7 Algebraic Reconstruction Algorithms

An entirely different approach for tomographic imaging consists of assuming that the cross section consists of an array of unknowns, and then setting up algebraic equations for the unknowns in terms of the measured projection data. Although conceptually this approach is much simpler than the transform-based methods discussed in previous sections, for medical applications it lacks the accuracy and the speed of implementation. However, there are situations where it is not possible to measure a large number of projections, or the projections are not uniformly distributed over 180 or 360°, both these conditions being necessary requirements for the transform-based techniques to produce results with the accuracy desired in medical imaging. An example of such a situation is earth resources imaging using cross-borehole measurements discussed in Chapter 4. Problems of this type are sometimes more amenable to solution by algebraic techniques. Algebraic techniques are also useful when the energy propagation paths between the source and receiver positions are subject to ray bending on account of refraction, or when the energy propagation undergoes attenuation along ray paths as in emission CT. [Unfortunately, many imaging problems where refraction is encountered also suffer from diffraction effects (see Chap. 4).] As will be obvious from the discussion to follow, in algebraic methods it is essential to know ray paths that connect the corresponding transmitter and receiver positions. When refraction and diffraction effects are substantial (medium inhomogeneities exceed 10% of the average background value and the correlation length of these inhomogeneities is comparable to a wavelength), it becomes impossible to predict these ray paths. If algebraic techniques are applied under these conditions, we often obtain meaningless results.

If the refraction and diffraction effects are small (medium inhomogeneities are less than 2 to 3% of the average background value and the correlation width of these inhomogeneities is much greater than a wavelength), in some cases it is possible to combine algebraic techniques with digital ray tracing techniques [And82], [And84a], [And84b] and devise iterative procedures in which we first construct an image ignoring refraction, then trace rays connecting the corresponding transmitter and receiver locations through this distribution, and finally use these rays to construct a more accurate set of

algebraic equations. Experimental verification of this iterative procedure for weakly refracting objects has been obtained [And84b].

Space limitations prevent us from discussing here the combined ray tracing and algebraic reconstruction algorithms. Our aim in this section is to merely introduce the reader to the algebraic approach for image reconstruction. First we will show how we may construct a set of linear equations whose unknowns are elements of the object cross section. The Kaczmarz method for solving these equations will then be presented. This will be followed by the various approximations that are used in this method to speed up its computer implementation.

7.1 Image and Projection Representation

Fig. 7.1: *In algebraic methods a square grid is superimposed over the unknown image. Image values are assumed to be constant within each cell of the grid. (From [Ros82].)*

In Fig. 7.1 we have superimposed a square grid on the image $f(x, y)$; we will assume that in each cell the function $f(x, y)$ is constant. Let f_j denote this constant value in the jth cell, and let N be the total number of cells. For algebraic techniques a ray is defined somewhat differently. A ray is now a "fat" line running through the (x, y)-plane. To illustrate this we have shaded the ith ray in Fig. 7.1, where each ray is of width τ. In most cases the ray width is approximately equal to the image cell width. A line integral will now be called a *ray-sum*.

Like the image, the projections will also be given a one-index representa-

$$w_{ji} \text{ for this cell} = \frac{\text{area of ABC}}{\delta^2}$$
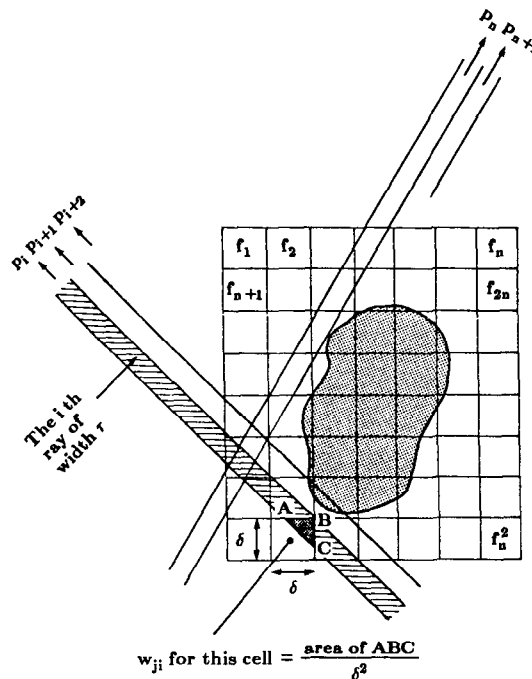
tion. Let p_i be the ray-sum measured with the ith ray as shown in Fig. 7.1. The relationship between the f_j's and p_i's may be expressed as

