6

Basic Concepts of Reconstruction Algorithms

With this chapter we begin our systematic study of reconstruction algorithms. We introduce the notation used in the rest of the book. We categorize reconstruction methods into two groups: transform methods and series expansion methods. We explain the nature of the algorithms in the two groups and indicate the desirable characteristics of reconstruction algorithms.

6.1 Problem Statement

Until now we have always used rectangular (Cartesian) coordinates for describing a function of two variables. Thus, we have used $\mu_{\bar{e}}(x, y)$ to denote the relative linear attenuation at the point (x, y), where (x, y) was in reference to a rectangular coordinate system, see Fig. 2.4. However, in the more mathematical work that follows it is more convenient to use polar coordinates (r, ϕ), which are related to the rectangular coordinates (x, y) by the formulas $r = \sqrt{x^2 + y^2}$, $\phi = \tan^{-1}(y/x)$, $x = r\cos\phi$, $y = r\sin\phi$. We use the phrase a *function of two polar variables* to describe a function f whose values $f(r, \phi)$ represent the value of some physical parameter (such as the relative linear attenuation) at the geometrical point whose polar coordinates are (r, ϕ). The mathematically distinguishing feature of a function f of two polar variables is that $f(0, \phi_1) = f(0, \phi_2)$, for all values of ϕ_1 and ϕ_2. This reflects the fact that the physical parameter represented by f can have only one value at the origin. Furthermore, we do not restrict the domain of the polar variables, that is, we allow r and ϕ to have any real values; hence a function f of two polar variables must also satisfy the condition $f(r, \phi) = f(-r, \phi + \pi)$.

In Section 4.1 we have defined a picture as a function of two variables whose value is zero outside the picture region, which is a square (of size $\sqrt{2}E \times \sqrt{2}E$, say) whose center is at the origin of the coordinate system. In what follows, we use f to denote the function of two polar variables r and ϕ, which is used to define the picture to be reconstructed. We know that

$$f(r,\phi) = 0, \quad \text{if } |r\cos\phi| > E/\sqrt{2} \text{ or } |r\sin\phi| > E/\sqrt{2}. \tag{6.1}$$

In particular, $f(r,\phi) = 0$ if $r > E$.

A possible physical interpretation of the picture function f is that the picture region is the reconstruction region of Fig. 2.4 and $f(r,\phi)$ is the relative linear attenuation at the point (r,ϕ). The remaining discussion is independent of such an interpretation. Reconstruction algorithms are applicable whatever physical property $f(r,\phi)$ is supposed to represent (see Section 1.1).

One important difference between studying f simply as a function and studying it as a representation of the distribution of some physical property is the way the mathematics is handled. Reconstruction algorithms are often based on mathematical theorems of the form: "If f has the property that …, then … ." We do not hesitate to use the conclusion of such a theorem, whenever the premise appears to be reasonable on physical grounds.

In particular, we shall not hesitate to assume, whenever needed, that pictures satisfy certain integrability conditions. (We use integrals without precise definition. While just about all that we say is valid for any standard definition of an integral; those who wish to make our approach mathematically watertight should use integrals in the sense of Lebesgue.) One of our assumptions is that any picture function f is *square integrable*; i.e., that

$$\int_0^{2\pi} \int_0^E (f(r,\phi))^2 \, r \, dr \, d\phi \tag{6.2}$$

exists. (Existence here means that the integral can be evaluated and its value is a real number.) It follows from this assumption that, for any two picture functions f_1 and f_2, the *distance*

$$d(f_1,f_2) = \sqrt{\int_0^{2\pi} \int_0^E (f_1(r,\phi) - f_2(r,\phi))^2 \, r \, dr \, d\phi}, \tag{6.3}$$

between them also exists. Clearly, (6.3) is related to the picture distance measure defined by (5.1).

We now define the *Radon transform* of a function f of two polar variables. First we introduce a notational convention that is used throughout the book. The Radon transform is an example of an *operator*; when acting on a function it produces another function. We use capital script letters to denote operators; for example, we use $\mathscr{R}$ to denote the Radon transform. If f is a function, then the function that is its Radon transform is denoted by $\mathscr{R}f$. The value of $\mathscr{R}f$ at a point (ℓ,θ) in its domain is denoted by $[\mathscr{R}f](\ell,\theta)$. The Radon transform of f is defined for real number pairs (ℓ,θ) as follows:

$$\begin{aligned} [\mathscr{R}f](\ell,\theta) &= \int_{-\infty}^{\infty} f\left(\sqrt{\ell^2+z^2},\, \theta + \tan^{-1}(z/\ell)\right) dz, \quad \text{if } \ell \neq 0, \\ [\mathscr{R}f](0,\theta) &= \int_{-\infty}^{\infty} f(z, \theta + \pi/2)\, dz. \end{aligned} \tag{6.4}$$

Observing Fig. 2.4, we see that $[\mathscr{R}f](\ell,\theta)$ is the line integral of f along the line L. (Note that the dummy variable z in (6.4) does not exactly match the variable z as indicated in Fig. 2.4. In (6.4) $z = 0$ corresponds to the point where the perpendicular dropped on L from the origin meets L.) The existence of the Radon transform for any ℓ and θ is another one of our integrability assumptions.

Observe that

$$[\mathscr{R}f](\ell,\theta) = [\mathscr{R}f](-\ell,\theta+\pi) = [\mathscr{R}f](\ell,\theta+2\pi) \tag{6.5}$$

and that, as a consequence of (6.1),

$$[\mathscr{R}f](\ell,\theta) = 0, \qquad \text{if} \quad |\ell| \geq E. \tag{6.6}$$

In view of these equations, the function $\mathscr{R}f$ is completely determined by its values at the points (ℓ,θ) with $-E < \ell < E$ and $0 \leq \theta < \pi$.

There is an important difference between the domains of the functions f and $\mathscr{R}f$. The picture function f is defined for pairs of real numbers (r,ϕ), which represent the polar coordinates of points in the plane. Hence the value of $f(0,\phi)$ is the same for all values of ϕ, since $(0,\phi)$ always represents the origin. This is not the case for $\mathscr{R}f$. Its value for the pair $(0,\theta)$ is the line integral of f along a line through the origin making an angle θ with the positive y axis. Hence, unless f is circularly symmetric about the origin, $[\mathscr{R}f](0,\theta)$ depends on θ. The pair of real numbers (ℓ,θ) in the domain of $\mathscr{R}f$ is not to be interpreted as polar coordinates of a point in the plane.

Roughly speaking, the operator $\mathscr{R}$ associates with a function f over the (r,ϕ) space another function $\mathscr{R}f$ over the (ℓ,θ) space. We can think of a single point in the (ℓ,θ) space as corresponding to a line L (at a distance ℓ from the origin making an angle θ with the positive y axis) in the (r,ϕ) space, since $[\mathscr{R}f](\ell,\theta)$ is the integral of f along L.

To further emphasize the relationship between the two spaces consider Fig. 6.1. It shows the loci in the (ℓ,θ) space of the points corresponding to two sets of lines in the (r,ϕ) space: (i) a set of parallel lines and (ii) a set of lines going through a fixed point.

Consider first the line K that makes an angle θ' with the baseline B (the positive x axis) in Fig. 6.1(a). Any line perpendicular to K makes an angle θ' with the positive y axis. Hence the locus of the set of points in the (ℓ,θ) space that corresponds to lines perpendicular to K is the straight line $\theta = \theta'$; see Fig. 6.1(b).

Consider next the point (r,ϕ) in Fig. 6.1(a). The distance ℓ from the origin of the line through it that makes an angle θ with the positive y axis is

$$\ell = r\cos(\theta - \phi). \tag{6.7}$$

Hence the locus of the set of points in (ℓ,θ) space that corresponds to lines through the point (r,ϕ) is the curve whose equation is (6.7), see Fig. 6.1(b).

