

Chapter 2

DISPLAY

It is normal in any area of science that when a new specimen is acquired, it is examined carefully before any measurements are taken. The same approach is applied to digital images. If lots of similar images are to be dealt with, we may not wish to spend time looking at all of them. We may in fact be wanting to use digital image analysis in order to avoid this. But we will still wish to spend some time looking at images while developing a method to deal with them in large numbers.

One might ask why we don't simply look at the object which was the source of the digital image. There are a number of reasons for this:

- It may not be possible. The original object may not be available for examination, and the digital image may be all we have. For example, satellite imagery exists in digital form only, and the same is true of many types of non-invasive medical imaging. If we want to look at the object, we must look at the digital image.
- Even if the original object is available for viewing, we ought still to look at the digital version with which we will be working. This will allow us to see what level of detail is retained, how much blurring is present, whether there are problems in the image capture mechanism, what difficulties there are likely to be in producing an automatic analysis of the image etc.
- It may be possible by image enhancement to see aspects of the image which are difficult to spot in its original form.

This chapter considers the issues involved in producing a display from the various types of image that we may be working with. The display of binary, greyscale and multivariate images are considered in §2.1, §2.2 and §2.3 respectively. This chapter will also consider enlarging and reducing an image for convenience of examination (§2.4) and manipulating two or more images of the same scene or object so that features are at the same pixel position in all of them (§2.5). The main points of the chapter are summarised in §2.6. Since the display will use some form of computer equipment, we will need to make some references to possible limitations in

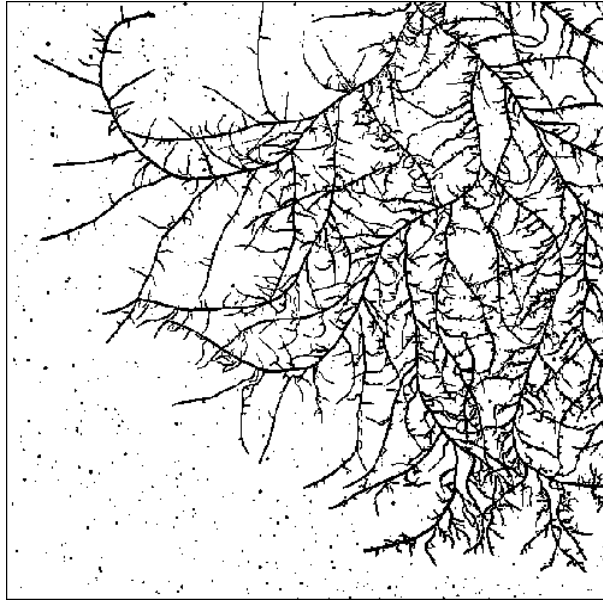


Figure 2.1: Binary display of thresholded fungal image, using a threshold of 150.

computer hardware. (For example, one obviously cannot produce any form of colour display on a monochrome monitor!) A consideration of what is wanted from the display component of an image analysis system should be considered when choosing one. Printing an image on paper should be considered as a form of display. General guidelines on matters of hardware and software are to be found in the Appendix.

2.1 Binary display

A binary image is the most straightforward type to display. Pixels take only values of 0 and 1. The display simply needs to distinguish between these two levels. A natural way to do this is to show one value as black and the other as white.

Any computer monitor capable of displaying images can do this. If the digital image is greyscale, rather than binary, a threshold must be chosen if a binary display is required: pixel values on one side of the threshold are displayed as black, those on the other side as white. Fig 2.1 shows a thresholded display of the fungal image. (Choosing a threshold is addressed in §4.1). We see that almost all the detail in the original (Fig 1.9(d)) is retained in this binary display.

Which pixel value should be shown as black and which white? Sometimes one choice will seem more natural. After thresholding a greyscale image, it is natural to show the pixel values which are below the threshold as black, and those above as white. However, one easily adapts to a form of display which at first seems unnatural. Astronomers generally look at negative photographic images of the night sky, in which stars are black on a white background.

2.2 Greyscale display

Most common digital images — certainly most in this book — are greyscale. The natural way to display such images is to use the pixel values to specify the brightness with which a pixel is illuminated on a computer screen, or how bright the pixel appears on a printed page. As described in §1.2, the pixel values are some measured physical property of the object being studied. If this property is the amount of reflected or transmitted light, then the display we produce will simply look like a black and white photograph of the object. We shall refer to such a display as **greyscale**, sometimes termed **monochrome**, rather than the more colloquial “black and white”, to distinguish it from binary images where each pixel is purely black or white.

Often, the pixel values are some other physical property of the image, and showing larger (or if preferred, smaller) pixel values as brighter is simply a device to enable us to see the spatial structure in what has been measured. This is not new in digital image analysis. It is standard practice, for example, to use brightness in displaying X-ray images, where brightness indicates how opaque a region of the object is to X-rays.

In using brightness in an image display, we have control over what level of brightness we assign to each pixel value. We can vary the contrast and overall brightness in ways which help us to see what is of interest in the image.

2.2.1 Perception of brightness

We have referred to a multilevel univariate image as a greyscale image because we use levels of greyness to display its values. The term ‘grey’ implies in effect that there is no colour present: all parts of the visible spectrum are equally represented. (Colour is discussed in §2.3) A physical measurement of the amount of light is its intensity. This is proportional to the energy in the light, and is the square of the amplitude of the light waves.

The human eye is not simply an instrument which registers the amount of light it receives. We tend to see zero intensity as black, low intensities as dark grey, with the greyness lightening as the intensities increase. The lightest objects we can see at any time tend to be perceived as white. However, the eye is very sensitive to the immediate surroundings of an object. This is best demonstrated by the familiar optical illusion shown in Fig 1.4(b). The main lesson to be derived from this is that the eye is not to be trusted for objective assessment of the absolute intensity of different parts of an image display.

Generally, there are a finite number of grey levels to which each pixel on a computer monitor can be set. On old monitors, just two levels were possible, meaning that only binary images could be displayed. On most modern monitors, at least 256 grey levels are possible. (Note that monitors are themselves analogue rather than digital, but appear digital because of the digital electronics which drive them.) If the number of display levels on a monitor is less than the number of distinct pixel values in the image, then some detail will be lost, as different pixel values will have to be shown as the same brightness on the monitor. However, 256 grey levels should be adequate for greyscale images.

It should not be assumed that the intensity response of a monitor is linear in the nominal values to which it is set. This is something which can be assessed by comparing, for example, an image in which all pixel values are half the maximum possible value with one constructed as a fine chessboard pattern of maximum values and zero. These two patterns should emit the same amount of light on average, and should be perceived similarly from a distance. Often, they are not.

2.2.2 Display enhancement

The most straightforward way to display a digital image is by a linear mapping of the range of pixel values onto the range of brightness intensities to which the monitor can be set. Sometimes the mapping is trivially simple. Often, both the pixel values and the possible display intensities will be integers in the range 0 to 255. If so, we simply display each pixel value as that display intensity.

A one-to-one display of the pixel values in an image may not be the best for seeing what is of interest. We may be able to see more by transforming the intensities displayed. This is analogous to the variable transformation (such as taking logs) that we might use with any other type of scientific data. A transformation would be appropriate if, for example, most of the pixel values were small, with a few large values. The differences between large and small values would swamp any subtle differences between small values. If the variation in the small values is of interest, then a transformation so that these small values occupied a wider range of display intensities would be of help.

A transformation may be effected in two distinct ways. One is to transform the pixel values stored in the computer. The other is to manipulate the way in which particular pixel values are displayed. An important concept in many image display systems is that of a **look-up-table** (LUT). This is a listing of the intensity on the monitor to be associated with each possible pixel value. Altering the LUT will have the same effect on the display as if the pixel values themselves were changed. It has the advantages of being much quicker in computer time (since every pixel in the image need not be transformed) and none of the detail in the image is lost, this being a possibility when some range of values is compressed. Of course, the possibility of transforming the original pixel values before subsequent analysis still remains. Use of LUTs is illustrated in Fig 2.2.

In the rest of this section, we shall assume, without loss of generality, that the LUT is being changed when a transformation is made. A number of transformations are commonly used.