$$\sum_{j=1}^{N} w_{ij} f_j = p_i, \qquad i = 1, 2, \cdots, M \qquad (1)$$

where M is the total number of rays (in all the projections) and w_{ij} is the weighting factor that represents the contribution of the jth cell to the ith ray integral. The factor w_{ij} is equal to the fractional area of the jth image cell intercepted by the ith ray as shown for one of the cells in Fig. 7.1. Note that most of the w_{ij}'s are zero since only a small number of cells contribute to any given ray-sum.

If M and N were small, we could use conventional matrix theory methods to invert the system of equations in (1). However, in practice N may be as large as 65,000 (for 256 × 256 images), and, in most cases for images of this size, M will also have the same magnitude. For these values of M and N the size of the matrix $[w_{ij}]$ in (1) is 65,000 × 65,000 which precludes any possibility of direct matrix inversion. Of course, when noise is present in the measurement data and when $M < N$, even for small N it is not possible to use direct matrix inversion, and some least squares method may have to be used. When both M and N are large, such methods are also computationally impractical.

For large values of M and N there exist very attractive iterative methods for solving (1). These are based on the "method of projections" as first proposed by Kaczmarz [Kac37], and later elucidated further by Tanabe [Tan71]. To explain the computational steps involved in these methods, we first write (1) in an expanded form:

$$w_{11}f_1 + w_{12}f_2 + w_{13}f_3 + \cdots + w_{1N}f_N = p_1$$
$$w_{21}f_1 + w_{22}f_2 + \qquad + \cdots + w_{2N}f_N = p_2$$
$$\vdots$$
$$w_{M1}f_1 + w_{M2}f_2 + \qquad + \cdots + w_{MN}f_N = p_M. \qquad (2)$$

A grid representation with N cells gives an image N degrees of freedom. Therefore, an image, represented by $(f_1, f_2, \cdots, f_N)$, may be considered to be a single point in an N-dimensional space. In this space each of the above equations represents a hyperplane. When a unique solution to these equations exists, the intersection of all these hyperplanes is a single point giving that solution. This concept is further illustrated in Fig. 7.2 where, for the purpose of display, we have considered the case of only two variables f_1 and f_2 satisfying the following equations:

$$w_{11}f_1 + w_{12}f_2 = p_1$$
$$w_{21}f_1 + w_{22}f_2 = p_2. \qquad (3)$$

The figure shows two lines intersecting. The line labeled $w_{21}f_1 + w_{22}f_2 = p_2$ and the line labeled $w_{11}f_1 + w_{12}f_2 = p_1$ are drawn in the f_1–f_2 coordinate plane. Points $\vec{f}^{(0)}$, $\vec{f}^{(1)}$, $\vec{f}^{(2)}$, G, and H (initial guess) are indicated.
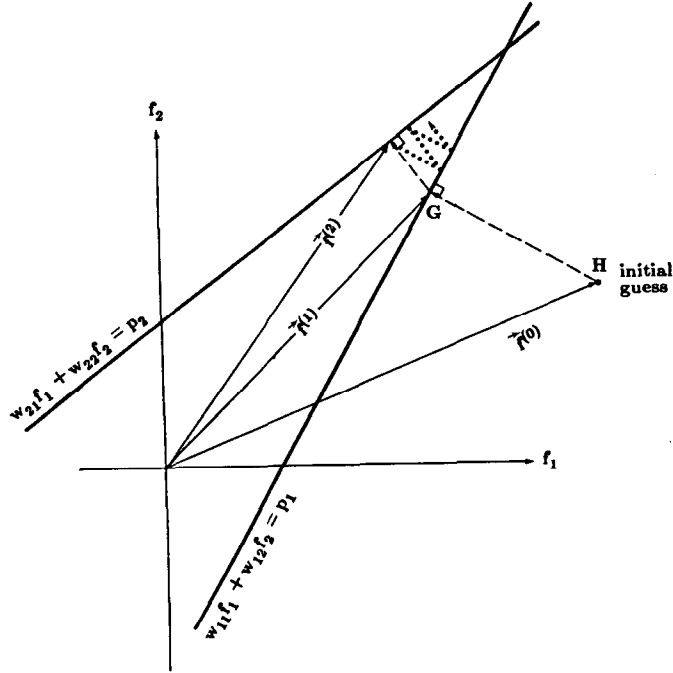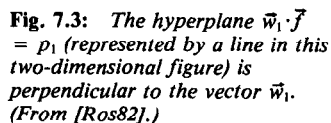
Fig. 7.2: *The Kaczmarz method of solving algebraic equations is illustrated for the case of two unknowns. One starts with some arbitrary initial guess and then projects onto the line corresponding to the first equation. The resulting point is now projected onto the line representing the second equation. If there are only two equations, this process is continued back and forth, as illustrated by the dots in the figure, until convergence is achieved. (From [Ros82].)*

The computational procedure for locating the solution in Fig. 7.2 consists of first starting with an initial guess, projecting this initial guess on the first line, reprojecting the resulting point on the second line, and then projecting back onto the first line, and so forth. If a unique solution exists, the iterations will always converge to that point.

For the computer implementation of this method, we first make an initial guess at the solution. This guess, denoted by $f_1^{(0)}, f_2^{(0)}, \cdots, f_N^{(0)}$, is represented vectorially by $\vec{f}^{(0)}$ in the N-dimensional space. In most cases, we simply assign a value of zero to all the f_i's. This initial guess is projected on the hyperplane represented by the first equation in (2) giving $\vec{f}^{(1)}$, as illustrated in Fig. 7.2 for the two-dimensional case. $\vec{f}^{(1)}$ is projected on the hyperplane represented by the second equation in (2) to yield $\vec{f}^{(2)}$ and so on. When $\vec{f}^{(i-1)}$ is projected on the hyperplane represented by the ith equation to yield $\vec{f}^{(i)}$, the process can be mathematically described by

$$\vec{f}^{(i)} = \vec{f}^{(i-1)} - \frac{(\vec{f}^{(i-1)} \cdot \vec{w}_i - p_i)}{\vec{w}_i \cdot \vec{w}_i} \vec{w}_i \qquad (4)$$

where $\vec{w}_i = (w_{i1}, w_{i2}, \cdots, w_{iN})$, and $\vec{w}_i \cdot \vec{w}_i$ is the dot product of $\vec{w}_i$ with itself. To see how (4) comes about we first write the first equation of (2) as

Fig. 7.3: *The hyperplane $\vec{w}_1 \cdot \vec{f}$ = p_1 (represented by a line in this two-dimensional figure) is perpendicular to the vector $\vec{w}_1$. (From [Ros82].)*

follows:

$$\vec{w}_1 \cdot \vec{f} = p_1. \tag{5}$$

The hyperplane represented by this equation is perpendicular to the vector $\vec{w}_1$. This is illustrated in Fig. 7.3, where the vector $\overrightarrow{OD}$ represents $\vec{w}_1$. This equation simply says that the projection of a vector $\overrightarrow{OC}$ (for any point C on the hyperplane) on the vector $\vec{w}_1$ is of constant length. The unit vector $\overrightarrow{OU}$ along $\vec{w}_1$ is given by