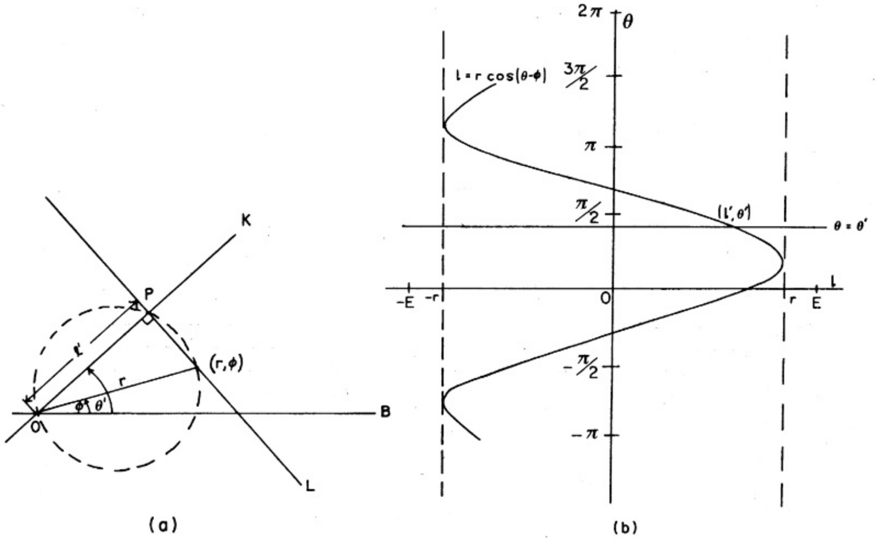


Fig. 6.1: The relationship between the (r, ϕ) space and the (ℓ, θ) space. (a) In the (r, ϕ) space, K is the line through the origin O making an angle θ' with the baseline B. The point (r, ϕ) is considered given and L is the line through (r, ϕ) orthogonal to K. L meets K at the point P, which is at a distance ℓ' from O. (b) In the (ℓ, θ) space, the points that correspond to the lines perpendicular to K in the (r, ϕ) space lie on the straight line $\theta = \theta'$. The points that correspond to the lines through (r, ϕ) in the (r, ϕ) space lie on the sinusoidal $\ell = r\cos(\theta - \phi)$. The point corresponding to L, namely (ℓ', θ'), is the intersection of these two curves. (Reproduced from [115], Copyright 1981.)

The point in (ℓ, θ) space that corresponds to the line L that is both perpendicular to K (and so makes an angle θ' with the positive y axis) and goes through the point (r, ϕ) is the point $(\ell', \theta') = (r\cos(\theta' - \phi), \theta')$.

The input data to a reconstruction algorithm are estimates (based on physical measurements) of the values of $[\mathscr{R}f](\ell, \theta)$ for a finite number of pairs (ℓ, θ); its output is an estimate, in some sense, of f. The main purpose of this chapter is to make this brief description precise.

Suppose that estimates of $[\mathscr{R}f](\ell, \theta)$ are known for I pairs: $(\ell_1, \theta_1), \ldots, (\ell_I, \theta_I)$. For $1 \leq i \leq I$, we define $\mathscr{R}_i f$ by

$$\mathscr{R}_i f = [\mathscr{R}f]\,(\ell_i, \theta_i). \tag{6.8}$$

$\mathscr{R}_i$ is a *functional*; when acting on a function, it produces a real number. In what follows we use, unless otherwise stated, y_i to denote the available estimate of $\mathscr{R}_i f$ and we use y to denote the I-dimensional column vector whose ith component is y_i. We refer to the vector y as the *measurement vector*.

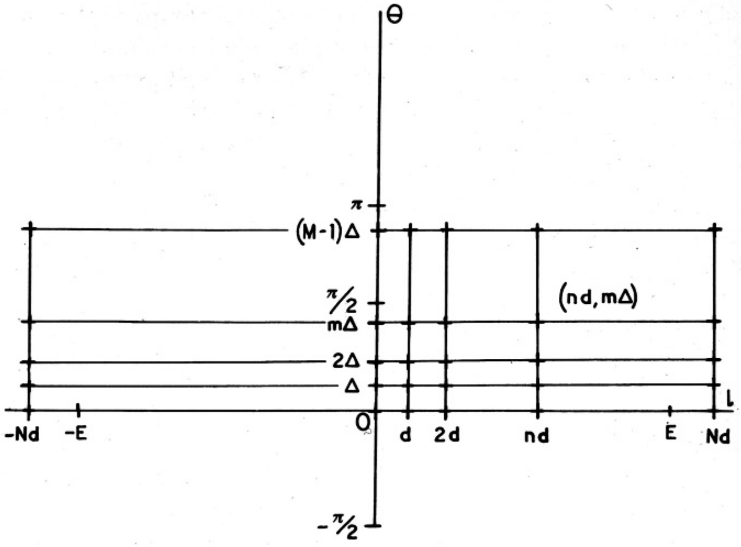


Fig. 6.2: The locations in the (ℓ, θ) space of the points that correspond to lines for which measurements have been collected in the parallel mode of data collection. It is assumed that a single source and a single detector move parallel to each other in $2N+1$ steps of size d, with $Nd > E$, the radius of the circular region containing the object to be reconstructed. After the data have been collected for these $2N+1$ lines (one view), the whole apparatus is rotated by an angle Δ, and the data are again collected for the $2N+1$ lines of the next view. This is repeated for a total of M views, where $M\Delta = \pi$. Thus, for a complete set of views, the apparatus rotates around to nearly cover a semicircle. A typical point in the (ℓ, θ) space is $(nd, m\Delta)$, which lies in the intersection of two straight lines $\ell = nd$ and $\theta = m\Delta$, with $-N \le n \le N$ and $0 \le m \le M-1$. (Reproduced from [115], Copyright 1981.)

When designing a reconstruction algorithm we assume that the method of data collection, and hence the set $\{(\ell_1, \theta_1), \ldots, (\ell_I, \theta_I)\}$, is fixed and known. Roughly stated, the reconstruction problem is

given the data y, **estimate** the picture f.

In the next two sections we discuss the basic approaches for estimating f. We shall usually use f^* to denote the estimate of the picture f.

$\mathscr{R}_i f$ is the value of $\mathscr{R} f$ at the point (ℓ_i, θ_i) in the (ℓ, θ) space. Any geometry of data collection provides us with a finite set of points (ℓ_i, θ_i) at which an estimate of $\mathscr{R}_i f$ is known. For example, Fig. 6.2 shows the arrangement of such points (ℓ_i, θ_i) for the parallel modes of data collection shown in Figs. 3.3(a) and (b); this arrangement forms a rectangular grid. The corresponding arrangements for the divergent modes of data collection (Figs. 3.3(c) and (d)) are more complicated; they are discussed in Chapter 10.

6.2 Transform Methods

One way of defining the estimate f^* of f is to give a formula that expresses the value of $f^*(r, \phi)$ in terms of r, ϕ, y_1, …, y_I. Such a formula may be a "discretized" version of a Radon inversion formula, which expresses f in terms of its Radon transform $\mathscr{R}f$. We refer to reconstruction methods based on such an approach as *transform methods*. In the rest of this section we give a more detailed explanation of what has been said in this paragraph.