Piecewise linear

The **piecewise linear** transformation function consists of linear segments. It is sometimes referred to as a **contrast stretch**. Often the linear segments will be of the special form illustrated in Fig 2.3. A selected range of pixel values is allocated the complete range of display intensities. Here they are applied to the X-ray image. The original image has pixel values

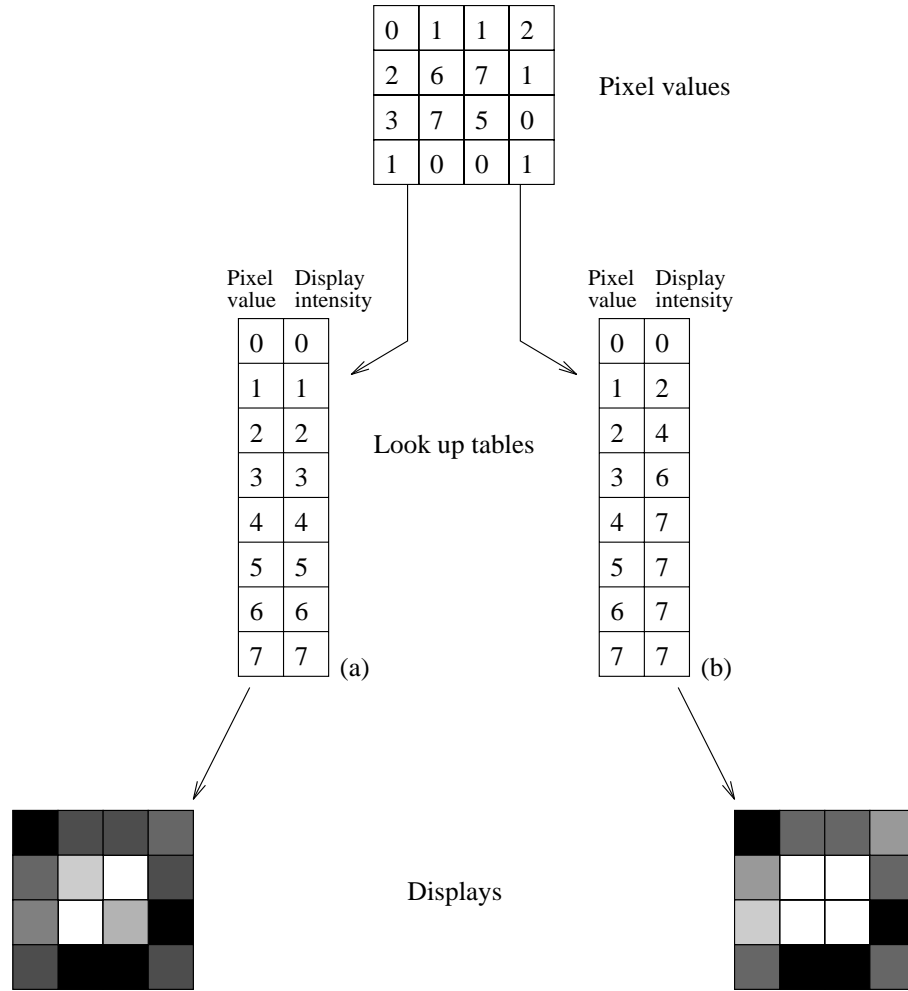


Figure 2.2: Use of look-up-tables for displaying pixel values. Both pixel values and display intensities are in the range 0-7. In (a) the LUT produces display intensities proportional to the pixel values, whereas in (b) they are modified to emphasise variation in small pixel values.

between -1000 and 834 . Fat tissue has values about -100 and muscle tissue about 30 , so these do not differ much in the original display. If the values between -250 and 260 are stretched to cover the whole range of display intensities, much greater detail can be seen in the internal tissues of the sheep. This transformed version of the image will hereafter be used in preference to the original.

The transformation has the effect that pixel values between two limits, say a and b , are allocated the full range of intensities available on the monitor. Anything above or below the range a to b is shown as full or zero intensity respectively. Assume 0 and I_{\max} are the minimum and maximum intensity available on the display. If f_{ij} is the pixel value at location (i, j) , the display intensity, I_{ij} , is given by

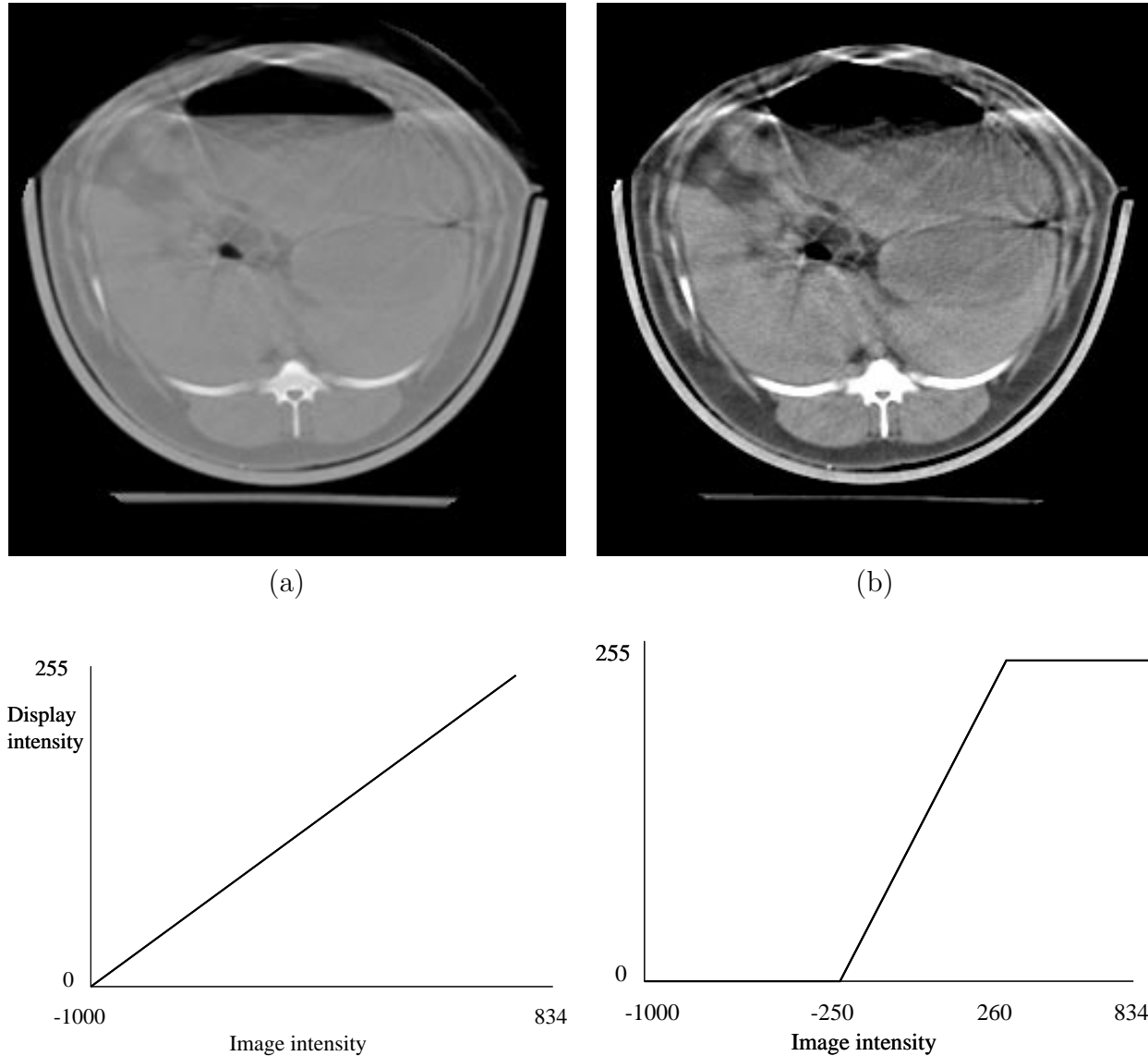


Figure 2.3: Piecewise linear intensity transform. **(a)** Original X-ray image. **(b)** Display chosen to emphasise detail in part of the range of image intensities. **(c)** and **(d)** show the LUT's (as plots rather than lists of intensities) used for (a) and (b).

$$I_{ij} = \begin{cases} 0 & \text{if } f_{ij} \leq a \\ I_{\max} \frac{f_{ij}-a}{b-a} & \text{if } a < f_{ij} < b \\ I_{\max} & \text{if } f_{ij} \geq b \end{cases}$$

Values for a and b may be chosen in a number of ways. These are illustrated in Fig 2.4 for band two of the Landsat image.