$$\overrightarrow{OU} = \frac{\vec{w}_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} \tag{6}$$

and the perpendicular distance of the hyperplane from the origin, which is

equal to the length of $\overline{OA}$ in Fig. 7.3, is given by $\overrightarrow{OC} \cdot \overrightarrow{OU}$:

$$|\overrightarrow{OA}| = \overrightarrow{OU} \cdot \overrightarrow{OC} = \frac{1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} (\vec{w}_1 \cdot \overrightarrow{OC})$$

$$= \frac{1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} (\vec{w}_1 \cdot \vec{f}) = \frac{p_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} . \tag{7}$$

Now to get $\vec{f}^{(1)}$ we have to subtract from $\vec{f}^{(0)}$ the vector $\overrightarrow{HG}$:

$$\vec{f}^{(1)} = \vec{f}^{(0)} - \overrightarrow{HG} \tag{8}$$

where the length of the vector $\overrightarrow{HG}$ is given by

$$|\overrightarrow{HG}| = |\overrightarrow{OF}| - |\overrightarrow{OA}|$$
$$= \vec{f}^{(0)} \cdot \overrightarrow{OU} - |\overrightarrow{OA}|. \tag{9}$$

Substituting (6) and (7) in this equation, we get

$$|\overrightarrow{HG}| = \frac{\vec{f}^{(0)} \cdot \vec{w}_1 - p_1}{\sqrt{\vec{w}_1 \cdot \vec{w}_1}} . \tag{10}$$

Since the direction of $\overrightarrow{HG}$ is the same as that of the unit vector $\overrightarrow{OU}$, we can write

$$\overrightarrow{HG} = |\overrightarrow{HG}| \overrightarrow{OU} = \frac{\vec{f}^{(0)} \cdot \vec{w}_1 - p_1}{\vec{w}_1 \cdot \vec{w}_1} \vec{w}_1. \tag{11}$$

Substituting (11) in (8), we get (4).

As mentioned before, the computational procedure for algebraic reconstruction consists of starting with an initial guess for the solution, taking successive projections on the hyperplanes represented by the equations in (2), eventually yielding $\vec{f}^{(M)}$. In the next iteration, $\vec{f}^{(M)}$ is projected on the hyperplane represented by the first equation in (2), and then successively onto the rest of the hyperplanes in (2), to yield $\vec{f}^{(2M)}$, and so on. Tanabe [Tan71] has shown that if there exists a unique solution $\vec{f}_s$ to the system of equations (2), then

$$\lim_{k \to \infty} \vec{f}^{(kM)} = \vec{f}_s. \tag{12}$$

A few comments about the convergence of the algorithm are in order here. If in Fig. 7.2 the two hyperplanes are perpendicular to each other, the reader may easily show that given for an initial guess any point in the (f_1, f_2)-plane, it is possible to arrive at the correct solution in only two steps like (4). On the other hand, if the two hyperplanes have only a very small angle between them, k in (12) may acquire a large value (depending upon the initial guess) before the correct solution is reached. Clearly the angles between the

hyperplanes considerably influence the rate of convergence to the solution. If the M hyperplanes in (2) could be made orthogonal with respect to one another, the correct solution would be arrived at with only one pass through the M equations (assuming a unique solution does exist). Although theoretically such orthogonalization is possible using, for example, the Gram–Schmidt procedure, in practice it is computationally not feasible. Full orthogonalization will also tend to enhance the effects of the ever present measurement noise in the final solution. Ramakrishnan *et al.* [Ram79] have suggested a pairwise orthogonalization scheme which is computationally easier to implement and at the same time considerably increases the speed of convergence. A simpler technique, first proposed in [Hou72] and studied in [Sla85], is to carefully choose the order in which the hyperplanes are considered. Since each hyperplane represents a distinct ray integral, it is quite likely that adjacent ray integrals (and thus hyperplanes) will be nearly parallel. By choosing hyperplanes representing widely separated ray integrals, it is possible to improve the rate of convergence of the Kaczmarz approach.

