Chapter 8
Spatial Transformation

8.1 Introduction

A spatial transformation of an image is an alteration that changes the image's orientation or 'layout' in the spatial domain. Spatial transformations change the position of intensity information but, at least ideally, do not change the actual information content.

A transformed image might be moved up or down, or left or right, relative to some reference point. Alternatively, it might be rotated, changed in size, or distorted in some way that changes the shape of objects in the image. All these transformations are possible with digital images. Sometimes they require only a simple rearrangement of the positions of pixel data without any recalculation of pixel intensities or colors. In other cases, in attempting to preserve an image's original intensity information and minimize artifacts, it may be necessary to calculate new pixel intensities from the original image data. Even for a well-defined transformation, for example a 10% reduction in size, the method of calculation of new pixel information can have a profound effect on the appearance of the transformed image.

8.2 Translation

The simplest image transformation is a spatial translation. A translation might, for example, be used to align anatomy in one image with the same anatomy in another image when the images are overlaid. To perform a translation the image data simply moves relative to some reference point, typically the edge of the image matrix, without changing size or rotating (Fig. 8.1b). To do this it is only necessary to copy the rows or columns of image data into new positions, and to make some decision about what values to put into the newly empty rows and columns. There is no need for calculation of new pixel data and the original data is simply copied to a new position. Apart from the data which has perhaps been moved off the edge of the image matrix there is no change in image quality. This type of operation happens routinely in a computer whenever we use a scroll bar to move the displayed image inside a window.

R. Bourne, *Fundamentals of Digital Imaging in Medicine*,
DOI 10.1007/978-1-84882-087-6_8, © Springer-Verlag London Limited 2010

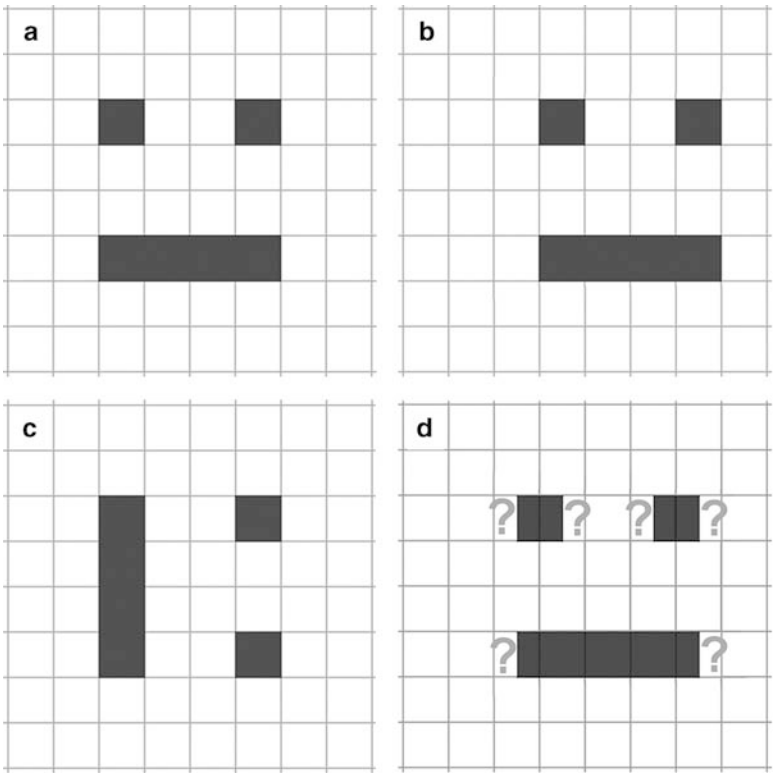

Fig. 8.1 Some simple image transformations. Image **b** shows the data of image **a** shifted (*translated*) one pixel to the right. Image **c** shows the data of **a** rotated *exactly* 90°. Translation is a relatively simple process if the spatial shift is a whole number of pixels – the original image data is simply copied into different positions in the new image matrix. Similarly, rotations that are simple multiples of 90° (**c**) only require a rearrangement of the original image data. A translation that is *not* a whole number of pixels (**d**) is problematic – a decision must be made about how to calculate new pixel values from the original image data. In this illustration the grid represents the rows and columns of the image matrix. In image **d** the 'translated' original image data is illustrated *as if* it was overlaid on the new image matrix. However, this representation has no physical form and the actual pixel values for the new image matrix must somehow be derived from the original data

