By comparing outputs (1.56) and (1.57) we see that we get the same answer whichever is the order by which we apply the operators.

Example 1.14

Use matrix multiplication to derive the matrix with which an image (written in vector form) has to be multiplied from the left so that the operators of examples 1.8 and 1.9 are applied in a cascaded way. Does the answer depend on the order by which the operators are applied?

If we apply first the operator of example 1.9 and then the operator of example 1.8, we must compute the product $H\hat{H}$ of matrices H and $\hat{H}$ given by equations (1.43) and (1.45), respectively:

$$
\begin{aligned}
H\hat{H} &= \begin{bmatrix}
0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\
\frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 \\
\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\
\frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\
0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\
0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} \\
\frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\
0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\
0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\
-1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix} \\
&= \begin{bmatrix}
-1/4 & 1/4 & 1/4 & 1/4 & -1/4 & -1/4 & 0 & 0 & 0 \\
1/4 & -1/4 & 1/4 & -1/4 & 1/4 & -1/4 & 0 & 0 & 0 \\
1/4 & 1/4 & -1/4 & -1/4 & -1/4 & 1/4 & 0 & 0 & 0 \\
0 & 0 & 0 & -1/4 & 1/4 & 1/4 & 1/4 & -1/4 & -1/4 \\
0 & 0 & 0 & 1/4 & -1/4 & 1/4 & -1/4 & 1/4 & -1/4 \\
0 & 0 & 0 & 1/4 & 1/4 & -1/4 & -1/4 & -1/4 & 1/4 \\
1/4 & -1/4 & -1/4 & 0 & 0 & 0 & -1/4 & 1/4 & 1/4 \\
-1/4 & 1/4 & -1/4 & 0 & 0 & 0 & 1/4 & -1/4 & 1/4 \\
-1/4 & -1/4 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 & -1/4
\end{bmatrix}
\end{aligned}
\tag{1.58}
$$

If we apply first the operator of example 1.8 and then the operator of example 1.9, we must compute the product $\hat{H}H$ of matrices H and $\hat{H}$ given by equations (1.43) and (1.45), respectively:

$$\hat{H}H =$$

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -1/4 & 1/4 & 1/4 & 1/4 & -1/4 & -1/4 & 0 & 0 & 0 \\ 1/4 & -1/4 & 1/4 & -1/4 & 1/4 & -1/4 & 0 & 0 & 0 \\ 1/4 & 1/4 & -1/4 & -1/4 & -1/4 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1/4 & 1/4 & 1/4 & 1/4 & -1/4 & -1/4 \\ 0 & 0 & 0 & 1/4 & -1/4 & 1/4 & -1/4 & 1/4 & -1/4 \\ 0 & 0 & 0 & 1/4 & 1/4 & -1/4 & -1/4 & -1/4 & 1/4 \\ 1/4 & -1/4 & -1/4 & 0 & 0 & 0 & -1/4 & 1/4 & 1/4 \\ -1/4 & 1/4 & -1/4 & 0 & 0 & 0 & 1/4 & -1/4 & 1/4 \\ -1/4 & -1/4 & 1/4 & 0 & 0 & 0 & 1/4 & 1/4 & -1/4 \end{bmatrix} \tag{1.59}$$

By comparing results (1.58) and (1.59) we see that the order by which we multiply the matrices, and by extension the order by which we apply the operators, does not matter.

Example 1.15

Process image (1.40) with matrix (1.59) and compare your answer with the output images produced in example 1.13.

To process image (1.40) by matrix (1.59), we must write it in vector form, by stacking its columns one under the other:

$$
\begin{bmatrix}
-\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 \\
\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\
0 & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\
0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\
\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & -\frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
-\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \\
-\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4}
\end{bmatrix}
\begin{bmatrix} g_{11} \\ g_{21} \\ g_{31} \\ g_{12} \\ g_{22} \\ g_{32} \\ g_{13} \\ g_{23} \\ g_{33} \end{bmatrix}
=
\begin{bmatrix}
\frac{-g_{11}+g_{21}+g_{31}+g_{12}-g_{22}-g_{32}}{4} \\
\frac{g_{11}-g_{21}+g_{31}-g_{12}+g_{22}-g_{32}}{4} \\
\frac{g_{11}+g_{21}-g_{31}-g_{12}-g_{22}+g_{32}}{4} \\
\frac{-g_{12}+g_{22}+g_{32}+g_{13}-g_{23}-g_{33}}{4} \\
\frac{g_{12}-g_{22}+g_{32}-g_{13}+g_{23}-g_{33}}{4} \\
\frac{g_{12}+g_{22}-g_{32}-g_{13}-g_{23}+g_{33}}{4} \\
\frac{g_{11}-g_{21}-g_{31}-g_{13}+g_{23}+g_{33}}{4} \\
\frac{-g_{11}+g_{21}-g_{31}+g_{13}-g_{23}+g_{33}}{4} \\
\frac{-g_{11}-g_{21}+g_{31}+g_{13}+g_{23}-g_{33}}{4}
\end{bmatrix}
\tag{1.60}
$$

To create an image out of the output vector (1.60), we have to use its first three elements as the first column of the image, the next three elements as the second column of the image, and so on. The image we obtain is:

$$
\begin{bmatrix}
\frac{-g_{11}+g_{21}+g_{31}+g_{12}-g_{22}-g_{32}}{4} & \frac{-g_{12}+g_{22}+g_{32}+g_{13}-g_{23}-g_{33}}{4} & \frac{g_{11}-g_{21}-g_{31}-g_{13}+g_{23}+g_{33}}{4} \\
\frac{g_{11}-g_{21}+g_{31}-g_{12}+g_{22}-g_{32}}{4} & \frac{g_{12}-g_{22}+g_{32}-g_{13}+g_{23}-g_{33}}{4} & \frac{-g_{11}+g_{21}-g_{31}+g_{13}-g_{23}+g_{33}}{4} \\
\frac{g_{11}+g_{21}-g_{31}-g_{12}-g_{22}+g_{32}}{4} & \frac{g_{12}+g_{22}-g_{32}-g_{13}-g_{23}+g_{33}}{4} & \frac{-g_{11}-g_{21}+g_{31}+g_{13}+g_{23}-g_{33}}{4}
\end{bmatrix}
\tag{1.61}
$$

By comparing (1.61) and (1.56) we see that we obtain the same output image either we apply the operators locally, or we operate on the whole image by using the corresponding matrix.