- We may set a to be the minimum of f over the image, and b to be the maximum. This ensures that detail in the extremes of the range of pixel values is not lost. This is

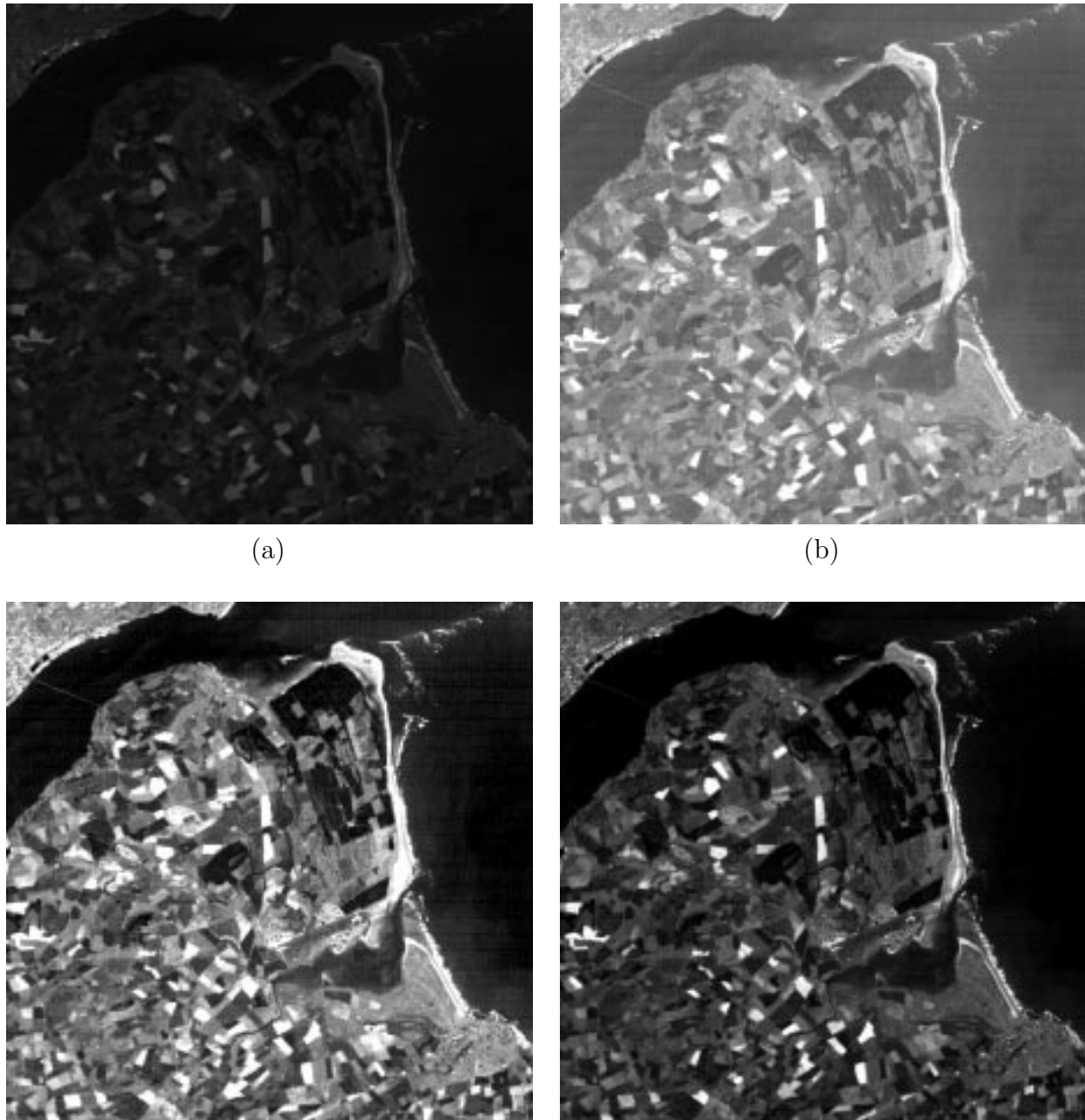


Figure 2.4: Piecewise linear contrast stretches applied to band 2 of the Landsat image. The values of a and b are chosen **(a)** as the minimum and maximum pixel values in the image, **(b)** as $\bar{f} \pm 2\frac{1}{2}\sigma$, **(c)** as the 5th and 95th percentiles, and **(d)** manually.

illustrated in Fig 2.4(a). The minimum pixel value is 21 and a few very large pixel values in the urban area near the top left give $b = 134$ and as a result most of the image remains dark.

- If we are prepared to lose some detail in the extremes, we could set

$$a = \bar{f} - c\sigma$$

$$b = \bar{f} + c\sigma$$

where \bar{f} is the mean pixel value in the image, σ is the standard deviation of the pixel values, i.e.

$$\sigma = \sqrt{\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (f_{ij} - \bar{f})^2},$$

and c is a multiplier such as 2 or $2\frac{1}{2}$. This will be useful if there are a few extreme pixel values, which would otherwise force the display intensities for the majority of pixels into a small range. This is illustrated in Fig 2.4(b), where we get $a = 15$ and $b = 45$, using a multiplier of $2\frac{1}{2}$. This transform shows improved contrast over most of the image.

- A closely related choice is to set a and b to the 5th and 95th percentiles (for example) of the distribution of pixel values. (The p th percentile is defined to be the number F for which $p\%$ of pixel values f_{ij} are less than F .) This is illustrated in Fig 2.4(c). Here $a = 24$ and $b = 41$ and even stronger contrast can be seen than in Fig 2.4(b), where $\bar{f} \pm 2\frac{1}{2}\sigma$ extended beyond the range of pixel values.
- a and b can be chosen manually. This was done in Fig 2.4(d). Here, $a = 25$ was chosen as a typical pixel value in the dark forest area in the top right of the image, and $b = 56$ as a typical value in the bright fields of oil-seed rape scattered throughout the image. This contrast stretch makes it apparent just how different these fields are from the rest of the landscape, a fact which was lost in Fig 2.4(c) since they represent less than 5% of the image.

Another type of piecewise linear transform is one which consists of simultaneously stretching several small ranges of pixel values to the full intensity range, so that a plot of the LUT looks like a series of steep ramps. This is useful for revealing small local variations in pixel values. Bright and dark areas in the transformed image will follow contours of pixel values.

Exponential, logarithmic

The **exponential** and **logarithmic** functions are examples of transformations which enhance the display of the upper and lower parts respectively of the range of pixel values (see Fig 2.5). The effect of the logarithmic transform on the SAR image is shown in Fig 2.6. In the original image there was greater speckle variability in the brighter parts of the image than in the darker. After the log transform the speckle variability should be equal in all parts of the image. This can be shown mathematically from the known properties of radar speckle (Skolnik, 1981; Oliver

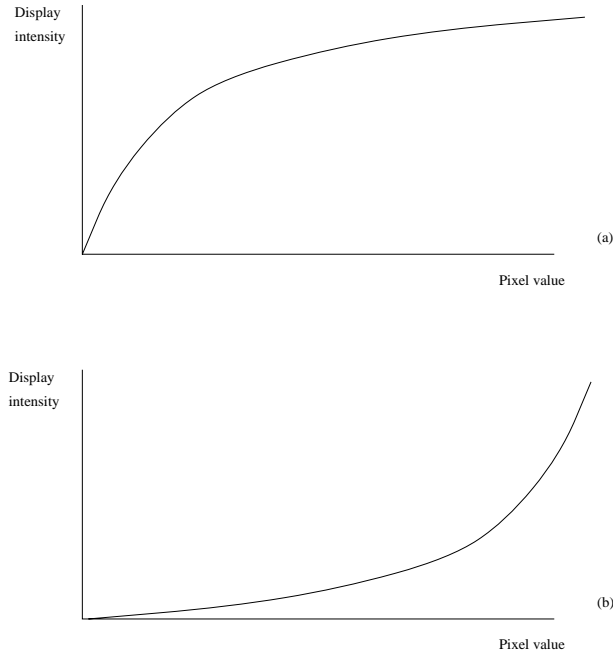


Figure 2.5: (a) Logarithmic and (b) Exponential intensity transforms.

1991). A piecewise linear transformation using the minimum and maximum pixel value was applied after the log transform. The transformed pixel values have been used in subsequent analyses in this book.

Histogram equalization

Histogram equalization is a transformation which ensures that all display intensities are approximately equally represented. The intention is that ranges of pixel values are allocated portions of the display intensity range according to the frequency with which they occur in the image. Figs 2.7(a) and (b) show the cumulative histograms of the intensities of the muscle fibres image before and after equalization, and (c) and (d) show the original and equalized images. We can see more detail in the cells in the equalized image. This is because the pixel values in the dark cells, for example, have been allocated a greater range of display intensities than in the original display. The brighter cells also have a greater range of intensities.

To see how a histogram equalization is achieved, let I denote the display intensity, where I can take the integer values $0, 1, \dots, I_{\max}$, and let $p(f)$ be the proportion of the pixel values $\leq f$. If pixel values f are displayed with intensity

$$I = I_{\max} p(f),$$

then the proportion of pixels with display intensity $\leq I$ will be I/I_{\max} , leading to a linear cumulative distribution of intensities, i.e. a uniform distribution. In practice, the discrete



Figure 2.6: Logarithmic transform of the SAR image.

(i.e. non-continuous) nature of the image histogram will mean that the transformed image will only have an approximate uniform distribution. For a more detailed discussion of histogram equalization, see Gonzalez and Wintz (1987, Ch.4).