Rearranging rows and columns of pixels is a relatively simple operation, but complications arises when the required shift is not equal to an integer number of pixels. Exactly what pixel data do we put in the new image matrix when there is no direct correspondence between pixels of the original image and the new image? Imagine that we wish to shift an image half of one pixel to the right (Fig. 8.1d). We might reasonably decide that the best value for each pixel in the new matrix would be the average of the two pixels that 'overly' it in the shifted original image matrix. The simplest way to do this would be to calculate the average of each adjacent pair of pixels on a row of the original image matrix and write this new value into the right pixel position in the new image matrix. This will give us a *representation* of

the original image *translated* half a pixel to the right. We will be missing a pixel at the extreme left side of the image, but we will assume this is not a problem. A possibly undesirable side-effect of the averaging method will be a slight blurring of the original image information. If there was a one pixel wide white (intensity 255) line on a black background in the original image it will be converted to a two pixel wide mid-gray (intensity 128) line in the translated image. In a similar way all vertical edges in the original image will become less sharp. Because we shifted our image horizontally, and thus only averaged pairs of pixels along rows, there will be no blurring in the vertical direction – horizontal edges will retain their original sharpness. The averaging of pairs of pixels in the image has caused a loss of some of the original intensity information. Obviously there are potentially serious problems arising from a pixel-averaging approach to the image translation problem. Some better methods are described below.

8.3 Rotation

A 90°, 180°, or 270° rotation of an image does not require any recalculation of pixel data. For example, to perform a 90° rotation all that is necessary is to copy all the rows of the original image data into the appropriate columns of a new image matrix (Fig. 8.1c). An $m \times n$ pixel image (rows × columns) will become an $n \times m$ pixel image. Similarly, flipping an image horizontally or vertically is only a matter of data rearrangement and requires no calculation of pixel data.

Rotating an image by an angle that is not a multiple of 90 degrees is a less trivial process because the rotated original image data no longer align exactly, pixel by pixel, with the new image matrix (Fig. 8.2). What values do we put in the new image matrix? Figure 8.3 illustrates two simple solutions to this problem based on how

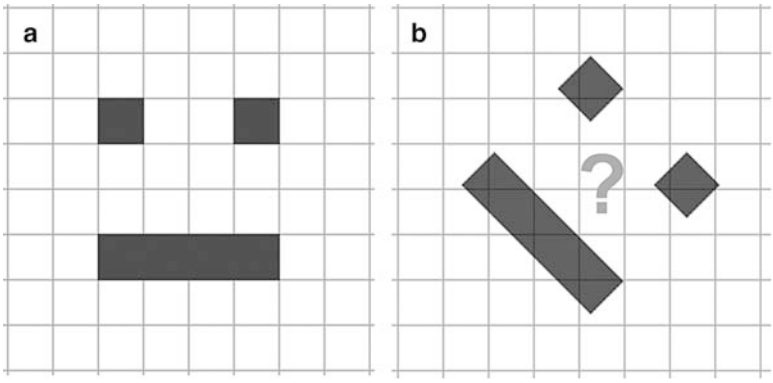

Fig. 8.2 When an image is rotated by an angle that is not a multiple of 90° the original pixel data cannot be mapped exactly, pixel by pixel, to the new image matrix. Figure 8.3 illustrates two trivial, but unsatisfactory, solutions