Example 1.16

By examining matrices H, $\hat{H}$ and $\tilde{H}$ given by equations (1.43), (1.45) and (1.47), respectively, deduce the point spread function of the corresponding operators.

As these operators are shift invariant, by definition, in order to work out their point spread functions starting from their corresponding H matrix, we have to reason about the structure of the matrix as exemplified by equation (1.39). The point spread function will be the same for all pixels of the input image, so we might as well pick one of them; say we pick the pixel in the middle of the input image, with coordinates $x = 2$ and $y = 2$. Then, we have to read the values of the elements of the matrix that correspond to all possible combinations of α and β, which indicate the coordinates of the output image. According to equation (1.39), β takes all its possible values along a column of submatrices, while α takes all its possible values along a column of any one of the submatrices. Fixing the value of $x = 2$ means that we shall have to read only the middle column of the submatrix we use. Fixing the value of $y = 2$ means that we shall have to read only the middle column of submatrices. The middle column then of *matrix H, when wrapped to form a 3×3 image, will be the point spread function of the operator, ie it will represent the output image we shall get if the operator is applied to an input image that consists of only 0s except in the central pixel where it has value 1 (a single point source). We have to write the first three elements of the central column of the H matrix as the first column of the output image, the next three elements as the second column, and the last three elements as the last column of the output image. For the three operators that correspond to matrices H, $\hat{H}$ and $\tilde{H}$, we obtain:*

$$h = \begin{pmatrix} 0 & \frac{1}{4} & 0 \\ \frac{1}{4} & 0 & \frac{1}{4} \\ 0 & \frac{1}{4} & 0 \end{pmatrix} \quad \hat{h} = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \tilde{h} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{1.62}$$

Example 1.17

What will be the effect of the operator that corresponds to matrix (1.59) on a 3×3 input image that depicts a point source in the middle?

The output image will be the point spread function of the operator. Following the reasoning of example 1.16, we deduce that the point spread function of this composite operator is:

$$\begin{pmatrix} -1/4 & 1/4 & 0 \\ 1/4 & -1/4 & 0 \\ -1/4 & 1/4 & 0 \end{pmatrix} \tag{1.63}$$

Example 1.18

What will be the effect of the operator that corresponds to matrix (1.59) on a 3×3 input image that depicts a point source in position $(1,2)$?

In this case we want to work out the output for $x=1$ and $y=2$. We select from matrix (1.59) the second column of submatrices, and the first column of them and wrap it to form a 3×3 image. We obtain:

$$\begin{pmatrix} 1/4 & -1/4 & 0 \\ -1/4 & 1/4 & 0 \\ -1/4 & 1/4 & 0 \end{pmatrix} \tag{1.64}$$

Alternatively, we multiply from the left input image

$$g^T = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1.65}$$

with matrix $H\hat{H}$ given by equation (1.58) and write the output vector as an image. We get exactly the same answer.

Box 1.3. What is the stacking operator?

The stacking operator allows us to write an $N \times N$ image array as an $N^2 \times 1$ vector, or an $N^2 \times 1$ vector as an $N \times N$ square array.

We define some vectors $\mathbf{V_n}$ and some matrices N_n as:

$$\mathbf{V_n} \equiv \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{rows } 1 \text{ to } n-1 \\ \left.\right\} \text{row } n \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \text{rows } n+1 \text{ to } N \end{array} \tag{1.66}$$

$$N_n \equiv \begin{bmatrix} & \mathbf{0} & & \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \\ & \mathbf{0} & & \end{bmatrix} \begin{array}{l} \left.\right\} n-1 \text{ square } N \times N \text{ matrices on top of each other with all their elements } 0 \\ \left.\right\} \text{the } n^{th} \text{ matrix is the unit matrix} \\ \left.\right\} N-n \text{ square } N \times N \text{ matrices on the top of each other with all their elements } 0 \end{array} \tag{1.67}$$

The dimensions of $\mathbf{V_n}$ are $N \times 1$ and of N_n $N^2 \times N$. Then vector $\mathbf{f}$ which corresponds to the $N \times N$ square matrix f is given by:

$$\mathbf{f} = \sum_{n=1}^{N} N_n f \mathbf{V_n} \tag{1.68}$$

It can be shown that if $\mathbf{f}$ is an $N^2 \times 1$ vector, we can write it as an $N \times N$ matrix f, the first column of which is made up from the first N elements of $\mathbf{f}$, the second column from the second N elements of $\mathbf{f}$, and so on, by using the following expression:

$$f = \sum_{n=1}^{N} N_n^T \mathbf{f} \mathbf{V_n}^T \tag{1.69}$$

Example B1.19

You are given a 3 × 3 image f and you are asked to use the stacking operator to write it in vector form.

Let us say that:

$$f = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \tag{1.70}$$

We define vectors $\mathbf{V_n}$ and matrices N_n for $n = 1, 2, 3$:

$$\mathbf{V_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{V_2} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{V_3} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{1.71}$$

$$N_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad N_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad N_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1.72}$$

According to equation (1.68):

$$\mathbf{f} = N_1 f \mathbf{V_1} + N_2 f \mathbf{V_2} + N_3 f \mathbf{V_3} \tag{1.73}$$

We shall calculate each term separately:

$$N_1 f \mathbf{V_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \Rightarrow$$

$$N_1 f \mathbf{V_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix} = \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{1.74}$$

Similarly:

$$N_2 f \mathbf{V_2} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ f_{12} \\ f_{22} \\ f_{32} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad N_3 f \mathbf{V_3} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix} \tag{1.75}$$

Then by substituting into (1.73), we obtain vector $\mathbf{f}$.

Example B1.20

You are given a 9×1 vector f. Use the stacking operator to write it as a 3×3 matrix.