2.3 Colour Display

Colour is an important aspect of human vision. It is so fundamental that it is impossible to describe its sensation in terms of anything more basic. We see the world in colour, and so colour is an aspect of many of the things which we look at in science. Some types of imaging systems using non light-based physical processes generate images without colour. Colour can be assigned to such images to help us see variability in the quantity measured. It is also a natural way to deal with images in which more than one variable has been measured at each pixel. To understand how colour can be used in display, it is necessary first to understand what colour is physically, and how it is perceived by the human visual system.

2.3.1 The physics and biology of colour

Light consists of electromagnetic waves. The human eye is sensitive only to waves with wavelengths within a certain range, which are termed visible light. Waves with other wavelengths are considered as different types of radiation depending on their wavelength. These include X-rays, ultra-violet and infra-red radiation, microwaves, and radio waves. Within the range of wavelengths we see as light (referred to as the **visible spectrum**), the actual wavelength determines the colour of the light, ranging from indigo/violet at the short end to red at the

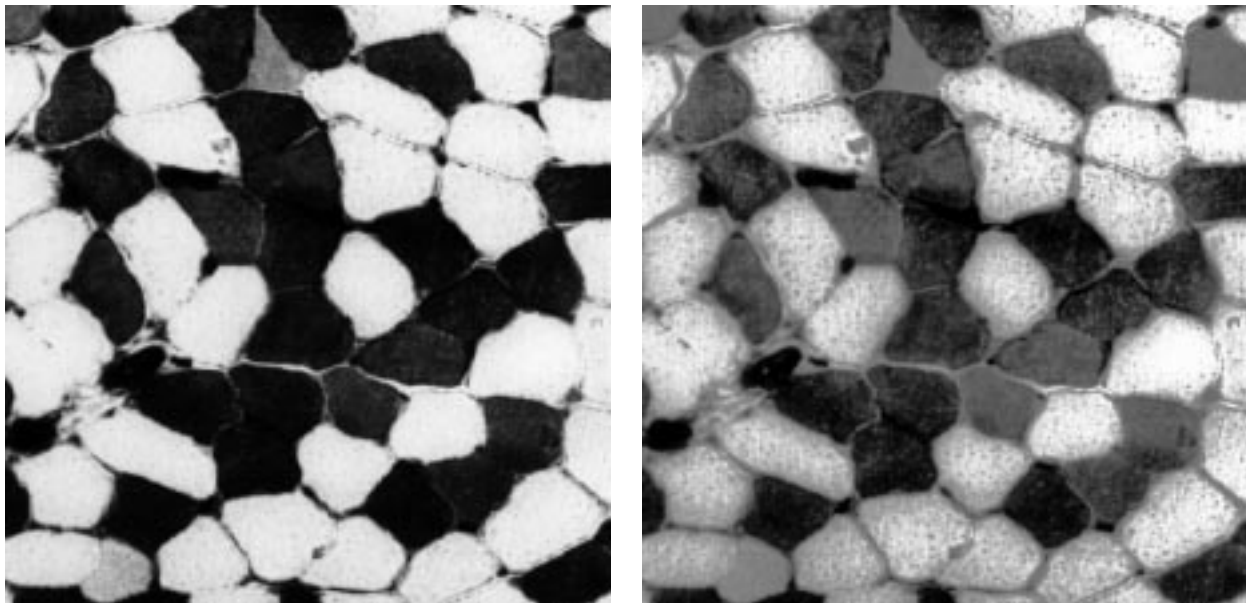
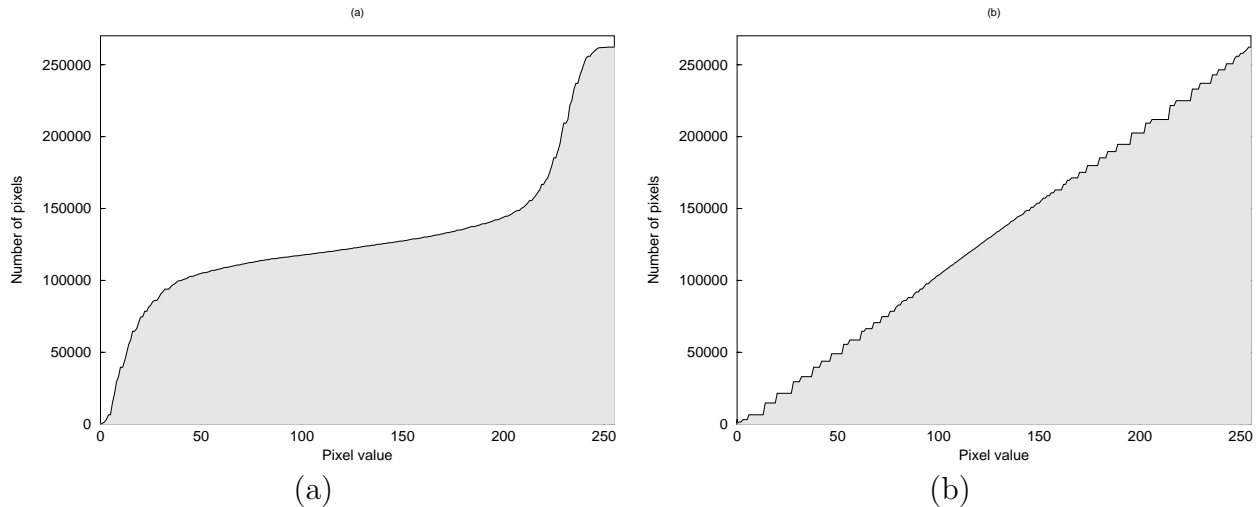


Figure 2.7: **(a)** Cumulative histogram of the muscle fibres image before and **(b)** after equalization. **(c)** Muscle fibre image before and **(d)** after equalization.

long end. Words are an imprecise and subjective way to specify the colour of a light source, and a physicist would simply quote the wavelength. Most light sources produce waves not with a single wavelength, but with a distribution of different wavelengths.

The biology of how our eyes see colour is different from the physical reality. It is this biology with which we are most concerned, since we are discussing the use of colour to display images for the human interpreter. The retina of the eye contains cells (referred to as cones) which differ in their sensitivity to light of particular wavelengths. There are three types of cones: one type has its maximum sensitivity to light in the blue part of the spectrum, another type to the green part and a final type to the red part. Light which consists of a single wavelength in the red part of the spectrum will be detected most strongly by the red-sensitive cells, and we see

it as red. Light of a single wavelength lying between the green and red parts of the spectrum produces a response in both the green and red-sensitive cells, and a sensation of yellow results. A similar sensation is produced by light which consists of a mixture of one red and one green wavelength — it looks yellow — since it produces the same effect on the colour-sensitive cones. Any combination of wavelengths produces a different colour sensation, according to how much of a response it produces in each of the three types of colour-sensitive cells. An equal response in each of the three basic colours, blue, green and red, produces a perception of grey.

Because this is the way colour is perceived, we can regard colour as a 3-dimensional quantity. To reproduce any colour, we specify the intensity of red light, green light and blue light, the so called primary colours. This is referred to as the **RGB** (Red, Green, Blue) system. We can think of the RGB components as specifying a point inside a cube, the sides of which correspond to the range of possible intensities of blue, green and red which can be specified. Most technological use of colour, from photography to television screens and computer monitors employs an RGB system to produce different colours. For example, a colour monitor has a screen which consists of many small, light-emitting dots of three types — blue, green and red. The brightness of each of the three types of dots determines what colour is seen in a particular part of the screen.

2.3.2 Pseudocolour

One use of colour in image display is to assign different colours to different pixel values in a greyscale image. This is termed **pseudocolour**, since it doesn't reflect actual colour variation in the scene imaged. Such a display sometimes has the advantage over a standard greyscale display that the eye is better able to see colour differences, and to compare colours in different parts of the display, than to see differences in grey levels.

There are many possibilities for assigning colours to pixel values. We can regard the sequence of colours we assign to the range of pixel values, from its minimum to its maximum, as defining a path in the RGB colour cube. Two possibilities for such paths are shown in Fig 2.8 (colour plate). Fig 2.9 (colour plate) shows the fish image (Fig 1.9(e)) displayed in pseudocolour. Many image features and comparisons can be more readily made in colour than in grey levels, since the problem of sensitivity to the surrounding area (as in Fig 1.4(b)) is much reduced. However, pseudocolour is less suitable for images which we are accustomed to seeing in greyscale or natural colour. A greyscale image of a human face looks unnatural when displayed in pseudocolour, and is more difficult to recognise.