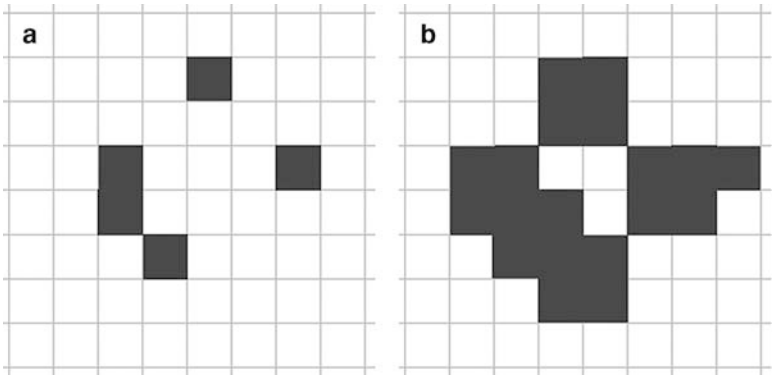

Fig. 8.3 Two possible (but rather unhappy looking) solutions to the image rotation problem posed in Fig. 8.2. In image **a** any pixel in the new image matrix that is overlaid by *more than half* a dark pixel of the rotated original image is given the dark value. In image **b** any pixel in the new image matrix that is overlaid by *any fraction* of a dark pixel of the rotated original image is given the dark value

much overlap there is between rotated original pixels and new image pixels. One approach is to make all new matrix pixels dark if any part of an original dark pixel overlies it. The second approach only makes a new image pixel dark if more than half an original dark pixel overlies it. In this example the original image information is very badly distorted by both the rotation processes – one method gives an image with too few dark pixels and the other method gives too many.

We can intuitively see that a better approximation to the original image appearance is possible, but we need some sort of rule or algorithm to calculate the new pixel values. Perhaps we should aim to have about the same number of dark pixels in our rotated image as in the original. One way to achieve this would be to give each pixel in the new image matrix the value of the *nearest* pixel in the rotated matrix. What is the nearest pixel? Let's say it is the overlying pixel whose center is closest to the center of the new pixel. This 'nearest neighbor' method gives a reasonably good result (Fig. 8.4a) except that the original straight edge has become jagged. Although this example shows highly magnified pixels, this jagged edge effect ('jaggies') can often be seen at normal display scales.

The jagged edge effect can be reduced by blurring – exploiting the limited ability of our vision to distinguish small changes of contrast at very small scales. We can do this by converting the original two intensity values that define the straight edge into several intensities. This approach is illustrated in Fig. 8.4b. In the example with enlarged pixels the smoothing effect is a little difficult to appreciate, but see Fig. 8.5 for an example at smaller scale.

When an $m \times n$ image is rotated by some arbitrary angle the original corners will not always lie inside an $m \times n$ or $n \times m$ matrix. If the information in the corners of the original image is important then the new image matrix will have to be made larger than the original and the 'empty' corners filled with some appropriate background values.

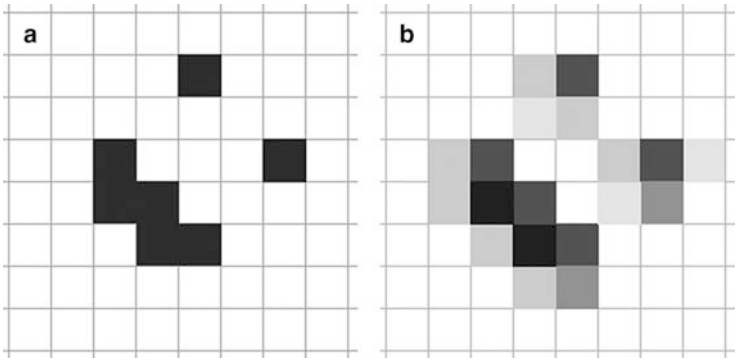

Fig. 8.4 More sophisticated solutions to the image rotation problem posed in Fig. 8.2. In image **a** pixels acquire the value of the original pixel whose *rotated* center position lies closest to the center of the new matrix pixel. In image **b** pixels acquire a value that is the product of the rotated original image pixel value and the degree to which it overlies the new pixel. These methods provide visually better representations of the original image data than the simpler methods illustrated in Fig. 8.3

The problems of how to represent a rotated edge, or a half pixel translation, are part of the more general problem of how to calculate pixel values for a transformed image when there is no direct correspondence between the transformed original pixels and the new image matrix. The general name for the process is *interpolation*.

8.4 Interpolation

Interpolation is the calculation of the expected value of a function at a particular point when only the values at nearby points are known. In the case of 2D images we are interested in 2D functions that describe image intensity relative to two spatial coordinates. There are three common interpolation methods used in 2D image processing: *Nearest-neighbor*, *Bilinear*, and *Bicubic*.