Let us say that:

$$\mathbf{f} = \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix} \tag{1.76}$$

According to equation (1.69)

$$f = N_1^T \mathbf{f} \mathbf{V_1}^T + N_2^T \mathbf{f} \mathbf{V_2}^T + N_3^T \mathbf{f} \mathbf{V_3}^T \tag{1.77}$$

(where N_1, N_2, N_3, $\mathbf{V_1}$, $\mathbf{V_2}$ and $\mathbf{V_3}$ are defined in Box 1.3)
We shall calculate each term separately:

$$N_1^T \mathbf{f} \mathbf{V_1}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} & 0 & 0 \\ f_{21} & 0 & 0 \\ f_{31} & 0 & 0 \\ f_{12} & 0 & 0 \\ f_{22} & 0 & 0 \\ f_{32} & 0 & 0 \\ f_{13} & 0 & 0 \\ f_{23} & 0 & 0 \\ f_{33} & 0 & 0 \end{pmatrix} = \begin{pmatrix} f_{11} & 0 & 0 \\ f_{21} & 0 & 0 \\ f_{31} & 0 & 0 \end{pmatrix} \tag{1.78}$$

Similarly:

$$N_2^T \mathbf{f} \mathbf{V_2}^T = \begin{pmatrix} 0 & f_{12} & 0 \\ 0 & f_{22} & 0 \\ 0 & f_{32} & 0 \end{pmatrix}, \qquad N_3^T \mathbf{f} \mathbf{V_3}^T = \begin{pmatrix} 0 & 0 & f_{13} \\ 0 & 0 & f_{23} \\ 0 & 0 & f_{33} \end{pmatrix} \tag{1.79}$$

Then by substituting into (1.77), we obtain matrix f.

Example B1.21

Show that the stacking operator is linear.

To show that an operator is linear we must show that we shall get the same answer either we apply it to two images w and g and sum up the results with weights α and β, respectively, or we apply it to the weighted sum $\alpha w+\beta g$ directly. We start by applying the stacking operator to composite image $\alpha w+\beta g$, following equation (1.68):

$$\begin{aligned}\sum_{n=1}^{N} N_n(\alpha w+\beta g)\mathbf{V_n} &= \sum_{n=1}^{N} N_n(\alpha w\mathbf{V_n}+\beta g\mathbf{V_n})\\ &= \sum_{n=1}^{N}(N_n\alpha w\mathbf{V_n}+N_n\beta g\mathbf{V_n})\\ &= \sum_{n=1}^{N} N_n\alpha w\mathbf{V_n}+\sum_{n=1}^{N} N_n\beta g\mathbf{V_n}\end{aligned} \tag{1.80}$$

Since α and β do not depend on the summing index n, they may come out of the summands:

$$\sum_{n=1}^{N} N_n(\alpha w+\beta g)\mathbf{V_n}=\alpha\sum_{n=1}^{N} N_n w\mathbf{V_n}+\beta\sum_{n=1}^{N} N_n g\mathbf{V_n} \tag{1.81}$$

Then, we define vector $\mathbf{w}$ to be the vector version of image w, given by $\sum_{n=1}^{N} N_n w\mathbf{V_n}$ and vector $\mathbf{g}$ to be the vector version of image g, given by $\sum_{n=1}^{N} N_n g\mathbf{V_n}$, to obtain:

$$\sum_{n=1}^{N} N_n(\alpha w+\beta g)\mathbf{V_n}=\alpha\mathbf{w}+\beta\mathbf{g} \tag{1.82}$$

This proves that the stacking operator is linear, because if we apply it separately to images w and g we shall get vectors $\mathbf{w}$ and $\mathbf{g}$, respectively, which, when added with weights α and β, will produce the result we obtained above by applying the operator to the composite image $\alpha w+\beta g$ directly.

Example B1.22

Consider a 9×9 matrix H that is partitioned into nine 3×3 submatrices. Show that if we multiply it from the left with matrix N_2^T, defined in Box 1.3, we shall extract the second row of its submatrices.

We apply definition (1.67) for $N = 3$ and $n = 2$ to define matrix N_2 and we write explicitly all the elements of matrix H before we perform the multiplication:

$$N_2^T H = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \times$$

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} & | & h_{14} & h_{15} & h_{16} & | & h_{17} & h_{18} & h_{19} \\ h_{21} & h_{22} & h_{23} & | & h_{24} & h_{25} & h_{26} & | & h_{27} & h_{28} & h_{29} \\ h_{31} & h_{32} & h_{33} & | & h_{34} & h_{35} & h_{36} & | & h_{37} & h_{38} & h_{39} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{41} & h_{42} & h_{43} & | & h_{44} & h_{45} & h_{46} & | & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & | & h_{54} & h_{55} & h_{56} & | & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & | & h_{64} & h_{65} & h_{66} & | & h_{67} & h_{68} & h_{69} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{71} & h_{72} & h_{73} & | & h_{74} & h_{75} & h_{76} & | & h_{77} & h_{78} & h_{79} \\ h_{81} & h_{82} & h_{83} & | & h_{84} & h_{85} & h_{86} & | & h_{87} & h_{88} & h_{89} \\ h_{91} & h_{92} & h_{93} & | & h_{94} & h_{95} & h_{96} & | & h_{97} & h_{98} & h_{99} \end{pmatrix}$$

$$= \begin{pmatrix} h_{41} & h_{42} & h_{43} & | & h_{44} & h_{45} & h_{46} & | & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & | & h_{54} & h_{55} & h_{56} & | & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & | & h_{64} & h_{65} & h_{66} & | & h_{67} & h_{68} & h_{69} \end{pmatrix} \tag{1.83}$$

We note that the result is a matrix made up from the middle row of partitions of the original matrix H.

Example B1.23

Consider a 9×9 matrix H that is partitioned into nine 3×3 submatrices. Show that if we multiply it from the right with matrix N_3, defined in Box 1.3, we shall extract the third column of its submatrices.