A not uncommon situation in image reconstruction is that of an overdetermined system in the presence of measurement noise. That is, we may have $M > N$ in (2) and $p_1, p_2, \cdots, p_m$ corrupted by noise. No unique solution exists in this case. In Fig. 7.4 we have shown a two-variable system represented by three ''noisy'' hyperplanes. The broken line represents the course of the solution as we successively implement (4). Now the ''solution'' doesn't converge to a unique point, but will oscillate in the neighborhood of the intersections of the hyperplanes.

When $M < N$ a unique solution of the set of linear equations in (2) doesn't exist, and, in fact, an infinite number of solutions are possible. For example, suppose we have only the first of the two equations in (3) to use for calculating the two unknowns f_1 and f_2; then the solution can be anywhere on the line corresponding to this equation. Given the initial guess $\vec{f}^{(0)}$ (see Fig. 7.3), the best one could probably do under the circumstances would be to draw a projection from $\vec{f}^{(0)}$ on this line, and call the resulting $\vec{f}^{(1)}$ a solution. Note that the solution obtained in this manner corresponds to that point on the line which is closest to the initial guess. This result has been rigorously proved by Tanabe [Tan71] who has shown that when $M < N$, the iterative approach described above converges to a solution, call it $\vec{f}'_s$, such that $|\vec{f}^{(0)} - \vec{f}'_s|$ is minimized.

Besides its computational efficiency, another attractive feature of the iterative approach presented here is that it is now possible to incorporate into the solution some types of a priori information about the image one is reconstructing. For example, if it is known a priori that the image $f(x, y)$ is nonnegative, then in each of the solutions $\vec{f}^{(k)}$, successively obtained by using (4), one may set the negative components equal to zero. One may similarly incorporate the information that $f(x, y)$ is zero outside a certain area, if this is known.
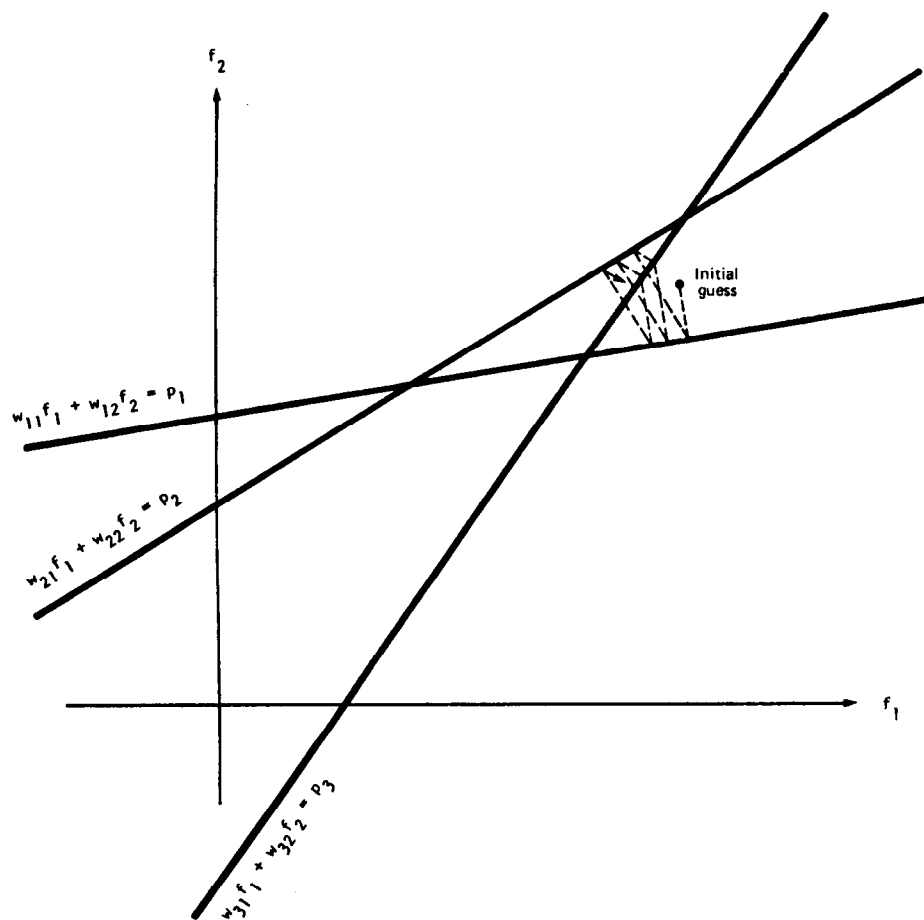
Fig. 7.4: *Illustrated here is the case when the number of equations is greater than the number of unknowns. The lines don't intersect at a single unique point, because the observations p_1, p_2, p_3 have been assumed to be corrupted by noise. No unique solution exists in this case, and the final solution will oscillate in the neighborhood of intersections of the three lines. (From [Ros82].)*