The Radon transform associates with a function f of two polar variables another function $\mathscr{R}f$ of two variables. What we are looking for is an operator $\mathscr{R}^{-1}$, which is an *inverse* of $\mathscr{R}$ in the sense that $\mathscr{R}^{-1}\mathscr{R}f$ is f (i.e., $\mathscr{R}^{-1}$ associates with the function $\mathscr{R}f$ the function f). Just as (6.4) describes how the value of $\mathscr{R}f$ is defined at any real number pair (ℓ, θ) based on the values f assumes at points in its domain, we need a formula that for functions p of two real variables defines $\mathscr{R}^{-1}p$ at points (r,ϕ). Such a formula is

$$\left[\mathscr{R}^{-1}p\right](r, \phi) = \frac{1}{2\pi^2} \int_0^\pi \int_{-E}^{E} \frac{1}{r\cos(\theta - \phi) - \ell} p_1(\ell, \theta)\, d\ell\, d\theta, \qquad (6.9)$$

where $p_1(\ell, \theta)$ denotes the partial derivative of $p(\ell, \theta)$ with respect to ℓ; it is of interest to compare this formula with (2.5). We prove in Section 15.3 that, for any picture function f of two polar variables (satisfying some physically reasonable conditions), $\mathscr{R}^{-1}\mathscr{R}f = f$, in the sense that, for all points (r, ϕ),

$$\left[\mathscr{R}^{-1}\mathscr{R}f\right](r, \phi) = f(r, \phi). \qquad (6.10)$$

In order to understand the nature of the operator $\mathscr{R}^{-1}$, we express it as a sequence of simpler operators.

We use $\mathscr{D}_Y$, to denote *partial differentiation* with respect to the first variable of a function of two real variables. Thus, for any function p of two real variables and for any real number pair (ℓ, θ),

$$\left[\mathscr{D}_Y p\right](\ell, \theta) = \lim_{\Delta\ell \to 0} \frac{p(\ell + \Delta\ell, \theta) - p(\ell, \theta)}{\Delta\ell}, \qquad (6.11)$$

assuming of course that the limit on the right-hand side exists.

In our application, the function p that is operated on by $\mathscr{D}_Y$ is the Radon transform of a picture. It is quite easy to describe pictures f such that $\mathscr{D}_Y\mathscr{R}f$ is not defined for all (ℓ, θ). An example is the picture that has value one everywhere inside the picture region. There are mathematically rigorous ways of extending the definition $\mathscr{D}_Y$, so that it makes sense even in such cases. Here we simply assume that for any picture f that we may wish to reconstruct, the right-hand side of (6.11) is defined for $p = \mathscr{R}f$.

The next operator we wish to define is the *Hilbert transform* $\mathscr{H}_Y q$ *with respect to the first variable* of a function q of two variables. For any real number pair (ℓ, θ), we define

$$[\mathscr{H}_Y q]\,(\ell',\theta) = -\frac{1}{\pi}\int_{-\infty}^{\infty}\frac{q(\ell,\theta)}{\ell'-\ell}\,d\ell. \tag{6.12}$$

Note that this is an *improper integral* since its integrand becomes infinite at $\ell = \ell'$. It is to be evaluated in the *Cauchy principal value* sense; i.e.,

$$[\mathscr{H}_Y q]\,(\ell',\theta) = -\frac{1}{\pi}\lim_{\varepsilon\to 0}\left(\int_{-\infty}^{\ell'-\varepsilon}\frac{q(\ell,\theta)}{\ell'-\ell}\,d\ell + \int_{\ell'+\varepsilon}^{\infty}\frac{q(\ell,\theta)}{\ell'-\ell}\,d\ell\right). \tag{6.13}$$

In our application, q is $\mathscr{D}_Y\mathscr{R}f$ for some picture f. We again assume that for pictures that we wish to reconstruct the limit on the right-hand side of (6.13) exists.

Finally, we introduce an important operator called *backprojection*. Given a function t of two variables, $\mathscr{B}t$ is another function of two polar variables, whose value at any point (r,ϕ) is defined by

$$[\mathscr{B}t]\,(r,\phi) = \int_0^{\pi} t\,(r\cos(\theta-\phi),\theta)\;d\theta. \tag{6.14}$$

Observing Fig. 6.1(b), we see that the value at point (r,ϕ) of the backprojection of a function t is obtained by integrating t on a segment of the curve (from $\theta = 0$ to $\theta = \pi$) whose equation is (6.7).

The reason for the name backprojection is the following. Look at the line K in Fig. 6.1(a). It makes an angle θ' with the positive x axis (the baseline B). The "projection" of a function of two variables onto the line K is the function of one variable obtained from the line integrals of f along lines perpendicular to K. In other words, it is $[\mathscr{R}f]\,(\ell,\theta')$, considered as a function of ℓ alone. The line L that goes through a point (r,ϕ) and is perpendicular to K meets the line K at a point P that is at a distance $\ell' = r\cos(\theta'-\phi)$ from the origin.

Now consider the reverse process. Rather than producing $\mathscr{R}f$ from f by integrating (projecting) along lines such as L, produce from a given function t of two variables another function $\mathscr{B}t$ by spreading (backprojecting) the values of t along such lines. For a fixed θ' (determining the line K), the contribution of t to $\mathscr{B}t$ is the same for all points (r,ϕ) lying on the same line L perpendicular to K; and the value of this contribution is proportional to $t(\ell',\theta')$, where ℓ' is the distance of L from the origin. More precisely, given a point (r,ϕ), we evaluate the value of $\mathscr{B}t$ at (r,ϕ) by summing up (integrating), as θ' varies, the values of $t(\ell',\theta')$ for the ℓ' that is the distance of the line L from the origin. Since L goes through (r,ϕ) and K goes through the origin, the locus of the points P where these perpendicular lines meet as θ' varies is the circle with its diameter from the origin to the point (r,ϕ).

Combining (6.11), (6.12), and (6.14) we get that, for a function p of two variables and for any point (r,ϕ),

$$[\mathscr{B}\mathscr{H}_Y\mathscr{D}_Y p]\,(r,\phi) = -\frac{1}{\pi}\int_0^{\pi}\int_{-\infty}^{\infty}\frac{p_1(\ell,\theta)}{r\cos(\theta-\phi)-\ell}\,d\ell\,d\theta. \tag{6.15}$$

The identity, except for a multiplicative constant, of the right-hand sides of (6.9) and (6.15) can be concisely described by stating the operator equation:

$$\mathscr{R}^{-1} = -\frac{1}{2\pi}\mathscr{B}\mathscr{H}_Y\mathscr{D}_Y. \tag{6.16}$$

In words, the inverse Radon transform $\mathscr{R}^{-1}p$ of a function p of two variables can be obtained by the following sequence of operations:

(i) partial differentiate p with respect to its first variable to obtain a function q,
(ii) Hilbert transform q with respect to its first variable to obtain a function t,
(iii) backproject t, and
(iv) multiply the value of the resulting function by $-(1/2\pi)$. This is sometimes called *normalization*.

Such a process assumes that the exact values of $p(\ell,\theta)$ are known for all ℓ and θ and that the required operations can be carried out precisely. Neither of these assumptions is satisfied when we use a computer to estimate a function from its experimentally obtained projection data. Transform methods for image reconstruction are based on (6.16), or on alternative expressions for the inverse Radon transform $\mathscr{R}^{-1}$, but they have to perform on finite and imperfect data using the not unlimited capabilities of computers. How this is done is explained in the following chapters. The essence of what needs to be done is to find *numerical procedures* (i.e., ones that can be implemented on a digital computer), which estimate the value of a double integral, such as appears on the right-hand side of (6.9), from given values of $p(\ell_i,\theta_i)$, $1 \le i \le I$.

6.3 Series Expansion Methods

In the approach to the image reconstruction problem that is summarized in the preceding section, the techniques of mathematical analysis are used to find an inverse of the Radon transform. The inverse transform is described in terms of operators on functions defined over the whole continuum of real numbers. For implementation of the inverse Radon transform on a computer we have to replace these continuous operators by discrete ones that operate on functions with a finite number of arguments. This is done at the very end of the derivation of the reconstruction method.

The series expansion approach is basically different. The problem itself is discretized at the very beginning: estimating the function is translated into finding a finite set of numbers. This is done as follows.