We apply definition (1.67) for $N = 3$ and $n = 3$ to define matrix N_3 and we write explicitly all the elements of matrix H before we perform the multiplication:

$$HN_3 =$$

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} & | & h_{14} & h_{15} & h_{16} & | & h_{17} & h_{18} & h_{19} \\ h_{21} & h_{22} & h_{23} & | & h_{24} & h_{25} & h_{26} & | & h_{27} & h_{28} & h_{29} \\ h_{31} & h_{32} & h_{33} & | & h_{34} & h_{35} & h_{36} & | & h_{37} & h_{38} & h_{39} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{41} & h_{42} & h_{43} & | & h_{44} & h_{45} & h_{46} & | & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & | & h_{54} & h_{55} & h_{56} & | & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & | & h_{64} & h_{65} & h_{66} & | & h_{67} & h_{68} & h_{69} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{71} & h_{72} & h_{73} & | & h_{74} & h_{75} & h_{76} & | & h_{77} & h_{78} & h_{79} \\ h_{81} & h_{82} & h_{83} & | & h_{84} & h_{85} & h_{86} & | & h_{87} & h_{88} & h_{89} \\ h_{91} & h_{92} & h_{93} & | & h_{94} & h_{95} & h_{96} & | & h_{97} & h_{98} & h_{99} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} h_{17} & h_{18} & h_{19} \\ h_{27} & h_{28} & h_{29} \\ h_{37} & h_{38} & h_{39} \\ - & - & - \\ h_{47} & h_{48} & h_{49} \\ h_{57} & h_{58} & h_{59} \\ h_{67} & h_{68} & h_{69} \\ - & - & - \\ h_{77} & h_{78} & h_{79} \\ h_{87} & h_{88} & h_{89} \\ h_{97} & h_{98} & h_{99} \end{pmatrix} \tag{1.84}$$

We observe that the result is a matrix made up from the last column of the partitions of the original matrix H.

Example B1.24

Multiply the 3×9 matrix produced in example 1.22 with the 9×3 matrix produced in example 1.23 and show that the resultant 3×3 matrix is the sum of the individual multiplications of the corresponding partitions.

$$
\begin{pmatrix}
h_{41} & h_{42} & h_{43} & | & h_{44} & h_{45} & h_{46} & | & h_{47} & h_{48} & h_{49} \\
h_{51} & h_{52} & h_{53} & | & h_{54} & h_{55} & h_{56} & | & h_{57} & h_{58} & h_{59} \\
h_{61} & h_{62} & h_{63} & | & h_{64} & h_{65} & h_{66} & | & h_{67} & h_{68} & h_{69}
\end{pmatrix}
\begin{pmatrix}
h_{17} & h_{18} & h_{19} \\
h_{27} & h_{28} & h_{29} \\
h_{37} & h_{38} & h_{39} \\
- & - & - \\
h_{47} & h_{48} & h_{49} \\
h_{57} & h_{58} & h_{59} \\
h_{67} & h_{68} & h_{69} \\
- & - & - \\
h_{77} & h_{78} & h_{79} \\
h_{87} & h_{88} & h_{89} \\
h_{97} & h_{98} & h_{99}
\end{pmatrix} =
$$

$$
\left(\begin{matrix}
h_{41}h_{17} + h_{42}h_{27} + h_{43}h_{37} + h_{44}h_{47} + h_{45}h_{57} + h_{46}h_{67} + h_{47}h_{77} + h_{48}h_{87} + h_{49}h_{97} \\
h_{51}h_{17} + h_{52}h_{27} + h_{53}h_{37} + h_{54}h_{47} + h_{55}h_{57} + h_{56}h_{67} + h_{57}h_{77} + h_{58}h_{87} + h_{59}h_{97} \\
h_{61}h_{17} + h_{62}h_{27} + h_{63}h_{37} + h_{64}h_{47} + h_{65}h_{57} + h_{66}h_{67} + h_{67}h_{77} + h_{68}h_{87} + h_{69}h_{97}
\end{matrix}\right.
$$

$$
\begin{matrix}
h_{41}h_{18} + h_{42}h_{28} + h_{43}h_{38} + h_{44}h_{48} + h_{45}h_{58} + h_{46}h_{68} + h_{47}h_{78} + h_{48}h_{88} + h_{49}h_{98} \\
h_{51}h_{18} + h_{52}h_{28} + h_{53}h_{38} + h_{54}h_{48} + h_{55}h_{58} + h_{56}h_{68} + h_{57}h_{78} + h_{58}h_{88} + h_{59}h_{98} \\
h_{61}h_{18} + h_{62}h_{28} + h_{63}h_{38} + h_{64}h_{48} + h_{65}h_{58} + h_{66}h_{68} + h_{67}h_{78} + h_{68}h_{88} + h_{69}h_{98}
\end{matrix}
$$

$$
\left.\begin{matrix}
h_{41}h_{19} + h_{42}h_{29} + h_{43}h_{39} + h_{44}h_{49} + h_{45}h_{59} + h_{46}h_{69} + h_{47}h_{79} + h_{48}h_{89} + h_{49}h_{99} \\
h_{51}h_{19} + h_{52}h_{29} + h_{53}h_{39} + h_{54}h_{49} + h_{55}h_{59} + h_{56}h_{69} + h_{57}h_{79} + h_{58}h_{89} + h_{59}h_{99} \\
h_{61}h_{19} + h_{62}h_{29} + h_{63}h_{39} + h_{64}h_{49} + h_{65}h_{59} + h_{66}h_{69} + h_{67}h_{79} + h_{68}h_{89} + h_{69}h_{99}
\end{matrix}\right) =
$$

$$
\begin{pmatrix}
h_{41}h_{17} + h_{42}h_{27} + h_{43}h_{37} & h_{41}h_{18} + h_{42}h_{28} + h_{43}h_{38} & h_{41}h_{19} + h_{42}h_{29} + h_{43}h_{39} \\
h_{51}h_{17} + h_{52}h_{27} + h_{53}h_{37} & h_{51}h_{18} + h_{52}h_{28} + h_{53}h_{38} & h_{51}h_{19} + h_{52}h_{29} + h_{53}h_{39} \\
h_{61}h_{17} + h_{62}h_{27} + h_{63}h_{37} & h_{61}h_{18} + h_{62}h_{28} + h_{63}h_{38} & h_{61}h_{19} + h_{62}h_{29} + h_{63}h_{39}
\end{pmatrix} +
$$