In applications requiring a large number of views and where large-sized reconstructions are made, the difficulty with using (4) can be in the calculation, storage, and fast retrieval of the weight coefficients w_{ij}. Consider the case where we wish to reconstruct an image on a 100×100 grid from 100 projections with 150 rays in each projection. The total number of weights, w_{ij}, needed in this case is 10^8, which is an enormous number and can pose problems in fast storage and retrieval in applications where reconstruction speed is important. This problem is somewhat eased by making approximations, such as considering w_{ij} to be only a function of the perpendicular distance between the center of the ith ray and the center of the jth cell. This perpendicular distance can then be computed at run time.

To get around the implementation difficulties caused by the weight coefficients, a myriad of other algebraic approaches have also been suggested, many of which are approximations to (4). To discuss some of the more implementable approximations, we first recast (4) in a slightly different

form:

$$f_j^{(i)} = f_j^{(i-1)} + \frac{p_i - q_i}{\sum\limits_{k=1}^{N} w_{ik}^2} \, w_{ij} \tag{13}$$

where

$$q_i = \vec{f}^{(i-1)} \cdot \vec{w}_i \tag{14}$$

$$= \sum_{k=1}^{N} f_k^{(i-1)} w_{ik}. \tag{15}$$

These equations say that when we project the $(i - 1)$th solution onto the ith hyperplane [ith equation in (2)] the gray level of the jth element, whose current value is $f_j^{(i-1)}$, is obtained by correcting its current value by $\Delta f_j^{(i)}$, where

$$\Delta f_j^{(i)} = f_j^{(i)} - f_j^{(i-1)} = \frac{p_i - q_i}{\sum\limits_{k=1}^{N} w_{ik}^2} \, w_{ij}. \tag{16}$$

Note that while p_i is the measured ray-sum along the ith ray, q_i may be considered to be the computed ray-sum for the same ray based on the $(i - 1)$th solution for the image gray levels. The correction Δf_j to the jth cell is obtained by first calculating the difference between the measured ray-sum and the computed ray-sum, normalizing this difference by $\Sigma_{k=1}^{N} w_{ik}^2$, and then assigning this value to all the image cells in the ith ray, each assignment being weighted by the corresponding w_{ij}.

With the preliminaries presented above, we will now discuss three different computer implementations of algebraic algorithms. These are represented by the acronyms ART, SIRT, and SART.