8.4.1 Nearest-Neighbor

We introduced the 'nearest-neighbor' idea in the image rotation section above, but it can be applied to all image transformations. This method calculates the distance between the 'notional' centers of the transformed image matrix pixels and the centers of the new matrix pixels. Each pixel in the new image matrix is assigned the *exact* value of the nearest pixel in the rotated original matrix. Because there is no adjustment of pixel intensities this method preserves contrast differences between adjacent pixels. Sharp edges are not blurred but they may appear spatially distorted at the pixel level – either becoming jagged if they were originally vertical or hor-

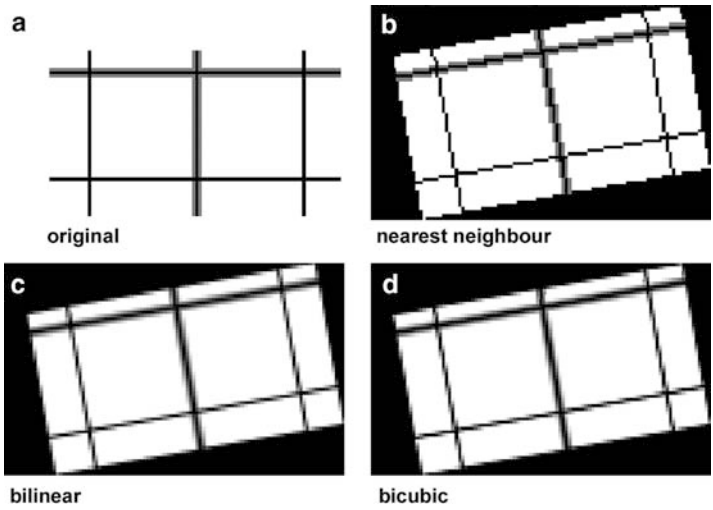

Fig. 8.5 Comparison of three different interpolation methods applied to the problem of image rotation. The original image (**a**) includes both sharp and blurred edge detail. Nearest neighbor interpolation (**b**) preserves the original sharp edge detail but produces conspicuous jagged aliasing artifacts. Bilinear interpolation (**c**) reduces aliasing but blurs the sharp edges. Bicubic interpolation (**d**), for this particular image, gives a result intermediate between nearest neighbor and bilinear. These example images have been enlarged to illustrate the pixel-level effects, some of which may not be obvious at more conventional display scales (see Fig. 8.6). Note that the rotated image requires a larger matrix than the original. Here the background has been filled with black pixels

izontal, or possibly becoming smooth and straight if they were originally jagged. These distortions are called *aliasing*.

8.4.2 Bilinear

The bilinear method calculates a new pixel value from the four centers of the transformed original pixels that surround the new pixel center. The new pixel value is calculated by linear interpolation in both the x and y directions – hence the term 'bilinear'. The value for the new pixel is the average of the nearest four original pixels *weighted* according to the proximity of the new pixel center to the centers of the four original pixels. Bilinear interpolation leads to some blurring of edge detail but, for the same reason, it reduces the creation of jagged aliasing artifacts.

8.4.3 Bicubic

The bicubic method is a more sophisticated version of the bilinear method. Instead of connecting pairs of points with straight lines the bicubic method fits two second

order polynomials (hence 'bicubic') to the *sixteen* pixels of the transformed original image matrix that are nearest to the center of each new image pixel. Because the bicubic method can represent a 'curvature' in the intensity profile of the original image it produces transformed images that more closely emulate the original than the bilinear method – especially if the image does not contain very sharp edges and lines.