2.3.3 Direct allocation

If we have two or more variables measured at each pixel in an image (i.e. a multivariate image), we may use them to control different RGB colour components in the display. This is particularly convenient if there are three variables. In this case, we simply assign each of them to one of the three RGB components. If the three variables are in fact measurements of the light in the red, green and blue components emitted or reflected by a real scene, then a natural looking colour

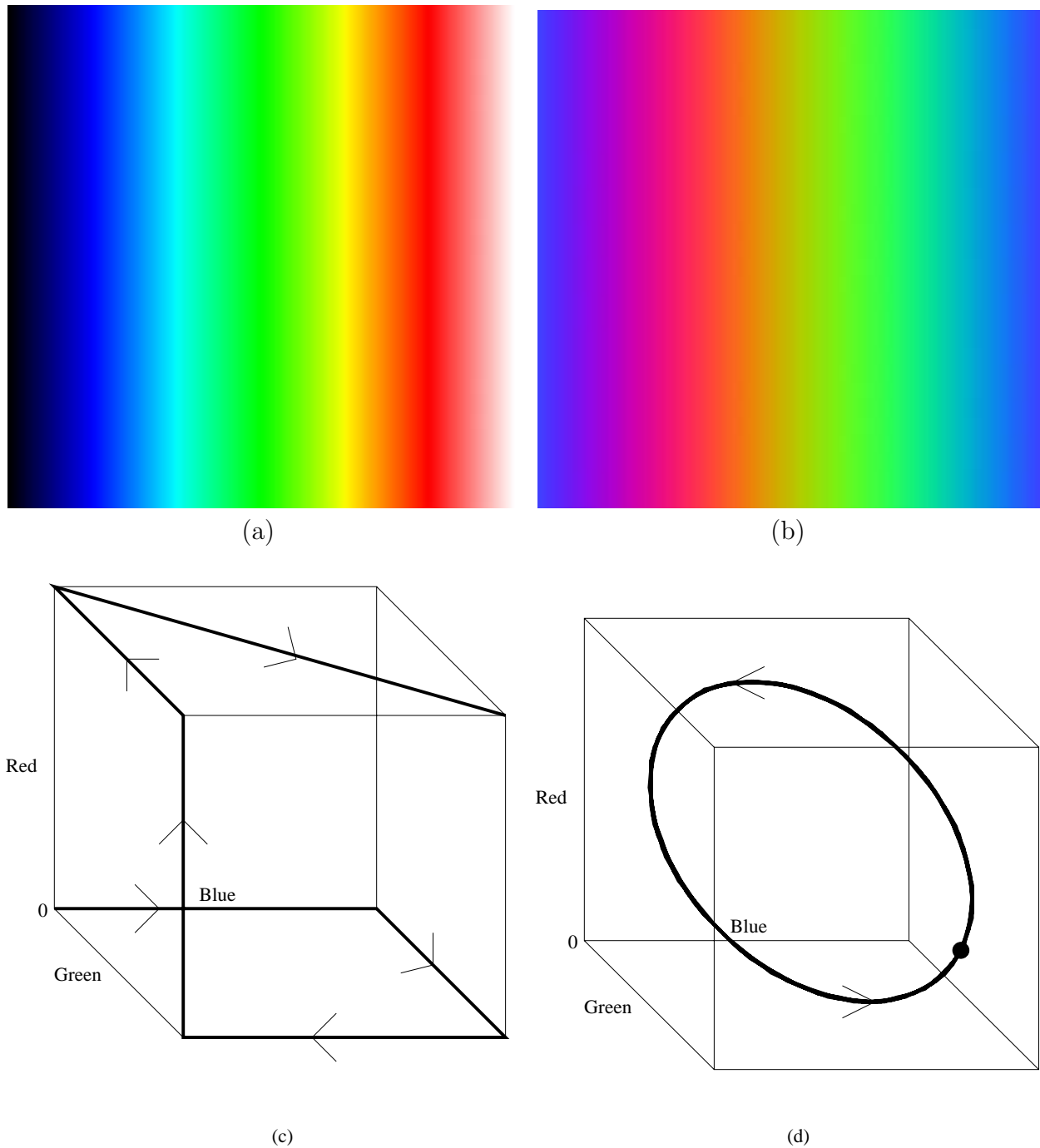


Figure 2.8: Two pseudocolour maps. (a) is suitable for general use and (b) is appropriate when the data are angles or directions, since large values are displayed similarly to small values. (c) and (d) show the maps as paths in the colour cube. In (d), the black dot indicates the start and finish points of the path.

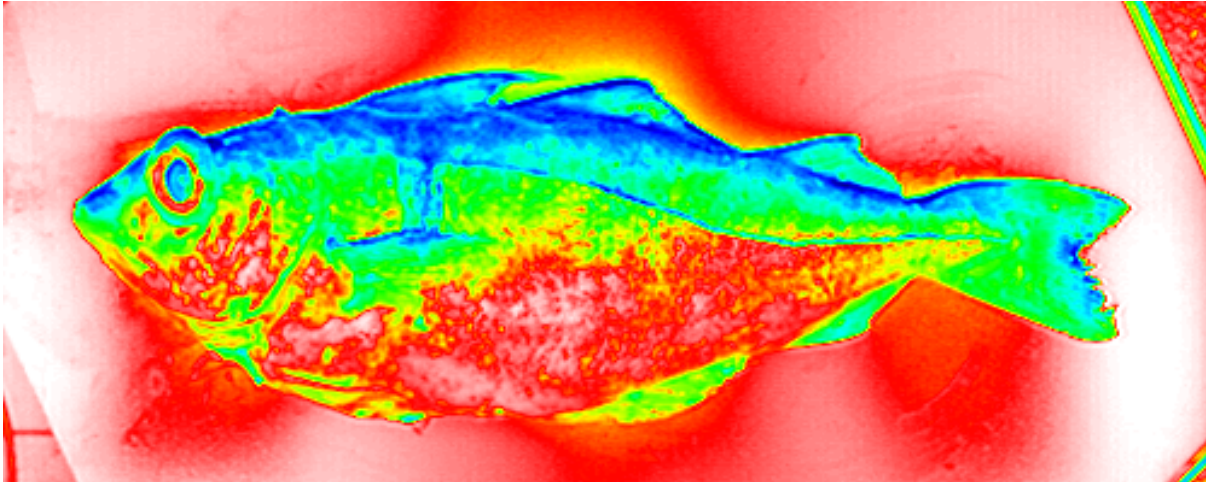


Figure 2.9: Pseudocolour display of the fish image, using the map in Fig 2.8(a).

picture of the scene is produced. This is illustrated in Fig 2.10 (colour plate) for the satellite image, where bands 1, 2 and 3 are the blue, green and red parts of the visible light spectrum respectively. The three bands have been individually stretched so that $\bar{f} \pm 2\frac{1}{2}\sigma$ occupies the full intensity range as described in §2.2. If the variables assigned to the RGB components do not represent blue, green and red light, then the display will not look like anything natural, but the eye will still be able to study its features.

If we have two image variables we could assign them to two of the colour components and set the third to a constant value. Another possibility is to let two of the components be determined by one of the variables, and the third by the other variable. This is shown in Fig 2.11 (colour plate) for the two MRI variables. If we have more than three image variables, there are a number of possibilities:

- We may feel that a selection of just three of them will produce a display in which most features of interest can be seen. Some experimentation may be needed to find this combination.
- We could create some derived functions of the variables, and use these to determine the colour components. One possibility here is to use the **principal components** (Krzanowski, 1988, p53). These are a sequence of linear combinations of the variables which contain the greatest amount of variability while being independent of all previous components in the sequence. We may then produce a display from the first three, or some other set of three components. This does not guarantee a display which has the most image features of interest. There is no certain way to find appropriate combinations of image variables for this purpose.
- The **maximum noise fraction** (MNF) technique (Green, Berman, Switzer and Craig, 1988) finds a sequence of linear combinations of variables, with noise levels decreasing through the sequence. Noise is assumed to be lowest when the correlation in pixel value



Figure 2.10: Direct colour display of bands 1, 2 and 3 of the Landsat image.

between neighbouring pixels is highest. Later terms in the sequence can be used to determine colour components.

- Some derived functions of the variables which are thought to reflect important features of the scene may be used. For example, in remote sensing, vegetation indices (Jensen, 1986, Ch.7) are considered to indicate amount or type of vegetation. When based on ratios of variables (which correspond to different parts of the electromagnetic spectrum) they can eliminate or reduce the effect of variations in illumination caused by topography.
- The statistical technique of projection pursuit (Jones and Sibson, 1987; Nason and Sibson, 1991) has been developed to attempt to find interesting projections of multidimensional data.