For any specified picture region, we fix a set of J *basis functions* $\{b_1,\ldots,b_J\}$, each of which is a picture function with the specified picture region. These ought to be chosen so that, for any picture f with the specified picture region that we may wish to reconstruct, there exists a linear combination of the basis functions that we consider an adequate approximation to f.

An example of such an approach is the $n \times n$ digitization discussed in Section 4.1. In that case $J = n^2$. We number the pixels from 1 to J, and define

$$b_j(r,\phi) = \begin{cases} 1, & \text{if } (r,\phi) \text{ is inside the } j\text{th pixel,} \\ 0, & \text{otherwise.} \end{cases} \tag{6.17}$$

Then the $n \times n$ digitization of the picture f is the picture $\hat{f}$ defined by

$$\hat{f}(r,\phi) = \sum_{j=1}^{J} x_j b_j(r,\phi), \tag{6.18}$$

where x_j is the average value of f inside the jth pixel. A shorthand notation we use for equations of this type is $\hat{f} = \sum_{j=1}^{J} x_j b_j$. Note that since the values of f are linear attenuation coefficients that have dimensionality inverse length as shown in Section 15.1, the dimensionality of each x_j is inverse length, while the b_j are dimensionless.

There are other ways of choosing the basis functions; some of these are discussed later on. Once the basis functions are fixed, any picture $\hat{f}$ that can be represented as a linear combination of the basis functions b_j is uniquely determined by the choice of the coefficients x_j, $1 \le j \le J$, in the formula (6.18). We use x to denote the column vector whose jth component is x_j and refer to x as the *image vector*.

This approach restricts the general problem of "estimating a picture f" to the more specific problem of "finding an image vector x such that the $\hat{f}$ defined by (6.18) is as near to f as possible using the given basis functions." To make the notion of "nearness" precise, we use the definition (6.3) of distance between two picture functions.

It follows from standard results of mathematical analysis that, irrespective of how the basis functions are chosen, for any picture f there is one, and only one, picture $\hat{f}$ with the following properties:

(i) $\hat{f}$ is a linear combination of the basis functions,

(ii) if $\hat{\hat{f}}$ is a linear combination of the basis functions, then

$$d\left(f,\hat{f}\right) \le d\left(f,\hat{\hat{f}}\right). \tag{6.19}$$

Furthermore, if the basis functions are chosen so that they are *linearly independent* (i.e., none of them can be expressed as a linear combination of the others), then there is a unique image vector x that has the relationship expressed in (6.18) to this $\hat{f}$. For example, if the basis functions are defined by (6.17), then the $n \times n$ digitization of f is the $\hat{f}$ satisfying (i) and (ii), and the associated image vector x is unique.

Ideally, the series expansion approach should aim at finding the image vector that gives rise to the $\hat{f}$ nearest to f. However, since our data do not uniquely determine f, usually we try to find an image vector x that satisfies a less efficacious, but achievable, optimization criterion. Such criteria are discussed in the next section.

In order to show how the image reconstruction problem translates into a discrete problem using the series expansion approach we need to observe two properties of the functionals $\mathscr{R}_i$ defined by (6.8). The first property is that they are *linear*. This means that for all pictures f_1 and f_2, for all real numbers c_1 and c_2, and for $1 \leq i \leq I$,

$$\mathscr{R}_i \left(c_1 f_1 + c_2 f_2\right) = c_1 \mathscr{R}_i f_1 + c_2 \mathscr{R}_i f_2. \tag{6.20}$$

This is easily proved using the definitions of $\mathscr{R}_i$ and $\mathscr{R}$. The other property is mathematically less rigorous. We would like to be able to say that "if f_1 and f_2 are near each other, then so are $\mathscr{R}_i f_1$ and $\mathscr{R}_i f_2$." Unfortunately, using the distance for functions given in (6.3), a mathematically precise version of this statement would not be always true. Nevertheless, it is reasonable to argue, based on the definition of $\mathscr{R}_i$, that if $\hat{f}$ is defined so that the previously stated properties (i) and (ii) hold, then $\mathscr{R}_i \hat{f}$ will be approximately the same as $\mathscr{R}_i f$. This property is called *continuity*. A basic weakness of the series expansion approach is that this assumption is sometimes violated. Combining these properties we can state that, for $1 \leq i \leq I$,

$$\mathscr{R}_i f \simeq \mathscr{R}_i \hat{f} = \sum_{j=1}^{J} x_j \mathscr{R}_i b_j. \tag{6.21}$$

Since the b_j are user-defined functions, usually the $\mathscr{R}_i b_j$ can be easily calculated by analytical means. For example, in the case when the b_j are defined by (6.17), $\mathscr{R}_i b_j$ is just the length of intersection with the jth pixel of the line of the ith position of the source–detector pair. (More precisely, of the line at a distance ℓ_i from the origin making angle θ_i with the positive y axis; see (6.8) and Fig. 2.4. In this case, for any given i, a list of all the j such that $\mathscr{R}_i b_j \neq 0$ and the values of these $\mathscr{R}_i b_j$ can be efficiently calculated using a DDA; see Section 4.6.) When using alternate basis functions, it can happen that the ith line misses the picture region, but nevertheless $\mathscr{R}_i b_j \neq 0$; causing a violation of (6.21). It is strongly advisable to remove the measurements associated with such lines from the projection data sets, and we have done this in all the relevant experiments on which we report in this book. Unless otherwise stated, we use $r_{i,j}$ to denote our calculated value of $\mathscr{R}_i b_j$. Hence,

$$r_{i,j} \simeq \mathscr{R}_i b_j. \tag{6.22}$$

Recall also that we use y_i to denote the physically obtained estimate of $\mathscr{R}_i f$. Combining this with (6.21) and (6.22), we get that, for $1 \leq i \leq I$,

$$y_i \simeq \sum_{j=1}^{J} r_{i,j} x_j. \tag{6.23}$$

Note that in CT the $r_{i,j}$ have dimensionality length, since they are line integrals of a dimensionless function. Since the x_j have dimensionality inverse length, the right-hand side of (6.23) is dimensionless, as it should be to match its dimensionless left-hand side.

Let R denote the matrix whose (i, j)th element is $r_{i,j}$. We refer to this matrix as the *projection matrix*. Let e be the I-dimensional column vector whose ith component, e_i, is the difference between the left- and right-hand sides of (6.23). We refer to this as the *error vector*. Then (6.23) can be rewritten as

$$y = Rx + e. \tag{6.24}$$

The series expansion approach leads us to the following *discrete reconstruction problem*: based on (6.24),

given the data y, **estimate** the image vector x.

If the estimate that we find as our solution to the discrete reconstruction problem is the vector x^*, then the estimate f^* to the picture to be reconstructed is given by

$$f^* = \sum_{j=1}^{J} x_j^* b_j. \tag{6.25}$$

We make the following important observation. Our justification for the series expansion approach did *not* need that the functionals $\mathscr{R}_i$ be defined by (6.8). It only needed that the $\mathscr{R}_i$s satisfy the property expressed by (6.21). Many different ways of defining the $\mathscr{R}_i$s will have this property: integration along curved rather than straight lines or even areas (such as strips) rather than lines are potentially relevant to the general reconstruction problem. A major advantage of the series expansion methods over the transform methods is that they are immediately applicable to such more general ways of data collection.

6.4 Optimization Criteria

In this section we discuss optimization criteria by which the image vector of the series expansion approach is estimated. Although this will not be explicitly indicated, much of what we say is also relevant to estimating pictures using transform methods.

In (6.24), the vector e is unknown. The very most we can hope for is that we can specify a random variable of which e is a sample, and in most cases even this is impossible. The simple approach of trying to solve (6.24) by first assuming that e is the zero vector is dangerous: $y = Rx$ may have no solutions, or it may have many solutions, possibly none of which is any good for the practical problem at hand. Some criteria have to be developed, indicating which x ought to be chosen as a solution of (6.24).