$$
\begin{pmatrix}
h_{44}h_{47} + h_{45}h_{57} + h_{46}h_{67} & h_{44}h_{48} + h_{45}h_{58} + h_{46}h_{68} & h_{44}h_{49} + h_{45}h_{59} + h_{46}h_{69} \\
h_{54}h_{47} + h_{55}h_{57} + h_{56}h_{67} & h_{54}h_{48} + h_{55}h_{58} + h_{56}h_{68} & h_{54}h_{49} + h_{55}h_{59} + h_{56}h_{69} \\
h_{64}h_{47} + h_{65}h_{57} + h_{66}h_{67} & h_{64}h_{48} + h_{65}h_{58} + h_{66}h_{68} & h_{64}h_{49} + h_{65}h_{59} + h_{66}h_{69}
\end{pmatrix} +
$$

$$
\begin{pmatrix}
h_{47}h_{77} + h_{48}h_{87} + h_{49}h_{97} & h_{47}h_{78} + h_{48}h_{88} + h_{49}h_{98} & h_{47}h_{79} + h_{48}h_{89} + h_{49}h_{99} \\
h_{57}h_{77} + h_{58}h_{87} + h_{59}h_{97} & h_{57}h_{78} + h_{58}h_{88} + h_{59}h_{98} & h_{57}h_{79} + h_{58}h_{89} + h_{59}h_{99} \\
h_{67}h_{77} + h_{68}h_{87} + h_{69}h_{97} & h_{67}h_{78} + h_{68}h_{88} + h_{69}h_{98} & h_{67}h_{79} + h_{68}h_{89} + h_{69}h_{99}
\end{pmatrix} =
$$

$$
\begin{pmatrix}
h_{41} & h_{42} & h_{43} \\
h_{51} & h_{52} & h_{53} \\
h_{61} & h_{62} & h_{63}
\end{pmatrix}
\begin{pmatrix}
h_{17} & h_{18} & h_{19} \\
h_{27} & h_{28} & h_{29} \\
h_{37} & h_{38} & h_{39}
\end{pmatrix} +
\begin{pmatrix}
h_{44} & h_{45} & h_{46} \\
h_{54} & h_{55} & h_{56} \\
h_{64} & h_{65} & h_{66}
\end{pmatrix}
\begin{pmatrix}
h_{47} & h_{48} & h_{49} \\
h_{57} & h_{58} & h_{59} \\
h_{67} & h_{68} & h_{69}
\end{pmatrix}
$$

$$
+ \begin{pmatrix}
h_{47} & h_{48} & h_{49} \\
h_{57} & h_{58} & h_{59} \\
h_{67} & h_{68} & h_{69}
\end{pmatrix}
\begin{pmatrix}
h_{77} & h_{78} & h_{79} \\
h_{87} & h_{88} & h_{89} \\
h_{97} & h_{98} & h_{99}
\end{pmatrix}
$$

Example B1.25

Use the stacking operator to show that the order of two linear operators can be interchanged as long as the multiplication of two circulant matrices is commutative.

Consider an operator with matrix H applied to vector $\mathbf{f}$ constructed from an image f, by using equation (1.68):

$$\tilde{\mathbf{f}} \equiv H\mathbf{f} = H \sum_{n=1}^{N} N_n f \mathbf{V_n} \tag{1.85}$$

The result is vector $\tilde{\mathbf{f}}$, to which we can apply matrix $\hat{H}$ that corresponds to another linear operator:

$$\tilde{\tilde{\mathbf{f}}} \equiv \hat{H}\tilde{\mathbf{f}} = \hat{H}H \sum_{n=1}^{N} N_n f \mathbf{V_n} \tag{1.86}$$

From this output vector $\tilde{\tilde{\mathbf{f}}}$ we can construct the output image $\tilde{\tilde{f}}$ by applying equation (1.69):

$$\tilde{\tilde{f}} = \sum_{m=1}^{N} N_m^T \tilde{\tilde{\mathbf{f}}} \mathbf{V_m}^T \tag{1.87}$$

We may replace $\tilde{\tilde{\mathbf{f}}}$ from (1.86) to obtain:

$$\begin{aligned} \tilde{\tilde{f}} &= \sum_{m=1}^{N} N_m^T \hat{H} H \sum_{n=1}^{N} N_n f \mathbf{V_n} \mathbf{V_m}^T \\ &= \sum_{m=1}^{N} \sum_{n=1}^{N} (N_m^T \hat{H})(H N_n) f (\mathbf{V_n} \mathbf{V_m}^T) \end{aligned} \tag{1.88}$$

The various factors in (1.88) have been grouped together to facilitate interpretation.

Factor $(N_m^T \hat{H})$ extracts from matrix $\hat{H}$ the m^{th} row of its partitions (see example 1.22), while factor (HN_n) extracts from matrix H the n^{th} column of its partitions (see example 1.23). The product of these two factors extracts the (m, n) partition of matrix $\hat{H}H$, which is a submatrix of size $N \times N$. This submatrix is equal to the sum of the products of the corresponding partitions of matrices $(N_m^T \hat{H})$ and (HN_n) (see example 1.24). Since these products are products of circulant matrices (see example 1.11), the order by which they are multiplied does not matter.

So, we shall get the same answer here either we have $(N_m^T \hat{H})(HN_n)$ or $(N_m^T H)(\hat{H}N_n)$, ie we shall get the same answer whichever way we apply the two linear operators.

What is the implication of the separability assumption on the structure of matrix H?

According to the separability assumption, we can replace $h(x,\alpha,y,\beta)$ with the product of two functions, $h_c(x,\alpha)h_r(y,\beta)$. Then inside each partition of H in equation (1.39), $h_c(x,\alpha)$ remains constant and we may write for H:

$$\begin{pmatrix} h_{r11}\begin{pmatrix} h_{c11} & \dots & h_{cN1} \\ h_{c12} & \dots & h_{cN2} \\ \vdots & & \vdots \\ h_{c1N} & \dots & h_{cNN} \end{pmatrix} & \dots & h_{rN1}\begin{pmatrix} h_{c11} & \dots & h_{cN1} \\ h_{c12} & \dots & h_{cN2} \\ \vdots & & \vdots \\ h_{c1N} & \dots & h_{cNN} \end{pmatrix} \\ h_{r12}\begin{pmatrix} h_{c11} & \dots & h_{cN1} \\ h_{c12} & \dots & h_{cN2} \\ \vdots & & \vdots \\ h_{c1N} & \dots & h_{cNN} \end{pmatrix} & \dots & h_{rN2}\begin{pmatrix} h_{c11} & \dots & h_{cN1} \\ h_{c12} & \dots & h_{cN2} \\ \vdots & & \vdots \\ h_{c1N} & \dots & h_{cNN} \end{pmatrix} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ h_{r1N}\begin{pmatrix} h_{c11} & \dots & h_{cN1} \\ h_{c12} & \dots & h_{cN2} \\ \vdots & & \vdots \\ h_{c1N} & \dots & h_{cNN} \end{pmatrix} & \dots & h_{rNN}\begin{pmatrix} h_{c11} & \dots & h_{cN1} \\ h_{c12} & \dots & h_{cN2} \\ \vdots & & \vdots \\ h_{c1N} & \dots & h_{cNN} \end{pmatrix} \end{pmatrix} \tag{1.89}$$