It can be seen that a great many colours can be obtained on a display unit from the combinations of three basic colour components. If each component can be displayed in 256 different intensities for example (which is often the case) then there are $256^3 = 16,777,216$ different possible colours which a pixel can have. However, the memory configuration of many computers is such that not all of these can be available at the same time. Many operate by allowing the display intensities to be one of a smaller number (often 256) of different colours. Which colour is assigned to

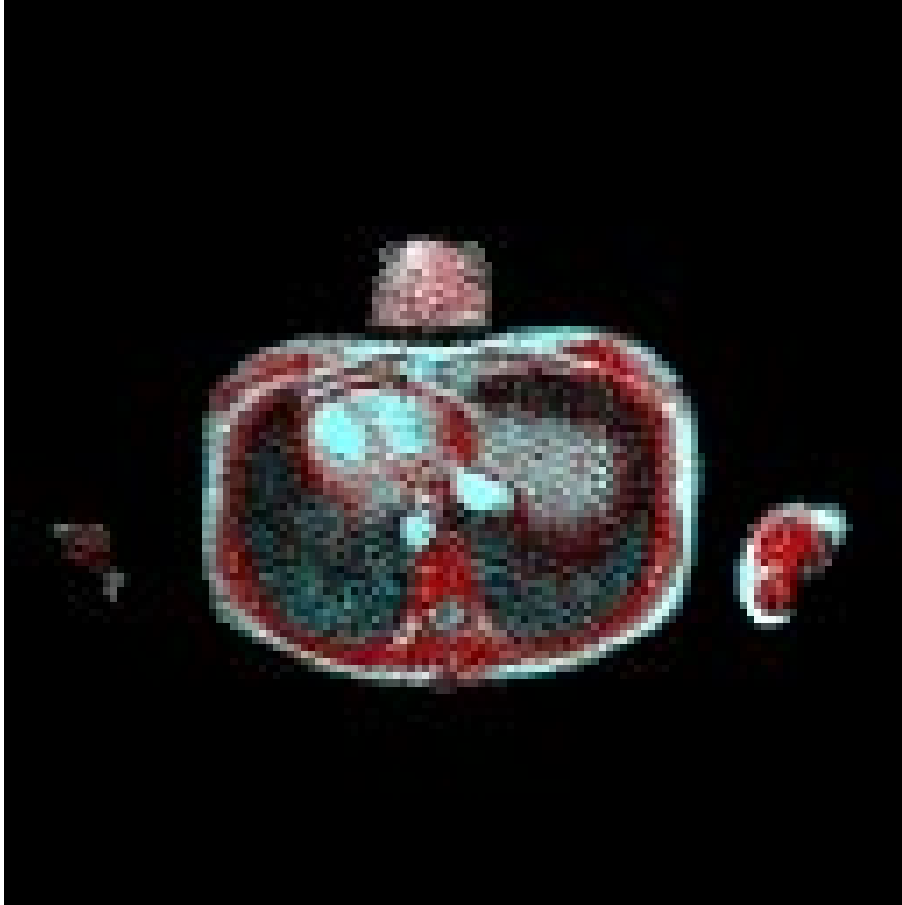


Figure 2.11: Colour display of the MRI image variables. The inversion recovery variable controls the blue and green components of the display, while the proton density variable controls the red.

each of the 256 possibilities is determined by a LUT. We must then choose which 256 colours out of the 16,777,216 possibilities to use. A straightforward way to do this is to select eight colours from their full range for two of the RGB components, and four from the other, making $8 \times 8 \times 4 = 256$ colour combinations. This is easily programmed since 4, 8 and 256 are powers of two. Other possibilities, sometimes implemented in software packages, are to select from the 16,777,216 possibilities according to the observed colour combinations in the image, to minimise the loss of detail when reducing to 256 colours (Heckbert, 1982).

Other colour coordinates

Colour can be specified using other coordinate systems in the colour cube. The most common alternative to the RGB coordinates is the **intensity, hue, saturation** system (**IHS**):

- *Intensity* or *lightness* is how bright the light is overall. It may be defined as the projection of a colour in RGB coordinates onto the diagonal from black to white.

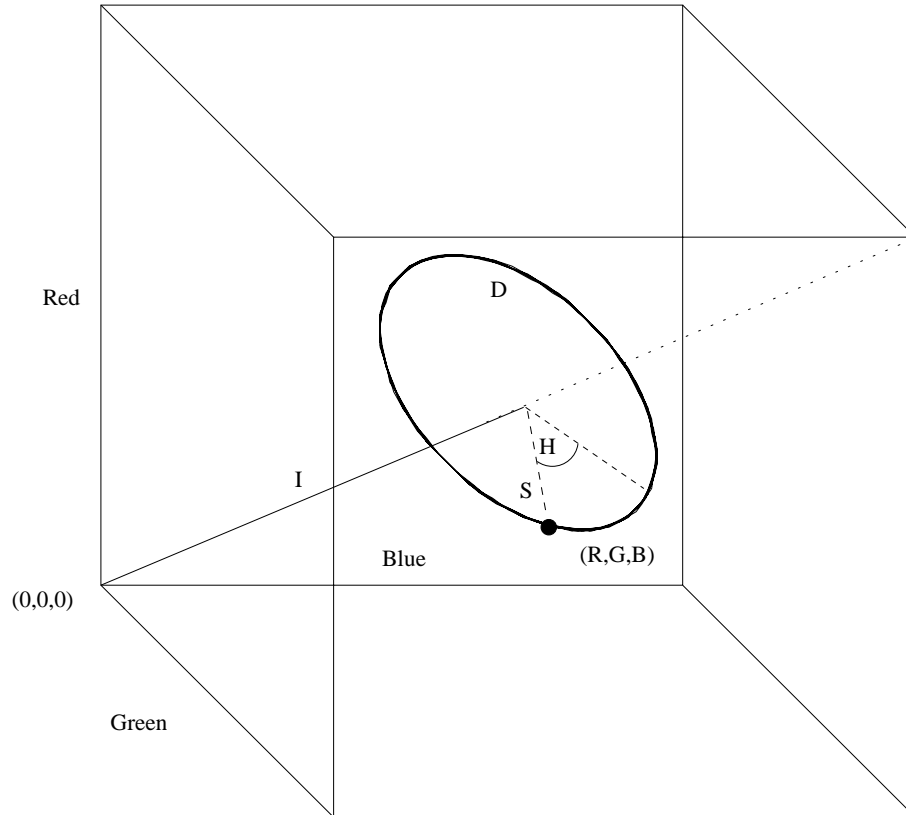


Figure 2.12: Illustration of the intensity, hue and saturation coordinates for describing colour. The point (R, G, B) lies on the disc D which is perpendicular to a diagonal line drawn from black $(0,0,0)$ to white $(1,1,1)$. The intensity I is the distance from $(0,0,0)$ to D . Hue, H , is the angle in D the point (R, G, B) makes with a line drawn towards blue $(0,0,1)$. Saturation, S , is the distance of the point (R, G, B) from the centre of D as a proportion of the distance along a line from the centre of D through (R, G, B) to the edge of the cube.

- *Hue* indicates angular position of the colour in the plane perpendicular to the black-to-white diagonal. The choice of zero angle is arbitrary.
- *Saturation* measures how far from the diagonal line the colour is, as a proportion of the distance from the diagonal through the colour to the edge of the cube.

This is illustrated in Fig 2.12. If we assign three of the variables in a multivariate image to defining intensity, hue and saturation, the above definition allows us to obtain the RGB components needed for the monitor. Alternatively, we use the definition to describe any colour specified by RGB coordinates in terms of IHS coordinates, which may have more intuitive appeal. Algorithms for performing these coordinate transformations may be found in Foley, Van Dam, Feiner, and Hughes (1991).

The IHS coordinates can be used in other ways than simply assigning three components of a multivariate image to intensity, hue and saturation. If we have two variables, we might assign one to intensity, one to hue and set saturation to its maximum. With one variable, we might

assign it to hue, set saturation to its maximum and intensity to some value between zero and its maximum. This leads to a form of pseudocolour (as shown in Figs 2.9(b) and (d)). This is illustrated in Fig 3.18 (colour plate; see §3.4.2). Note that we cannot use maximum intensity, as this leads to an all-white image regardless of hue and saturation. Hue is undefined when red, green and blue are equal.

Other colour coordinate systems also exist. Some of these have been defined to take account of the properties of monitors or of human perception of colour. A description of some of these may be found in Foley *et al.* (1991). If an image is being captured in colour, for example with a colour digital camera, it can be important to understand its colour response and to calibrate it with respect to some colour coordinate system. For an example, see Strachan, Nesvadba and Allen (1990b).

A final point to note here is that there are some differences between how images are displayed on monitors, and how they are produced on paper. A great many different devices for printing images exist, with varying quality of reproduction. The quality is often poorer than the display on a monitor.

2.4 Zooming and reduction

When an image has fine detail, we may not be able to examine this detail easily in a standard display on a monitor. We would like to enlarge a part of the image. This is known as **zooming**. We cannot usually enlarge the pixels on the monitor, and zooming must consist of allocating more than one pixel on the monitor for each pixel in the image.