Comparisons of the three interpolation methods are illustrated for a simple test image (Fig. 8.5) and a detailed medical image (Fig. 8.6). It is plain that for anatomical medical images viewed at normal scale these three different interpolation methods do not produce markedly different effects. The general consensus from photographic image processing is that bicubic interpolation gives the best looking

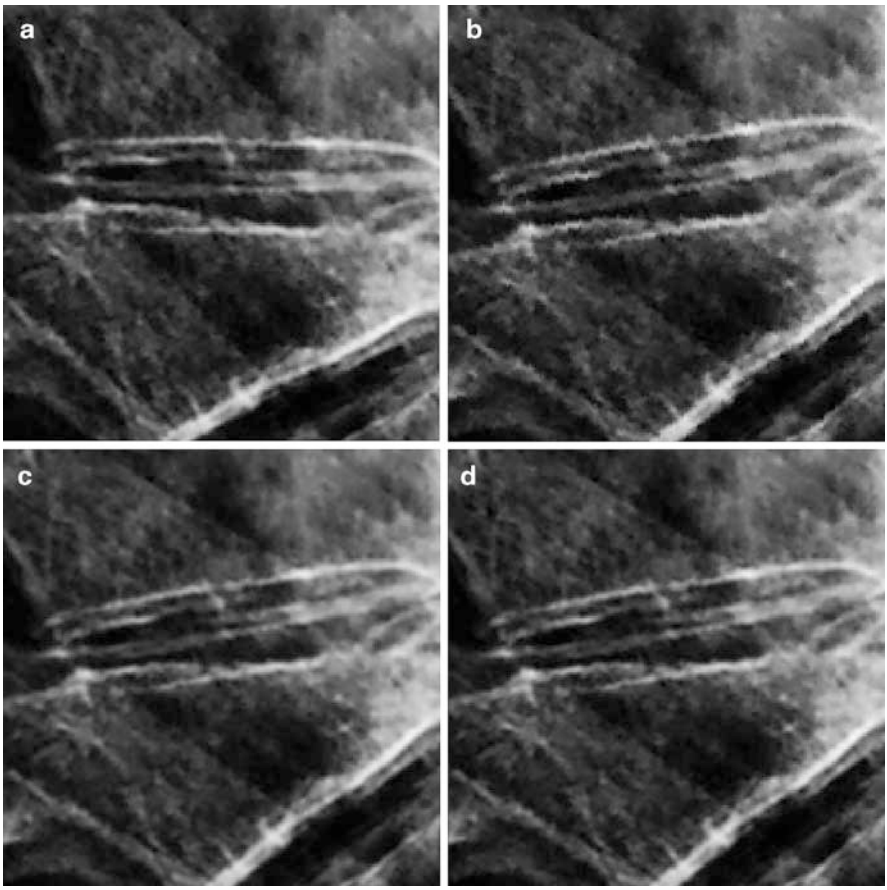

Fig. 8.6 Comparison of interpolation methods applied to rotation (10°) of a detailed X-ray image. The original image (**a**) was 200 × 200 pixels. The nearest neighbor method (**b**) produces just-detectable aliasing artifacts along the sharpest edges. The bilinear (**c**) and bicubic (**d**) methods appear artifact-free when the image is viewed at a conventional scale

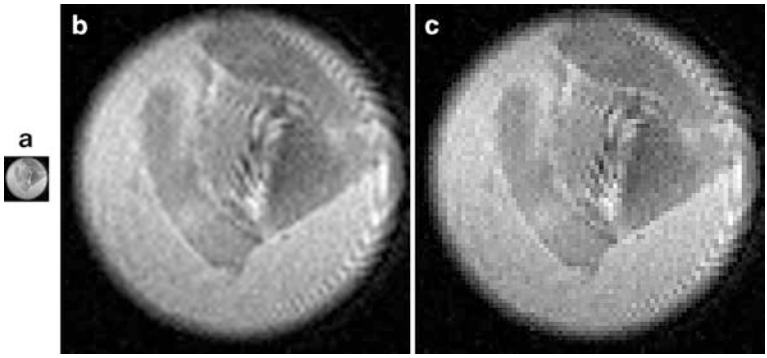

Fig. 8.7 Nearest neighbor interpolation is useful when enlarging an image for the purpose of examining or illustrating the raw data. Here we see an MRI of a rat brain fragment. The original 80 × 80 pixel image (**a**) is much too small to study when displayed (in this case printed) as an 80 × 80 pixel print image. Because the printer works at about 118 dots/cm we need to increase the original image matrix to about 590 × 590 pixels to make the print size 5 cm^2. Enlargement by the bicubic method (**b**) blurs the boundaries between the original pixels. The nearest neighbor method (**c**) preserves the original pixel information. For the original pixel information to be completely undistorted the enlargement factor must be an integer (in this image the unusual edge pattern is an MRI artifact not related to image enlargement)

results for most images, and most image processing software uses bicubic interpolation as the default method. The extra calculations needed for bicubic interpolation are not a significant problem for modern computers. However, this does not mean that bicubic interpolation should be used for *all* image transformations. Nearest neighbor interpolation is useful when we want to enlarge images for the specific purpose of examining the original pixel intensity details (Fig 8.7).