Here the arguments of functions $h_c(x,\alpha)$ and $h_r(y,\beta)$ have been written as indices to save space. We say then that matrix H is the **Kronecker product** of matrices h_r^T and h_c^T and we write this as:

$$H = h_r^T \otimes h_c^T \tag{1.90}$$

Example 1.26

Calculate the Kronecker product $A \otimes B$ where

$$A \equiv \begin{pmatrix} 1 & 2 & 3 \\ 4 & 3 & 1 \\ 2 & 4 & 1 \end{pmatrix} \quad B \equiv \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} \tag{1.91}$$

$$A \otimes B = \begin{pmatrix} 1\times B & 2\times B & 3\times B \\ 4\times B & 3\times B & 1\times B \\ 2\times B & 4\times B & 1\times B \end{pmatrix} =$$

$$\begin{pmatrix} 1\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} & 2\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} & 3\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} \\ 4\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} & 3\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} & 1\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} \\ 2\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} & 4\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} & 1\times \begin{pmatrix} 2 & 0 & 1 \\ 0 & 1 & 3 \\ 2 & 1 & 0 \end{pmatrix} \end{pmatrix} =$$

$$\begin{pmatrix} 2 & 0 & 1 & 4 & 0 & 2 & 6 & 0 & 3 \\ 0 & 1 & 3 & 0 & 2 & 6 & 0 & 3 & 9 \\ 2 & 1 & 0 & 4 & 2 & 0 & 6 & 3 & 0 \\ 8 & 0 & 4 & 6 & 0 & 3 & 2 & 0 & 1 \\ 0 & 4 & 12 & 0 & 3 & 9 & 0 & 1 & 3 \\ 8 & 4 & 0 & 6 & 3 & 0 & 2 & 1 & 0 \\ 4 & 0 & 2 & 8 & 0 & 4 & 2 & 0 & 1 \\ 0 & 2 & 6 & 0 & 4 & 12 & 0 & 1 & 3 \\ 4 & 2 & 0 & 8 & 4 & 0 & 2 & 1 & 0 \end{pmatrix} \tag{1.92}$$

How can a separable transform be written in matrix form?

Consider again equation (1.19) which expresses the separable linear transform of an image:

$$g(\alpha, \beta) = \sum_{x=1}^{N} h_c(x, \alpha) \sum_{y=1}^{N} f(x, y) h_r(y, \beta) \tag{1.93}$$

Notice that factor $\sum_{y=1}^{N} f(x, y) h_r(y, \beta)$ actually represents the product fh_r of two $N \times N$ matrices, which must be another matrix $s \equiv fh_r$ of the same size. Let us define an element of s as:

$$s(x, \beta) \equiv \sum_{y=1}^{N} f(x, y) h_r(y, \beta) = fh_r \tag{1.94}$$

Then (1.93) may be written as:

$$g(\alpha, \beta) = \sum_{x=1}^{N} h_c(x, \alpha) s(x, \beta) \tag{1.95}$$

Thus, in matrix form:

$$g = h_c^T s = h_c^T f h_r \tag{1.96}$$

What is the meaning of the separability assumption?

Let us assume that operator $\mathcal{O}$ has point spread function $h(x, \alpha, y, \beta)$, which is separable. The separability assumption implies that operator $\mathcal{O}$ operates on the rows of the image matrix f independently from the way it operates on its columns. These independent operations are expressed by the two matrices h_r and h_c, respectively. That is why we chose subscripts r and c to denote these matrices (r = rows, c = columns). Matrix h_r is used to multiply the image from the right. Ordinary matrix multiplication then means that the *rows* of the image are multiplied with it. Matrix h_c is used to multiply the image from the left. Thus, the *columns* of the image are multiplied with it.

Example B1.27

Are the operators which correspond to matrices H, $\hat{H}$ and $\tilde{H}$ given by equations (1.43), (1.45) and (1.47), respectively, separable?

If an operator is separable, we must be able to write its H matrix as the Kronecker product of two matrices. In other words, we must check whether from every submatrix of H we can get out a common factor, such that the submatrix that remains is the same for all partitions. We can see that this is possible for matrix $\hat{H}$, but it is not possible for matrices H and $\tilde{H}$. For example, some of the partitions of matrices H and $\tilde{H}$ are diagonal, while others are not. It is impossible to express them then as the product of a scalar with the same 3×3 matrix. On the other hand, all partitions of $\hat{H}$ are diagonal (or zero), so we can see that the common factors we can get out from each partition form matrix

$$A \equiv \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{pmatrix} \tag{1.97}$$

while the common matrix that is multiplied in each partition by these coefficients is the 3×3 identity matrix:

$$I \equiv \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.98}$$

So, we may write:

$$\hat{H} = A \otimes I \tag{1.99}$$

Box 1.4. The formal derivation of the separable matrix equation

We can use equations (1.68) and (1.69) with (1.38) as follows. First, express the output image g using (1.69) in terms of $\mathbf{g}$:

$$g = \sum_{m=1}^{N} N_m^T \mathbf{g} \mathbf{V}_\mathbf{m}^T \tag{1.100}$$

Then express $\mathbf{g}$ in terms of H and $\mathbf{f}$ from (1.38) and replace $\mathbf{f}$ in terms of f using (1.68):

$$\mathbf{g} = H \sum_{n=1}^{N} N_n f \mathbf{V}_\mathbf{n} \tag{1.101}$$

Substitute (1.101) into (1.100) and group factors with the help of brackets to obtain:

$$g = \sum_{m=1}^{N} \sum_{n=1}^{N} (N_m^T H N_n) f (\mathbf{V}_\mathbf{n} \mathbf{V}_\mathbf{m}^T) \tag{1.102}$$