The criteria that have been used for the reconstruction problem are usually of the form: choose as the "solution" of (6.24) an image vector x for which the value of some function $\phi_1(x)$ is minimal, and if there is more than one x that minimizes $\phi_1(x)$ choose among these one for which the value of some other function $\phi_2(x)$ is minimal. In this section we survey some of the choices for ϕ_1 and ϕ_2 that have been proposed.

A theoretically attractive approach is the following. Consider both the image vector x and the error vector e to be samples of random variables, denoted by X and E, respectively. Since our discussion in Section 1.2 was restricted to continuous random variables whose samples are real numbers, while here we deal with column vectors of real numbers, further explanation is needed. (A reader who is not desirous to learn about the foundations of Bayesian estimation may safely skip to (6.33).)

In fact, there is an additional subtle point that needs to be appreciated, especially because ignoring it can have some undesirable consequences. As discussed after (6.18), the dimensionality of the components of the image vector x is inverse length. As opposed to this, it follows from the discussion after (6.23) that the components of the error vector e are dimensionless. Because of this, any formulas involving samples from both X and E have to be formulated with the unit of length in mind.

The random variable X has an associated probability density function p_X, which is a real number valued function on J-dimensional vectors of real numbers (the possible samples of X). This function p_X is defined so that, for any J pairs (ℓ_j, u_j) of numbers such that $\ell_1 < u_1, \ldots, \ell_J < u_J$, the probability that a sample x of X will have the property that $\ell_j \leq x_j \leq u_j$, for $1 \leq j \leq J$, is

$$\int_{\ell_1}^{u_1} \cdots \int_{\ell_J}^{u_J} p_X(x)\, dx_J \cdots dx_1. \tag{6.26}$$

For notational convenience we sometimes abbreviate such integrals as

$$\int_{\ell}^{u} p_X(x)\, dx. \tag{6.27}$$

Since the probability expressed in (6.26) is dimensionless, it follows from the dimensionality of the J components x that $p_X(x)$ has to have dimensionality length to the Jth power.

Corresponding to the concepts of mean and variance of a continuous random variable as defined in (1.8) and (1.9), we have the concepts of *mean vector* μ_X and *covariance matrix* V_X, defined as

$$\mu_X = \int_{-\infty}^{\infty} x p_X(x)\, dx, \tag{6.28}$$

$$V_X = \int_{-\infty}^{\infty} (x - \mu_X)(x - \mu_X)^T p_X(x)\, dx, \tag{6.29}$$

where x^T denotes the row vector that is the *transpose* of the column vector x (i.e., a row vector whose ith component is x_i). These integrals are to be interpreted component by component. For example, using $(x - \mu_X)_i$ to denote the ith component of the vector $x - \mu_X$, the (i, j)th entry of V_X is given by

$$(V_X)_{i,j} = \int_{-\infty}^{\infty} (x - \mu_X)_i \, (x - \mu_X)_j \, p_X(x) \, dx. \tag{6.30}$$

It follows from these formulas that the dimensionality of the components of μ_X is inverse length, while the dimensionality of the entries of V_X is inverse length squared. Note that V_X is a *symmetric matrix* since it is clear from (6.30) that $(V_X)_{i,j} = (V_X)_{j,i}$.

The discussion in the previous two paragraphs has a simpler analog for the distribution E of the error vectors. In that case all numbers (the values of $p_E(e)$ and the components of μ_E and of V_E) are dimensionless. Similarly, the discussion of the next paragraph concerning X has a simpler dimensionless analog concerning E.

Let μ denote a J-dimensional vector of real numbers with dimensionality inverse length and let V denote a $J \times J$ symmetric matrix of real numbers with dimensionality inverse length squared. Let us further assume that V is *positive definite*, which means that $x^T V x$ is positive for any J-dimensional vector x with at least one nonzero component. Using elementary matrix algebra it can be shown that V has an inverse (denoted by V^{-1}) and its determinant (denoted by $\det V$) is positive. Furthermore, the dimensionality of the entries of V^{-1} is length squared and the dimensionality of $\det V$ is inverse length to the $2J$th power. Using such a μ and V, we can define a function p_X over the set of all J-dimensional vectors of real numbers of dimensionality inverse length by

$$p_X(x) = \frac{1}{(2\pi)^{J/2} (\det V)^{1/2}} \exp\left(-\frac{1}{2} (x - \mu)^T \, V^{-1} \, (x - \mu) \right). \tag{6.31}$$

It is not difficult to check that this p_X is a probability density function on the set of all J-dimensional vectors of real numbers of dimensionality inverse length and, using (6.28) and (6.29), that $\mu_x = \mu$ and $V_X = V$. A random variable X defined in such a fashion is called a *multivariate Gaussian random variable*. The probability density function of a multivariate Gaussian random variable peaks at its mean vector.

The importance of multivariate Gaussian random variables rests on two facts. One is that many random variables occurring in practice are approximately multivariate Gaussian. The other is that the assumption that an unknown random variable is multivariate Gaussian usually makes the mathematical treatment of the problem much easier than it would be otherwise.

Let us return now to the random variables X and E associated with x and e of (6.24). In this case p_X is referred to as the *prior probability density function*, since $p_X(x)$ indicates the likelihood of coming across an image vector similar to x. In CT it makes sense to adjust p to the area of the body we are imaging; the probabilities of the same picture representing a cross section of the head or of the thorax should be different. Our treating X and E separately is by itself a simplifying assumption, since in practice E is not independent of X, as can be seen from the discussion in Section 3.1. The theory that we are describing can be developed without making this assumption, but it becomes more complicated.

At last we are in position to state an optimization criterion (it assumes that p_X and p_E are known): given the data y, choose the image vector x for which the value of

$$p_E(y - Rx)p_X(x) \tag{6.32}$$

is as large as possible. Note that the second term in the product is large for vectors x that have large prior probabilities, while the first term is large for vectors x that are consistent with the data (at least if p_E peaks at the zero vector). The relative importance of the two terms depends on the nature of p_X and p_E. If p_X is flat (many image vectors are equally likely) and p_E is highly peaked near the zero vector, then our criterion will produce an image vector x^* that fits the measured data y in the sense that Rx^* will be nearly the same as y. On the other hand, if p_E is flat (large errors are nearly as likely as small ones) but p_X is highly peaked, our having made our measurements will have only a small effect on our preconceived idea as to how the image vector should be chosen. The x^* that maximizes (6.32) is called the *Bayesian estimate*.

A difficulty with using Bayesian estimation is that it presupposes knowledge of p_X and p_E. Precise knowledge of the true distributions of the image vector and of the error vector is usually not available. A second difficulty is that, for many p_X and p_E, the estimation of x that maximizes (6.32) may be far from trivial.

If we assume that both X and E are multivariate Gaussian, the optimization problem becomes much simpler. In that case it is easy to see from (6.31) that, assuming that μ_E is the zero vector, the x that maximizes (6.32) is the same x that minimizes

$$(y - Rx)^T V_E^{-1} (y - Rx) + (x - \mu_X)^T V_X^{-1} (x - \mu_X) . \tag{6.33}$$

Note that both terms in this sum are dimensionless.

A less sophisticated approach is to aim at finding a *least squares solution* of (6.24), i.e., an x that minimizes

$$\|e\|^2 = \|y - Rx\|^2 = \sum_{i=1}^{I} \left(y_i - \sum_{j=1}^{J} r_{i,j} x_j \right)^2 . \tag{6.34}$$

Such a criterion does not necessarily determine x; there may be more than one vector x that minimizes (6.34). In such a case one has to select an x by a second criterion, choices for which are described in the following.