The simplest form of zooming is **pixel replication**. For example, if we display an image pixel using a block of $k \times k$ pixels on the display, then we will apparently have increased the size of the image. (See Fig 2.13.) Clearly, no more detail is present, although it may be easier to see what is there. Of course, there may no longer be room on the display for the whole image, as in Fig 2.13. It is equivalent to holding a magnifying glass in front of the screen. It is not always necessary to replicate pixel values in the computer memory. Some computer hardware has the ability to replicate pixels on the screen by making use of the way that the display control electronics access the computer memory. Pixel replication is satisfactory if we wish to magnify the image by integer ratios. If we wish to magnify by fractional ratios, there is the possibility of replicating pixels by different amounts in some pattern. For example, if we replicate every second pixel only, and every second row, we will have magnified the image by 1.5. This gives an acceptable result if we are concerned only with the larger features of the image, but distortion of small scale details can be noticeable.

On other occasions, we may wish to **reduce** the size of an image. This may be for display purposes, because our original image has too many pixels to be accommodated on the monitor. Alternatively, we may wish to reduce the size of an image so that it can be processed in less computer time, or occupy less storage space. (Image compression is also considered in §A.1.4.) Reducing the image we intend to work with will involve some loss of detail. We may reduce the size of an image by integer ratios in two straightforward ways.



Figure 2.13: Zooming to enlarge part of the fungal image (Fig 1.9(d)). An enlargement of the top right of the original image is shown. Each pixel is repeated in a 5×5 block.

- We can form the reduced image by selecting a subset of pixels from the original image. For example, if we take every third pixel in every third row of the original image, we will have reduced the image size by a factor of nine. This is known as **pixel sampling**.
- We may define each pixel in the reduced image to be the average of a three by three block of pixels in the original image. This reduces the loss of information, unlike the equivalent pixel sampling which throws away 8/9 of the pixels in the original image. This is known as **block averaging**.

The difference between the two methods is illustrated in Fig 2.14. The effect of the two methods is demonstrated in Fig 2.15. If neighbouring pixel values are similar, there will be little difference in these two methods. Pixel sampling has one advantage in that it requires less computer time. For non-integer reductions, sampling and averaging procedures can be devised, with distortion problems similar to those found in zooming.

For both zooming and reduction, particularly with non-integer factors, **interpolation** provides a method which produces smoother results with fewer distortions than simple pixel replication or sampling. The basic idea is that we regard the digital image as a discrete version of some continuous variable, which we have sampled at points on a lattice. To enlarge or reduce the image, we sample at a greater or lesser frequency. The pixel locations in the new image will fall at different positions on the continuous surface. We do not of course know what the pixel values at these positions should be, but we may estimate them by assuming that there is a smooth change in pixel value between the four nearest pixels in the original image. This is illustrated in Fig 2.16 for the one dimensional case. If we assume that the inter-pixel distance in the original image is 1, and that we wish to estimate the value, f'_y at a pixel in the interpolated image, then we have

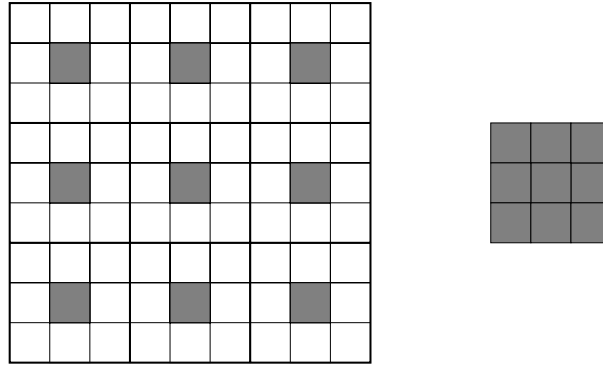


Figure 2.14: Image reduction. In reduction by pixel sampling, the shaded squares in the full image are used to form the reduced image. In reduction by block averaging, the average intensity in the blocks shown is used.

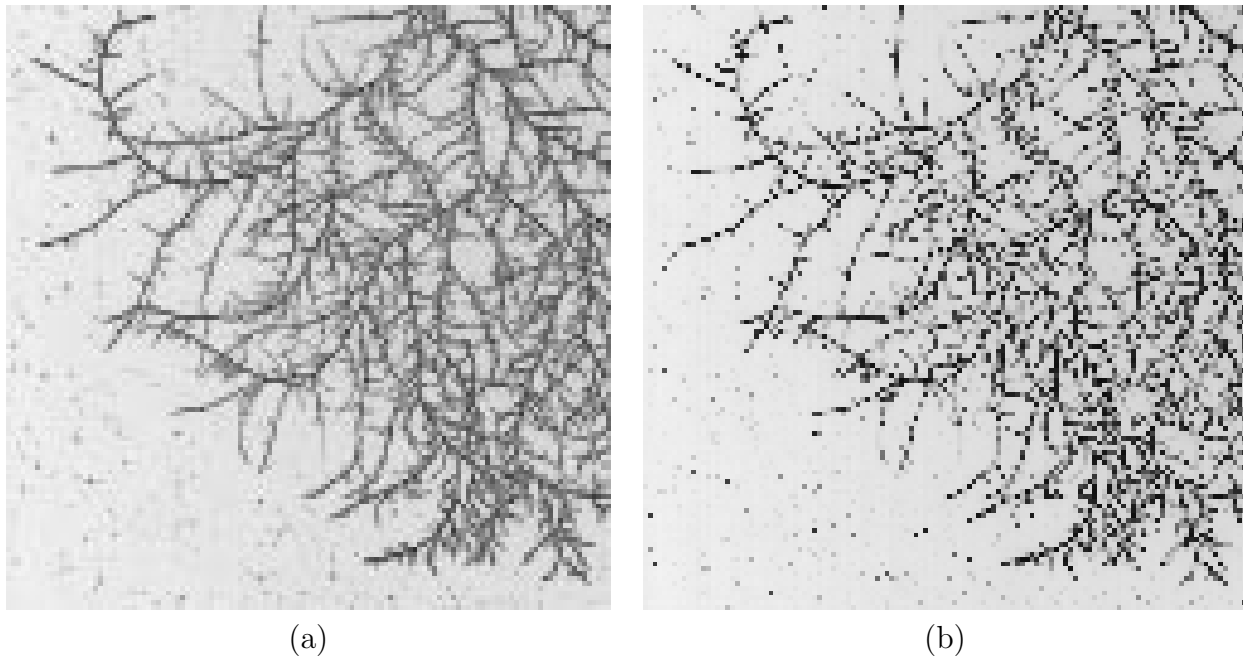


Figure 2.15: Effect of **(a)** block averaging and **(b)** pixel sampling in reducing the resolution of the fungal image. Here the size is reduced by four horizontally and vertically.

$$f'_y = (1 - a)f_i + af_{i+1}$$

where i labels the nearest original image pixel $< y$, and $y = i + a$. In two dimensions, we may use bilinear interpolation based on the four nearest pixels in the original image, leading to the formula

$$f'_{yx} = (1 - a)(1 - b)f_{ij} + (1 - a)bf_{i,j+1} + a(1 - b)f_{i+1,j} + abf_{i+1,j+1}$$

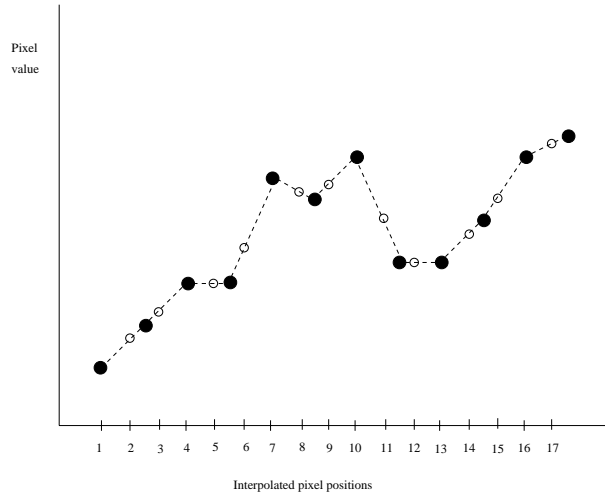


Figure 2.16: Interpolation in one dimension. If the filled circles denote the original pixel values, then linear interpolation will assign the values on the dotted line to the inter-pixel points. If, for example, an enlargement by a factor of 1.5 is required, then the open circles are used to supply pixel values. At every third pixel, they coincide with original pixel positions.

where the position y, x in the interpolated image corresponds to position $(i + a, j + b)$ in the original image, with $0 \leq a, b < 1$.

2.5 Registration

If we have several images of the same scene, which we wish to analyse jointly, it simplifies matters if they match exactly, i.e. each pixel position corresponds to the same part of the scene in the different images. The images may be different variables obtained with the same instrument, or the same scene viewed with different instruments, under different conditions or at different times. Sometimes the images will not exactly coincide, in which case they must be distorted in order to match. This process is known as **image registration**.