H is a $N^2 \times N^2$ matrix. We may think of it as partitioned in $N \times N$ submatrices stacked together. Then it can be shown that $N_m^T H N_n$ is the H_{mn} such submatrix (see example 1.28).
Under the separability assumption, matrix H is the Kronecker product of matrices h_c and h_r:

$$H = h_c^T \otimes h_r^T \tag{1.103}$$

Then partition H_{mn} is essentially $h_r^T(m,n) h_c^T$. If we substitute this in (1.102), we obtain:

$$\begin{aligned} g &= \sum_{m=1}^{N} \sum_{n=1}^{N} \underbrace{h_r^T(m,n)}_{\text{a scalar}} h_c^T f (\mathbf{V}_\mathbf{n} \mathbf{V}_\mathbf{m}^T) \\ \Rightarrow g &= h_c^T f \sum_{m=1}^{N} \sum_{n=1}^{N} h_r^T(m,n) \mathbf{V}_\mathbf{n} \mathbf{V}_\mathbf{m}^T \end{aligned} \tag{1.104}$$

Product $\mathbf{V}_\mathbf{n} \mathbf{V}_\mathbf{m}^T$ is the product between an $N \times 1$ matrix with the only nonzero element at position n, with a $1 \times N$ matrix, with the only nonzero element at position m. So, it is an $N \times N$ square matrix with the only nonzero element at position (n,m). When multiplied with $h_r^T(m,n)$, it places the (m,n) element of the h_r^T matrix in position (n,m) and sets to zero all other elements. The sum over all m's and n's is matrix h_r. So, from (1.104) we have:

$$g = h_c^T f h_r \tag{1.105}$$

Example 1.28

You are given a 9×9 matrix H which is partitioned into nine 3×3 submatrices. Show that $N_2^T H N_3$, where N_2 and N_3 are matrices of the stacking operator, is partition H_{23} of matrix H.

$$H \equiv \begin{pmatrix} h_{11} & h_{12} & h_{13} & | & h_{14} & h_{15} & h_{16} & | & h_{17} & h_{18} & h_{19} \\ h_{21} & h_{22} & h_{23} & | & h_{24} & h_{25} & h_{26} & | & h_{27} & h_{28} & h_{29} \\ h_{31} & h_{32} & h_{33} & | & h_{34} & h_{35} & h_{36} & | & h_{37} & h_{38} & h_{39} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{41} & h_{42} & h_{43} & | & h_{44} & h_{45} & h_{46} & | & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & | & h_{54} & h_{55} & h_{56} & | & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & | & h_{64} & h_{65} & h_{66} & | & h_{67} & h_{68} & h_{69} \\ - & - & - & - & - & - & - & - & - & - & - \\ h_{71} & h_{72} & h_{73} & | & h_{74} & h_{75} & h_{76} & | & h_{77} & h_{78} & h_{79} \\ h_{81} & h_{82} & h_{83} & | & h_{84} & h_{85} & h_{86} & | & h_{87} & h_{88} & h_{89} \\ h_{91} & h_{92} & h_{93} & | & h_{94} & h_{95} & h_{96} & | & h_{97} & h_{98} & h_{99} \end{pmatrix} \quad (1.106)$$

$$\begin{aligned} N_2^T H N_3 &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \\ & \quad \begin{pmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & h_{18} & h_{19} \\ h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & h_{28} & h_{29} \\ h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & h_{38} & h_{39} \\ h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} & h_{49} \\ h_{51} & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} & h_{57} & h_{58} & h_{59} \\ h_{61} & h_{62} & h_{63} & h_{64} & h_{65} & h_{66} & h_{67} & h_{68} & h_{69} \\ h_{71} & h_{72} & h_{73} & h_{74} & h_{75} & h_{76} & h_{77} & h_{78} & h_{79} \\ h_{81} & h_{82} & h_{83} & h_{84} & h_{85} & h_{86} & h_{87} & h_{88} & h_{89} \\ h_{91} & h_{92} & h_{93} & h_{94} & h_{95} & h_{96} & h_{97} & h_{98} & h_{99} \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} h_{17} & h_{18} & h_{19} \\ h_{27} & h_{28} & h_{29} \\ h_{37} & h_{38} & h_{39} \\ h_{47} & h_{48} & h_{49} \\ h_{57} & h_{58} & h_{59} \\ h_{67} & h_{68} & h_{69} \\ h_{77} & h_{78} & h_{79} \\ h_{87} & h_{88} & h_{89} \\ h_{97} & h_{98} & h_{99} \end{pmatrix} \\ &= \begin{pmatrix} h_{47} & h_{48} & h_{49} \\ h_{57} & h_{58} & h_{59} \\ h_{67} & h_{68} & h_{69} \end{pmatrix} = H_{23} \end{aligned} \quad (1.107)$$

What is the "take home" message of this chapter?

Under the assumption that the operator with which we manipulate an image is *linear* and *separable*, this operation may be expressed by an equation of the form

$$g = h_c^T f h_r \tag{1.108}$$

where f and g are the input and output images, respectively, and h_c and h_r are matrices expressing the point spread function of the operator.