Another reason why a least squares solution is not necessarily very good is that the criterion expressed in (6.34) does not contain any information regarding the nature of a "desirable" solution x. In the Bayesian approach of (6.33) such information is incorporated into the prior covariance matrix V_X.

It can be reasonably argued that a desirable property of the solution of (6.24) is that the variance

$$\sum_{j=1}^{J} (x_j - \bar{x})^2, \tag{6.35}$$

where

$$\bar{x} = \frac{1}{J}\left(\sum_{j=1}^{J} x_j\right) \tag{6.36}$$

should be small. If the basis functions are chosen according to (6.17), then $\bar{x}$ is the average density in the digitized picture. It can be shown that if $\bar{x}$ is considered fixed for all acceptable solutions to (6.24), then the x that minimizes (6.35) is the same x that minimizes the (*Euclidean*) *norm* $\|x\|$ of x, where

$$\|x\|^2 = \sum_{j=1}^{J} x_j^2. \tag{6.37}$$

In other words, in such a case the *minimum variance* and *minimum norm* solutions are the same.

The criteria expressed in (6.35) and (6.37) are not to be used as "primary" criteria in image reconstruction. That is, in terms of the notation introduced at the beginning of this section, it is not reasonable to define $\phi_1(x)$ by (6.35). That would lead to the "solution" in which all components of x are the same, namely $\bar{x}$. The use of (6.35) is either as a secondary criterion, or as a component of the primary criterion, where the other components force the "solution" to be consistent with the measurements, or express other properties of desirable solutions of (6.24).

For example, in the case when the basis functions are chosen according to (6.17) it may be considered "desirable" that the values x_j assigned to neighboring pixels should be close to one another on the average. Such a criterion can be expressed (see Section 12.3) by saying that we desire to minimize $x^{\mathrm{T}}Bx$, where B is an appropriately chosen matrix. This, in conjunction with the desire to minimize (6.34) and (6.37) at the same time, leads us to state that the sought solution x of (6.24) is the one that minimizes

$$a\|y - Rx\|^2 + x^T(bB + U)x, \tag{6.38}$$

where a and b are appropriately chosen positive numbers, indicating the relative importance we attach to minimizing the various expressions previously discussed, and U is the identity matrix. Here is where the potential for making a mistake by ignoring dimensionality lies. By stating that U is the identity matrix, we are implicitly assuming that its entries are dimensionless, otherwise changing units could turn U into a matrix other than the identity. Hence bB also has to be dimensionless and the dimensionality of the second term is inverse length squared. In order to keep the two terms of (6.38) physically consistent, we need to use an a that has dimensionality of inverse length squared. In other words, a cannot be a fixed number that is independent of the unit of length used. Also the expression in (6.38) (which has dimensionality of inverse length squared) is a different kind of thing from the expression in (6.33) (which is dimensionless), but this is a minor technical matter: by dividing both terms in (6.38) by the positive a, we get a dimensionless expression and an x^* minimizes this expression if, only if, it minimizes (6.38).

The approaches indicated by (6.33), (6.34), (6.35), (6.37), and (6.38) are special cases of a *quadratic optimization* problem that can be stated as follows. Find an x that minimizes

$$a\,(y-Rx)^T A\,(y-Rx) + (x-x_0)^T \left(bB + cC^{-1}\right)(x-x_0)\,, \tag{6.39}$$

where A is a symmetric $I \times I$ matrix, B and C are $J \times J$ matrices, a, b, and c are nonnegative real numbers, and x_0 is a J-dimensional vector. (Further details on the nature of these matrices, constants, and vectors are given in Section 12.1, which also contains the reasons for writing the matrix in the second term in the cumbersome form $bB + cC^{-1}$.) There may be more than one x that minimizes (6.39), in which case we need a second criterion for selecting one of them. As indicated in the last paragraph, in order to avoid making mistakes careful attention needs to be paid to the dimensionalities that occur in (6.39).

There are alternative ways of incorporating prior information about pictures of interest into the process of selecting a solution to (6.24). One example is to use the knowledge that x_j must lie within a certain range. In many applications, all pictures $f(r,\phi)$ that may occur have only nonnegative values. Then it is reasonable to demand that we accept an image vector x based on the digitization process of (6.17) as a solution to (6.24) only if $x_j \geq 0$, for $1 \leq j \leq J$. In fact, one may go further and demand also that for any solution of (6.24), the error should be within a certain bound, i.e., specify positive numbers $\varepsilon_1, \ldots, \varepsilon_I$, and accept as solutions only those x_js that have the property

$$-\varepsilon_i \leq y_i - \sum_{j=1}^{J} r_{i,j} x_j \leq \varepsilon_i, \tag{6.40}$$

for $1 \leq i \leq I$. Other inequality constraints may also be introduced.

Using such arguments, we can replace the system of equations (6.24) with the unspecified e and possibly with inequality side conditions, by a system of inequalities of the form

$$\sum_{j=1}^{J} n_{i,j} x_j \leq q_i, \tag{6.41}$$

which may be written in matrix notation as

$$Nx \leq q, \tag{6.42}$$

and restate the reconstruction problem as a search for an image vector x that satisfies (6.42). One must bear in mind here that there may be no x that satisfies all inequalities in (6.42), and if there is one such x, then usually there are many others as well. Just as in the case when there is more than one minimizing vector of (6.39), we need a secondary criterion to select one of these vectors as the desired solution. There have been several secondary optimization criteria proposed in the reconstruction literature.

One of these is based on the minimization of the norm $\|x\|$, which we already discussed above. More generally, a unique solution will be ensured, if among all the image vectors that satisfy the primary criterion we choose the one that minimizes

$$\left\|D^{-1}x\right\|, \tag{6.43}$$

where D is a positive definite symmetric $J \times J$ matrix. (Recall that this implies that $x^{\mathrm{T}}Dx > 0$ for all nonzero vectors x.) As discussed below, some reconstruction techniques minimize (6.43) for various Ds.

An alternative secondary criterion is applicable if the average value $\bar{x}$ of the x_js is known. In such a case there is at most one vector x for which $x_j \geq 0$, for $1 \leq j \leq J$, whose average value is $\bar{x}$ and that maximizes

$$-\sum_{j=1}^{J} (x_j/J\bar{x}) \ln (x_j/J\bar{x}). \tag{6.44}$$

This has been referred to as the *maximum entropy* criterion. The use of this criterion is usually justified by arguments (which are too long to be reproduced here) aimed at showing that of all the pictures that satisfy the primary criterion the maximum entropy solution has the smallest information content, and so it is least likely to mislead the user by the presence of spurious features.

The reason why one may assume that $\bar{x}$ is known is the following. Consider Fig. 2.4. For any source–detector pair, the ray sum divided by the length of intersection of the line with the picture region (reconstruction region) gives an estimate of the average relative linear attenuation for that line. If we have many such lines that provide a fairly uniform and dense covering of the reconstruction region, then the sum of all the ray sums divided by the sum of the lengths of intersections is a reasonable estimate of $\bar{x}$. For example, for our standard head phantom $\bar{x} = 0.1315$. The estimate of $\bar{x}$ obtained from the standard projection data (Section 5.8) by the method described above is 0.1307. This is in spite of the fact that the standard projection data are contaminated with errors due to photon statistics, beam hardening, scatter, etc. Similarly, the estimate of $\bar{x}$ obtained from the standard parallel projection data is 0.1312. Such experiments justify the use of the method described above for the estimation of $\bar{x}$ in conjunction with optimization criteria, such as maximum entropy or minimum variance.