8.5 Resizing Images

We resize images on monitor displays so frequently that we rarely give any thought to the process, but changing the size of images presents the same sort of problems in calculation of the new image pixel values as does translation and rotation (Fig. 8.8). Most of the time the bicubic method is ideal – it is built in to the graphics card of computers so that the displayed image size can be changed easily without obvious artifacts, and it is the usual default option for resizing image data in image processing software. The smoothing effect of bicubic interpolation can be used to suppress pixelation effects.

Sometimes the spatial resolution of raw image data is such that pixelation is clearly visible if the image is viewed at its native resolution. This is especially the case for imaging techniques that have poor spatial resolution, or small image matrices – e.g. some MRI methods. The propensity of human visual perception to notice lines and edges means that pixelated images are generally regarded as difficult to

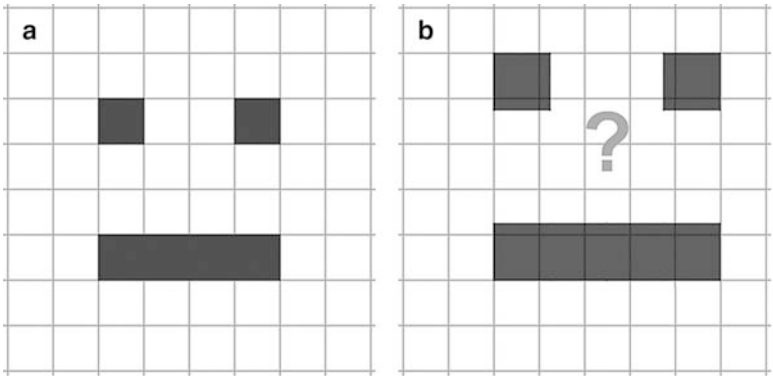

Fig. 8.8 When an image is scaled, in this example enlarged by 25%, similar problems occur to those we saw for rotation in Fig. 8.2 – the original pixel data cannot be mapped exactly, pixel by pixel, to the new image matrix

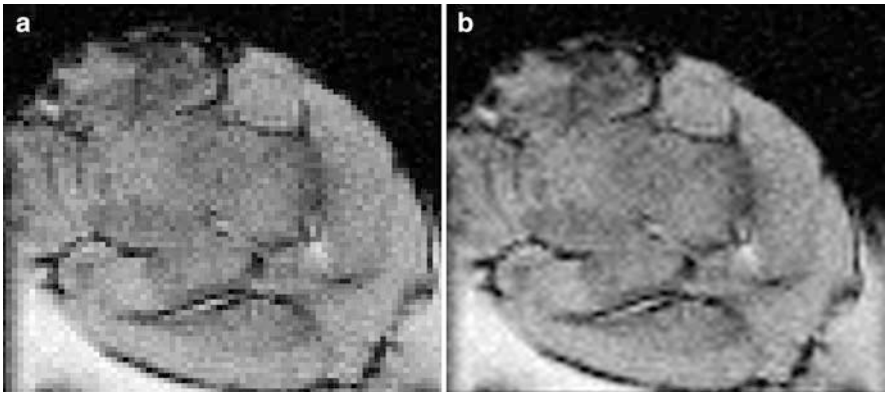

Fig. 8.9 Intentional blurring of pixel boundaries can enhance the 'readability' of anatomical information. This mouse brain was originally imaged with a 128 × 128 matrix – part of which is illustrated as image **a**. Bicubic interpolation (image **b**) increases only the *apparent* spatial resolution and blurs the raw data pixel boundaries. The anatomical information is easier to see in image **b**. In Fig. 8.7c we chose the nearest neighbor method to enlarge and illustrate the raw pixel data. Here the printed 'raw data' image (**a**) is also the result of nearest neighbor interpolation

interpret because the visible edges of the pixels distract from the tonal information. These images are often interpolated into a larger image matrix which, when displayed, does not exhibit obvious pixelation. A typical example is the MRI data illustrated in Fig. 8.9 which shows both the raw and interpolated image data. Interpolation increases the *apparent* spatial resolution.