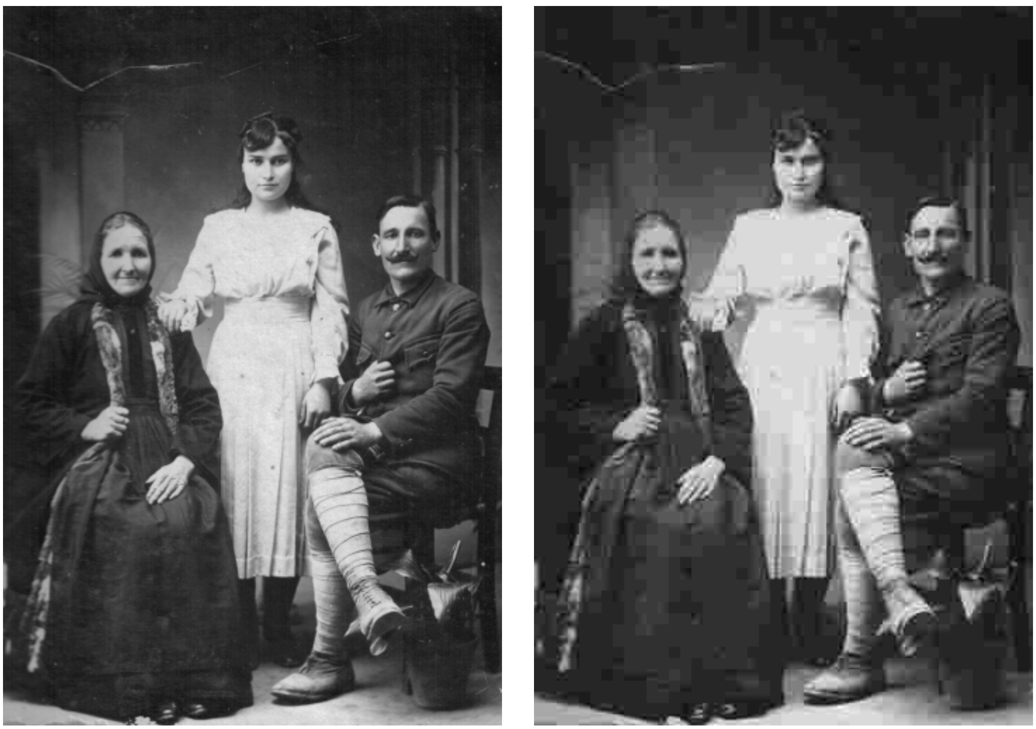


Figure 1.7: The original image "Ancestors" and a compressed version of it.

What is the significance of equation (1.108) in linear image processing?

In linear image processing we are trying to solve the following four problems in relation to equation (1.108).

- Given an image f, *choose* matrices h_c and h_r so that the output image g is "better" than f, according to some subjective criteria. This is the problem of **image enhancement**. Linear methods are not very successful here. Most of image enhancement is done with the help of nonlinear methods.

- Given an image f, *choose* matrices h_c and h_r so that g can be represented by fewer bits than f, without much loss of detail. This is the problem of **image compression**. Quite a few image compression methods rely on such an approach.
- Given an image g *and* an estimate of $h(x, \alpha, y, \beta)$, recover image f. This is the problem of **image restoration**. A lot of commonly used approaches to image restoration follow this path.
- Given an image f, *choose* matrices h_c and h_r so that output image g salienates certain features of f. This is the problem of **feature extraction**. Algorithms that attempt to do that often include a linear step that can be expressed by equation (1.108), but most of the times they also include nonlinear components.

Figures 1.7–1.11 show examples of these processes.

Figure 1.8: The original image "Birthday" and its enhanced version.

What is this book about?

This book is about introducing the mathematical foundations of image processing in the context of specific applications in the four main themes of image processing as identified above. The themes of image enhancement, image restoration and feature extraction will be discussed in detail. The theme of image compression is only touched upon as this could be the topic of a whole book on its own. This book puts emphasis on linear methods, but several nonlinear techniques relevant to image enhancement, image restoration and feature extraction will also be presented.

Figure 1.9: The blurred original image “Hara” and its restored version.

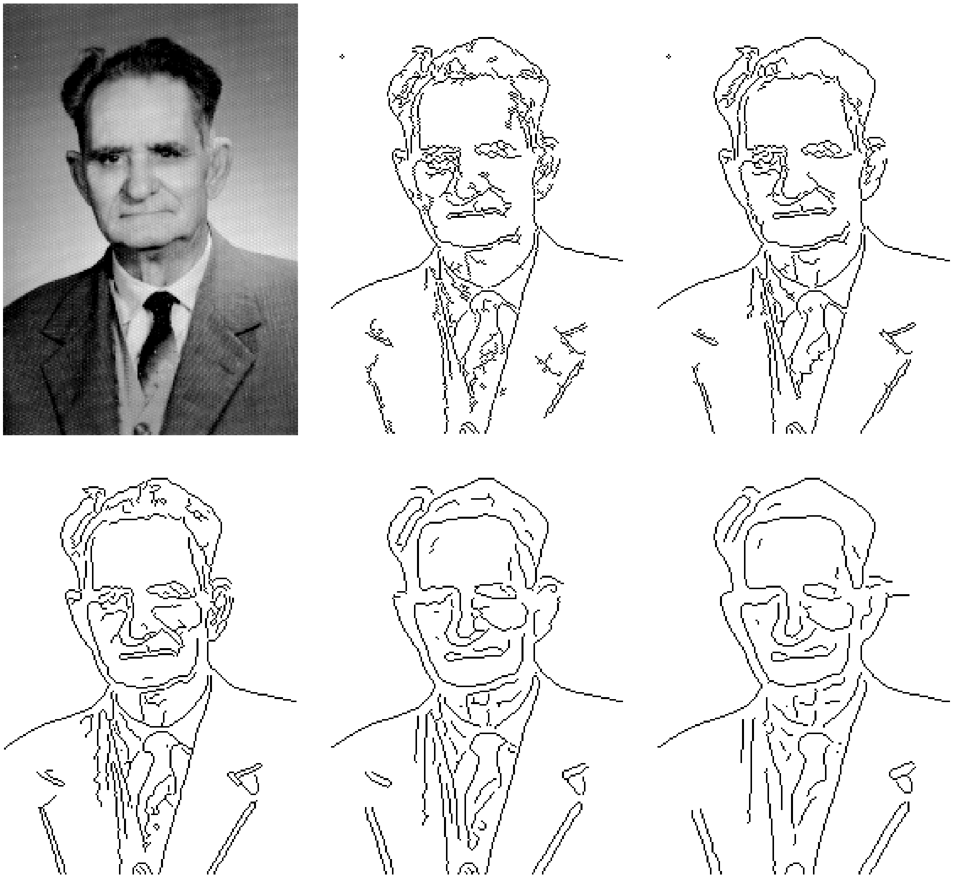


Figure 1.10: The original image “Mitsos” and its edge maps of decreasing detail (indicating locations where the brightness of the image changes abruptly).

(a) Image "Siblings" (b) Thresholding and binarisation

(c) Gradient magnitude (d) Region segmentation

Figure 1.11: There are various ways to reduce the information content of an image and salienate aspects of interest for further analysis.