A third secondary criterion that has been gaining popularity in recent years is *total variation (TV) minimization.* We restrict our discussion of this to a special case in which the basis functions are chosen according to (6.17). Let T denote the set of all indices of pixels that are not in the rightmost column or in the bottom row of the $n \times n$ digitization and, for any pixel with index i in T, let $r(i)$ and $b(i)$ denote the index of the pixel to its right and below it, respectively. Then the *total variation* of the image vector x is defined as

$$TV(x) = \sum_{i \in T} \sqrt{\left(x_{r(i)} - x_i\right)^2 + \left(x_{b(i)} - x_i\right)^2}. \tag{6.45}$$

A widely studied optimization criterion in the field of image reconstruction from projections is provided by the concept of *maximum likelihood estimation.* This is a quite general concept that can be described in our context as follows. Assume that we have a statistical model that provides us, for any image vector x, with a probability density function p_Y^x of the multivariate random variable Y associated with the process that generates the measurement vector y. In practice we choose such a model based on our understanding of the nature of our application and how the data are collected in that application. For example, if we already know the probability density function p_E associated with the error vector e that we discussed earlier, then we can define

$$p_Y^x(y) = p_E(y - Rx). \tag{6.46}$$

Then, having observed the measurement vector y, a *maximum likelihood estimate* of the image vector is an x that maximizes $p_Y^x(y)$. (Note that the name of this estimator has an unjustified positive connotation: a maximum likelihood estimator x is not really a "most likely" one, but rather it is the case that among all possible image vectors there are none for which the likelihood of observing y is greater than the likelihood of observing y when x is the image vector.) Comparing (6.46) with (6.32) that is used to define the Bayesian estimate, we see that the essential difference is that the formula for the maximum likelihood estimate does not make use of an assumed prior probability density function p_X for the distribution of the image vectors.

Such an approach is likely to be useful in applications in which the nature of p_Y^x is reasonably well understood, but there is uncertainty regarding the nature of p_X. For example, in positron emission tomography (see Section 1.1), one may assume that, for $1 \le i \le I$, y_i is a sample from the Poisson random variable with parameter $\sum_{j=1}^{J} r_{i,j} x_j$ and that these I samples are independent. Under these assumptions it follows from (3.1) that

$$p_Y^x(y) = \prod_{i=1}^{I} \frac{\left(\sum_{j=1}^{J} r_{i,j} x_j\right)^{y_i} \exp\left(-\sum_{j=1}^{J} r_{i,j} x_j\right)}{y_i!}. \tag{6.47}$$

Since the natural logarithm is a monotonically increasing function, finding the x that maximizes this $p_Y^x(y)$ is the same as finding the x that minimizes

$$\sum_{i=1}^{I}\left(\left(\sum_{j=1}^{J} r_{i,j}x_j\right) - y_i \ln\left(\sum_{j=1}^{J} r_{i,j}x_j\right)\right). \tag{6.48}$$

In practice it has been found that the image vector that minimizes (6.48) is often very noisy looking, as if some salt-and-pepper type of noise had been superimposed on what is basically a good reconstruction. To counteract this, the criterion is often *regularized*, for example, by replacing it with

$$\sum_{i=1}^{I}\left(\left(\sum_{j=1}^{J} r_{i,j}x_j\right) - y_i \ln\left(\sum_{j=1}^{J} r_{i,j}x_j\right)\right) + bx^T Bx, \tag{6.49}$$

where B is the already mentioned smoothing matrix (compare this with (6.38) and see also Section 12.3). We could have derived the same formula using Bayesian estimation based on (6.32), combined with (6.46) and (6.47) and using a multivariate Gaussian p_X with μ_X the zero vector and $V_X^{-1} = bB$; compare with (6.33).

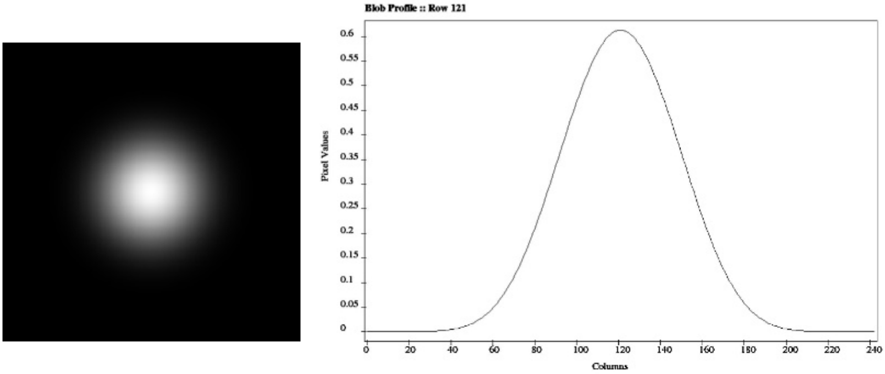
6.5 Blob Basis Functions

Generalized Kaiser–Bessel window functions, which are also known by the simpler name *blobs*, form a large family of functions that can be defined in a Euclidean space of any dimension. Here we restrict ourselves to a subfamily in the two-dimensional plane, whose elements have the form

$$b_{a,\alpha,\delta}(r,\phi) = \begin{cases} C_{a,\alpha,\delta}\left(1-\left(\frac{r}{a}\right)^2\right) I_2\left(\alpha\sqrt{1-\left(\frac{r}{a}\right)^2}\right), & \text{if } 0 \le r \le a, \\ 0, & \text{otherwise,} \end{cases} \tag{6.50}$$

where I_k denotes the modified Bessel function of the first kind of order k, a stands for the nonnegative radius of the blob and α is a nonnegative real number that controls the blob's taper (the shape of the blob). The multiplying constant $C_{a,\alpha,\delta}$ is defined below. Note that such a blob is circularly symmetric, since its value does not depend on ϕ. It has the value zero for all $r \ge a$ and its first derivatives are continuous everywhere. In this sense (and in a deeper mathematical sense that we do not detail here) blobs are very "smooth" functions, see Fig. 6.3. Their smoothness can be controlled by the choice of the parameters a, α and δ, as we demonstrate shortly.

For now let us consider the parameters a, α and δ, and hence the function $b_{a,\alpha,\delta}$, to be fixed. This fixed function gives rise to a set of J basis functions $\{b_1, \ldots, b_J\}$ as follows. We define a set $G = \{g_1, \ldots, g_J\}$ of *grid points* in the



(a) (b)

Fig. 6.3: (a) A 243 × 243 digitization of a blob. (b) Its values on the central row.

picture region. Then, for $1 \leq j \leq J$, b_j is obtained from $b_{a,\alpha,\delta}$ by shifting it in the plane so that its center is moved from the origin to g_j. This definition leaves a great deal of freedom in the selection of G, but it was found in practice advisable that it should consists of those points of a set (in rectangular coordinates)

$$G_\delta = \left\{ \left(\frac{m\delta}{2}, \frac{\sqrt{3}n\delta}{2} \right) \middle| \, m \text{ and } n \text{ are integers and } m+n \text{ is even} \right\} \quad (6.51)$$

that are also in the picture region. Here δ has to be a positive real number and G_δ is referred to as the *hexagonal grid with sampling distance* δ. Having fixed δ, we complete the definition in (6.50) by

$$C_{a,\alpha,\delta} = \frac{\sqrt{3}\delta^2\alpha}{4\pi a^2 I_3(\alpha)}. \quad (6.52)$$

The Radon transform (6.4) maps a picture into its line integrals. Its inversion in practice tends to amplify errors in the measured data. One way of reducing this is to seek a smoothed version of the theoretical solution. This is often done by a regularization term (see, for example, (6.49)), but it can be also tackled by using smooth basis functions.

Pixel-based basis functions (6.17) have a unit value inside the pixels and zero outside. Blobs on the other hand, have a bell-shaped profile that tapers smoothly in the radial direction from a high value at the center to the value 0 at the edge of their supports (i.e., at $r = a$ in (6.50)); see Fig. 6.3. The smoothness of blobs suggests that reconstructions of the form (6.18) are likely to be resistant to noise in the data. This has been shown to be particularly useful in fields in which the projection data are noisy, such as positron emission tomography and electron microscopy, which were discussed in Section 1.1.