It is important to remember that interpolation of spatial information in order to increase the apparent spatial resolution and 'readability' of an image does not add information to the image. The spatial resolution of an image is absolutely limited by the spatial resolution of the raw data.

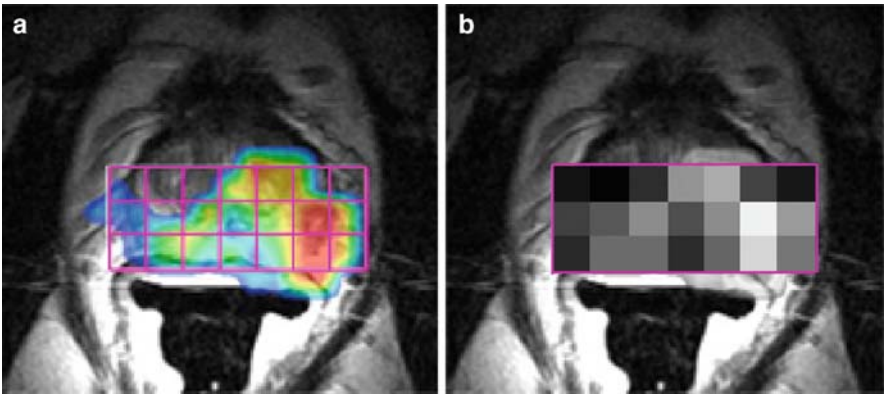

Fig. 8.10 Interpolation of magnetic resonance spectroscopy data. High resolution 'metabolite maps' (**a**) are commonly derived from low spatial resolution raw data (**b**) acquired from a specific measurement region – inside the purple rectangle. The high resolution map is easier to interpret than the raw data but suggests spatial details, both within and outside the measurement region, that have not been measured. The true spatial resolution is in fact lower than the nominal resolution (individual purple boxes) due to the point spread function

Spatial interpolation is very commonly used to aid in the interpretation of low spatial resolution data, but it can easily lead to interpretation errors. Figure 8.10 illustrates a typical color 'metabolite map' derived from magnetic resonance spectroscopy (MRS) of the human prostate. The nominal raw spatial resolution of the data is of the order of 0.5–1 cubic centimeters (Fig. 8.10b). This low-resolution data is difficult to interpret so it is interpolated into a high resolution color contour map overlaid on the anatomical image (Fig. 8.10a). This contour map *suggests* a level of spatial detail that is not present in the acquired data, and even includes contours in regions from which no measurements were acquired. If the raw data is not examined carefully there is a strong tendency for the viewer to forget that the true spatial resolution of the measurement is very coarse and to overinterpret the high resolution image.

8.6 Summary

- Image transformation – the enlarging, shrinking, or rotating of an image – requires generation of new image matrices. In general, the pixel intensities in the transformed image matrix must be calculated from those in the original image matrix. Pixel intensities in the transformed image are calculated or assigned by interpolation methods – most commonly *nearest neighbor* and *bicubic*.
- The nearest neighbor interpolation method assigns only original image matrix intensity values to pixels in the transformed image matrix. The value

assigned is the value of the nearest pixel center in the notionally transformed original image matrix. The nearest neighbor method may produce 'jagged' edge artifacts, but is useful for display of original intensity information in transformed images.

- The bicubic interpolation method calculates new intensity values for pixels in the transformed image matrix. The value assigned is the estimated local value on a curved surface representing the intensities of the notionally transformed original image matrix. The bicubic method is often used to produce more visually appealing images by suppression of pixelation, however, this may lead to misinterpretation of spatial resolution. Interpolation *cannot* create new image information.