In order to distort images to match each other, we must have an accurate knowledge of the differences between them. This may be expressed in terms of which parts of the digital images refer to which particular parts of the scene, and should therefore be aligned in the two images. Occasionally this may be known *a priori* from the nature of the imaging instruments. More often it will not, and we have to discover what the image differences are. This is usually done by finding **control points**, sometimes called **landmarks**. These are points in the scene whose pixel position can be obtained precisely in all of the images to be registered. Often they are corners of objects or intersections of narrow straight line or curved features. Enough control points are needed so that the pattern of their positions captures all of the significant differences in object feature positions between the images. Control points may be identified manually, but automatic methods for finding what distortion is needed for images to match have also been proposed — see for example Herbin, Venot, Devaux, Walter, Lebruchec, Dubertret and

Roucaÿrol (1989) and Rignot, Kowk, Curlander and Pang (1991)

The next stage is to model the differences between the images. To simplify matters, let us assume that we have just two images. We select one of the images and model the position of features in the second image as some function of their position in the first image. A pixel with coordinates (i, j) in the first image will correspond to row and column coordinates $u(i, j)$ and $v(i, j)$ in the second image. If we have more than two images to register, we can register all of them to one of the images, which reduces to repeating two-image registrations, or register all of the images to a configuration based on the average position of the control points in the different images.

A simple model for the distortion is the **affine model**, which assumes that the distortion consists only of translation, rotation, size scaling in the horizontal and vertical directions (possibly by different amounts) and shear (what turns a square into a diamond shape). It is approximately the distortion you would see if you rotated this book and viewed it from different distances and angles. The distortion may be modelled

$$\begin{aligned}u(i, j) &= a_0 + a_1i + a_2j \\v(i, j) &= b_0 + b_1i + b_2j\end{aligned}$$

The coefficients $a_0, a_1, a_2, b_0, b_1, b_2$ may be estimated from the control points by linear regression (Draper and Smith, 1981).

If the distortion is non-linear, then a more complex function is needed. One generalisation is to use a quadratic or higher-order polynomial function. A quadratic function would have the form

$$\begin{aligned}u(i, j) &= a_0 + a_1i + a_2j + a_3i^2 + a_4ij + a_5j^2 \\v(i, j) &= b_0 + b_1i + b_2j + b_3i^2 + b_4ij + b_5j^2\end{aligned}$$

and this may also be estimated by linear regression. Another possibility is to use thin-plate spline functions (Bookstein, 1989). These are intended to provide an exact fit to the distortion in the control points, subject to minimising a function (the ‘bending energy’) of the curvature.

The next stage of image registration is to modify one of the images so that it matches the other. This is often referred to as **unwarping** (and sometimes, confusingly, as **warping**). If we have identified the distortion function as defined above, we unwarped the second image so that it matches the first. To do this we take each pixel position (i, j) in the first image, and assign to it the image value at position $(u(i, j), v(i, j))$ in the second image. In general, $(u(i, j), v(i, j))$ will not define an integer position in image two, and so will not correspond to a unique pixel value. We have a number of options for how to obtain a pixel value:

- We may take the value at the pixel which is nearest to $(u(i, j), v(i, j))$ by rounding. This is known as **nearest neighbour** interpolation. It is computationally the quickest, but in some circumstances produces unsatisfactory results. Pixels may be replicated (be assigned to more than one pixel in the unwarped image) by different amounts.
- **Bilinear interpolation** as described in §2.4 uses the four pixels nearest to $(u(i, j), v(i, j))$

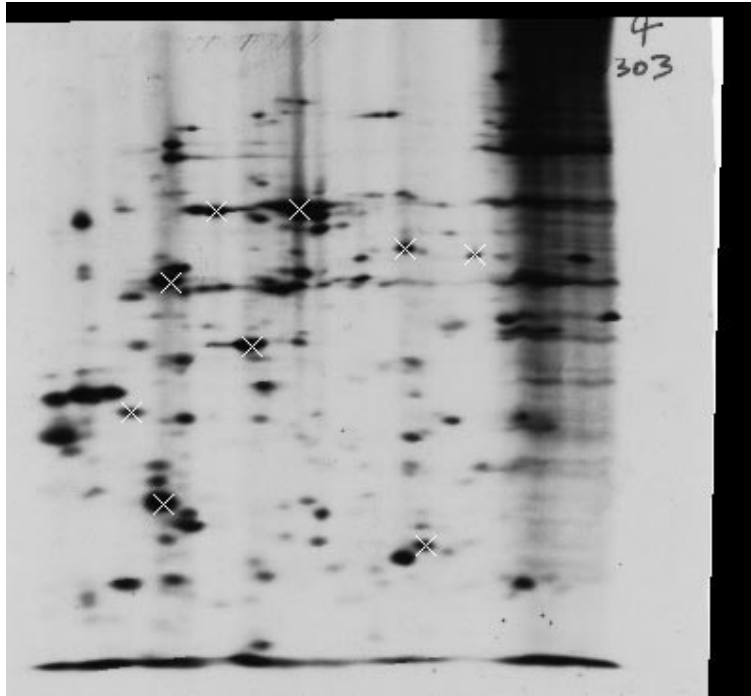
- **Cubic convolution** (Bernstein, 1976) requires the fitting of a cubic polynomial to the nine nearest pixels.

A discussion of these issues, and image registration in general may be found in Rosenfeld and Kak (1982; Ch.9).

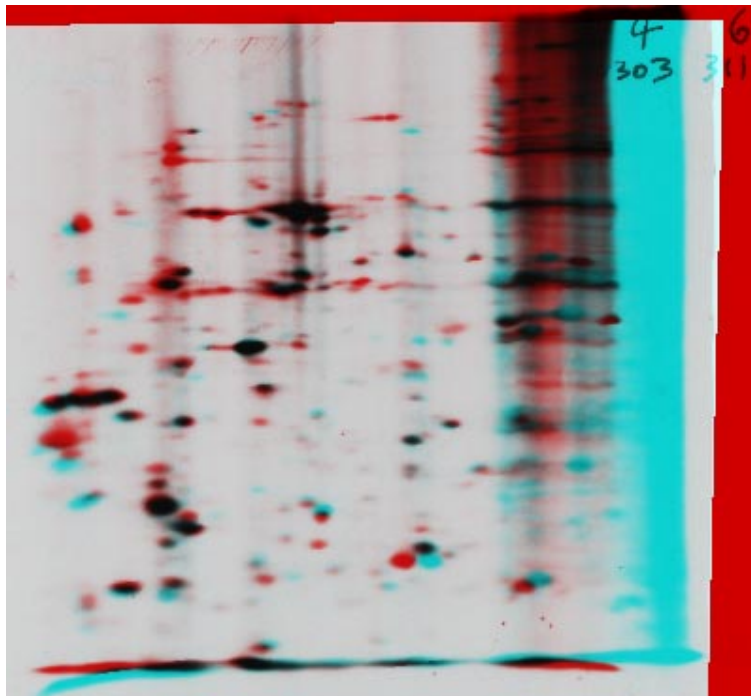
Fig 2.17 shows the registration of the electrophoresis image of Fig 1.9(a) to that of Fig 1.9(b). This was based on nine control points at the centre of protein spots which have been found not to vary in different genetic strains of the malarial parasite. The affine transformation and nearest neighbour interpolation was found to give acceptable image registration for these images. Fig 2.17 can be used to find which proteins are the same in both strains, and which are different. This application is discussed in greater detail in Horgan, Creasey and Fenton (1992).

2.6 Summary

- Images need to be examined before an automatic analysis is produced.
- Different displays are appropriate depending on whether the image is binary, greyscale or multivariate.
- Binary display is straightforward: we show one pixel type as black, the other as white.
- To display greyscale images, we allocate a display intensity to each possible pixel value. This generates a look-up-table.
- Transforming the pixel values, directly or only in the look-up-table, may help features to be seen. Useful transformations are
 - Piecewise linear (to allocate the full display intensity range to a limited range of pixel values).
 - Exponential or logarithmic (to emphasise variation in large or small pixel values respectively).
 - Histogram equalization (to allocate ranges of display intensity according to frequency of occurrence of pixel values in the image).
- Multivariate images are best displayed in colour, which can be based on the human red, green, blue (RGB) system of perceiving colour.
- Pseudocolour can help in examining a greyscale image.
- Zooming may be done by pixel replication, if we wish to enlarge by an integer ratio, or by interpolation otherwise.
- Reduction should be done by block averaging, which is slower, but loses less detail than pixel sampling.
- Registration forces the features in more than one image of the same scene to be in the same pixel position.



(a)



(b)

Figure 2.17: Registration of electrophoresis image in Fig 1.9(a) to Fig 1.9(b) **(a)** shows the warped version of Fig 1.9(a). The protein spots used as control points are shown by \times 's. **(b)** shows a superimposition of this image, in blue/green, with Fig 1.9(b) in red. Spots which are in both images appear black, those in warped Fig 1.9(a) only appear red and those in Fig 1.9(b) only appear blue/green. This is because a black spot in the blue and green components which is not black in the red component appears red, and *vice versa*. This superimposition makes it easy to see protein spot variation.