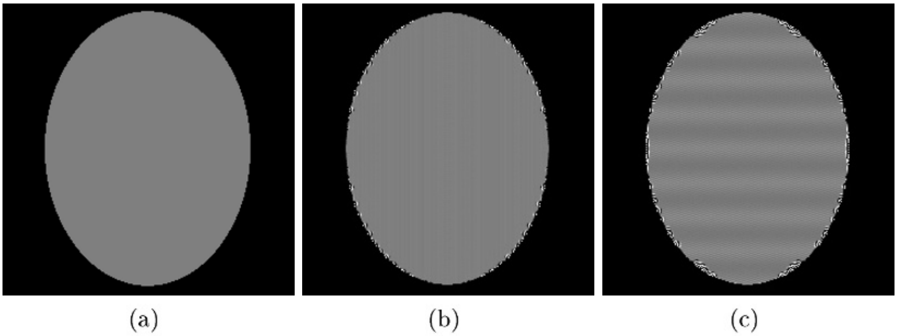


Fig. 6.4: (a) A 243×243 digitization of a solid bone "head" cross section. (b) Its approximation with default blob parameters and (c) with slightly different parameters. The display window is extremely narrow for better indication of errors.

For blobs to achieve their full potential, the selection of the parameters a, α and δ is important. The mathematical analysis of how they should be chosen is beyond the scope of this book. In SNARK09, the software automatically calculates good ***default blob parameters*** based on the geometry of the digitized picture that is being produced as output. For example, for the 243×243 digitizations used in this book, the default values (to four decimal place accuracy) are $a = 0.1551$, $\alpha = 11.2829$ and $\delta = 0.0868$. Using these default parameters, one can approximate homogeneous regions very well, in spite of the bell-shaped profile of the individual blobs. This is illustrated in Fig. 6.4(b), in which a cross section through solid bone shown in Fig. 6.4(a) is approximated by a linear combination of the blob basis functions with the default parameters. There are some inaccuracies very near the sharp edges, but the interior of the bone is approximated with great accuracy. On the other hand, if we change the parameters ever so slightly to $a = 0.16$, $\alpha = 11.28$ and $\delta = 0.09$, then the best approximation that can be obtained by a linear combination of the blob basis functions is shown in Fig. 6.4(c), which is clearly inferior.

Based on the mathematical formulas (6.50), (6.51) and (6.52) one can calculate, for any line i and for any blob basis function b_j, the value of $r_{i,j} \simeq \mathscr{R}_i b_j = [\mathscr{R} f]\,(\ell_i, \theta_i)$. In fact, because blobs are circularly symmetric, these integrals are not dependent on the orientation θ_i of the line of integration but only on the distance of the line from the center g_j of the basis function b_j. In practice, for any fixed blob parameters, the values of the integrals as a function of distance from the blob center can be precalculated and stored on the computer and so, during any particular reconstruction using such blobs, the computation of the integral is efficiently achieved by the retrieval of a precalculated value. This combined with a DDA-like mechanism that indicates which blobs may possibly be intersected by the given line, allows one to calculate rapidly the right-hand side of (6.23) for any given image vector x.

6.6 Computational Efficiency

In the succeeding chapters we show reconstructions of our head phantom from the standard projection data (or from the standard parallel projection data) using many different methods. We also show plots of the 131st column and give picture distance measures defined in Section 5.1 and statistical performance comparisons of the kind discussed in Section 5.2.

In addition, we indicate the cost of the reconstruction in terms of computer time. All the algorithms are implemented in the SNARK09 programming system (see Chapter 4), and the times reported are the number of seconds when using a computer with an AMD Athlon™ 64 Processor 3500+, 2.2 GHz, 1GB DDR Memory, running Linux Fedora 9.

While these timings are given for the sake of completeness, they are not to be taken too seriously. A general framework of computer programs containing many different algorithms, such as SNARK09, is by necessity not as efficient for any single algorithm as a program specially written for that purpose. Thus the absolute, and even the relative, values of computer times quoted below may be misleading. Implementations of algorithms used in actual CT scanners usually involve low (i.e., assembly or machine) level programming and even special-purpose hardware, making the execution of reconstructions orders of magnitude faster than what is possible using SNARK09. (The reason for using SNARK09 is ease of implementation; it would be quite beyond the capability of an individual to implement all algorithms to be reported on in this book by special-purpose programming.)

This attitude towards timing reflects the fact that electronic hardware used for calculations is getting cheaper and cheaper at an amazing rate. It is unlikely that an efficacious reconstruction algorithm would for long remain unused solely because of computational considerations.

Notes and References

Much of the material in this chapter is based on a survey paper on iterative reconstruction algorithms [127]. That paper contains discussions of and references to many earlier publications concerning reconstruction algorithms based on the series expansion approach and optimization criteria. There are many more recent texts discussing reconstruction algorithms; a good treatment from a more mathematical point of view is given in [211].

A good coverage of Lebesgue integrals and square integrable functions, operators, and linear functionals is given by [161].

Our treatment of the inverse Radon transform adapted the approach and notation of [234]. A thorough mathematical discussion of Hilbert transforms can be found in [37]. References to literature on derivations of the Radon inversion formula without assuming properties such as differentiability are given at the end of Chapter 15.

Our treatment of multivariate random variables is based on [235]. That book also contains a discussion of Bayes' theorem, which provides the mathematical justification for the use of the Bayesian estimate. The equivalence of the minimum norm and minimum variance criteria is shown in [130].

The maximum entropy formalism is a general scientific approach; there are whole books devoted to the subject; see, e.g., [180]. The suggestion that it be used for image reconstruction first appeared in the open literature in [99]. It has been extensively used in the related field of digital image restoration; see, e.g., [8]. As examples of works on the computation of maximum entropy solutions, see [74] and [203]. Total variation minimization has become something of a fad at the time of writing this edition; for a critical discussion with background references see [121]. TV minimization has been applied in a variety of fields, for an example in IMRT see [280].

The maximum likelihood formalism was introduced to the image reconstruction community by L.A. Shepp and A. Vardi [242]. The idea of combining the likelihood function with one that expresses assumed prior knowledge about the space of pictures that we are likely to come across in a reconstruction application was presented in [181]. Original implementations of such approaches tended to be slow, many faster variants have been developed over the years, for example, in [34, 122, 147]. The last of these references seems to have found great popularity in the emission tomography community. It achieves its efficiency by using essentially the same idea that was proposed much earlier in [73] for finding the minimizer of (6.39): divide the system of equations, such as (6.23) or (6.24) into subsets (also called blocks) and get an overall solution by repeatedly cycling through the blocks, one at a time. A recent interesting application of the maximum likelihood formalism to image reconstruction from projections is reported in [237]: several conformations of a molecule are simultaneously reconstructed from a heterogeneous mixture of their electron microscopic projections taken at unknown orientations.

There are many additional optimization criteria proposed in the literature include, for instance, maximum signal-to-noise power ratio [255].

Generalized Kaiser–Bessel window functions (blobs) were first proposed for image reconstruction by R.M. Lewitt [183, 184]. They are also applicable, if anything more significantly so, in the reconstruction of 3D objects from 2D projections. The blob basis functions have proved to be more suitable than the pixel basis functions (in 2D) and the voxel basis functions (in 3D). It has been shown that the use of blobs as basis functions can produce superior results for different types of applications, such as positron emission tomography [160, 198] and electron microscopy [194, 195]. The choice of the hexagonal grid and the selection of the default blob parameters is justified by the material in [199, 200].

Optimization using parallel (and hence fast) computations is discussed in [48] with special reference to series expansion reconstruction methods. A recent development along this line is [80]. For methods of using standard hardware to speed up reconstruction algorithms, see [206, 274] and their references.