7.2 ART (Algebraic Reconstruction Techniques)

In many ART implementations the w_{ik}'s in (16) are simply replaced by 1's and 0's, depending upon whether the center of the kth image cell is within the ith ray. This makes the implementation easier because such a decision can easily be made at computer run time. In this case the denominator in (16) is given by $\Sigma_{k=1}^{N} w_{ik}^2 = N_i$ which is the number of image cells whose centers are within the ith ray. The correction to the jth image cell from the ith equation in (2) may now be written as

$$\Delta f_j^{(i)} = \frac{p_i - q_i}{N_i} \tag{17}$$

for all the cells whose centers are within the ith ray. We are essentially smearing back the difference $(p_i - q_i)/N_i$ over these image cells. In (17), q_i's are calculated using the expression in (15), except that one now uses the binary approximation for w_{ik}'s.

The approximation in (17), although easy to implement, often leads to artifacts in the reconstructed images, especially if N_i isn't a good approximation to the denominator. Superior reconstructions may be obtained if (17) is replaced by

$$\Delta f_j^{(i)} = \frac{p_i}{L_i} - \frac{q_i}{N_i} \tag{18}$$

where L_i is the length (normalized by δ, see Fig. 7.1) of the ith ray through the reconstruction region.

ART reconstructions usually suffer from *salt and pepper noise*, which is caused by the inconsistencies introduced in the set of equations by the approximations commonly used for w_{ik}'s. The result is that the computed ray-sums in (15) are usually poor approximations to the corresponding measured ray-sums. The effect of such inconsistencies is exacerbated by the fact that as each equation corresponding to a ray in a projection is taken up, it changes some of the pixels just altered by the preceding equation in the same projection. The SIRT algorithm described briefly below also suffers from these inconsistencies in the forward process [appearing in the computation of q_i's in (16)], but by eliminating the continual and competing pixel update as each new equation is taken up, it results in smoother reconstructions.

It is possible to reduce the effects of this noise in ART reconstructions by relaxation, in which we update a pixel by $\alpha \cdot \Delta f_j^{(i)}$, where α is less than 1. In some cases, the relaxation parameter α is made a function of the iteration number; that is, it becomes progressively smaller with increase in the number of iterations. The resulting improvements in the quality of reconstruction are usually at the expense of convergence.

## 7.3 SIRT (Simultaneous Iterative Reconstructive Technique)

In this approach, which at the expense of slower convergence usually leads to better looking images than those produced by ART, we again use (17) or (18) to compute the change $\Delta f_j^{(i)}$ in the jth pixel caused by the ith equation in (2). However, the value of the jth cell isn't changed at this time. Before making any changes, we go through all the equations, and then only at the end of each iteration are the cell values changed, the change for each cell being the average value of all the computed changes for that cell. This constitutes one iteration of the algorithm. In the second iteration, we go back to the first equation in (2) and the process is repeated.

7.4 SART (Simultaneous Algebraic Reconstruction Technique)

We will now discuss a variation on the algebraic approaches discussed above that seems to combine the best of ART and SIRT. This technique, first reported in [And84a], yields reconstructions of good quality and numerical accuracy in only one iteration. Here are the main features of SART: First, to reduce errors in the approximation of ray integrals of a smooth image by finite sums, the traditional pixel basis is abandoned in favor of bilinear elements. Also, for a circular reconstruction region, only partial weights are assigned to the first and last picture elements on the individual rays. To further reduce the noise resulting from the unavoidable but now presumably considerably smaller inconsistencies with real projection data, the correction terms are simultaneously applied for all the rays in one projection; this is in contrast with the ray-by-ray updates in ART. In addition, a heuristic procedure is used to improve the quality of reconstructions: a longitudinal Hamming window is used to emphasize the corrections applied near the middle of a ray relative to those applied near its ends.

In what follows we will describe in more detail the individual steps outlined above. The contribution that each step makes in improving the overall accuracy of the proposed procedure will be illustrated with reconstructions of the test image of Fig. 7.5. Note that this image differs slightly from a similar image in Chapter 3 by the presence of a "subdural hematoma," which is a small ellipse right next to the "skull" in the lower right-hand part. All these reconstructions were carried out on a 128 × 128 sampling lattice with 100 projections of 127 rays each.

Fig. 7.5: *(a) The Shepp and Logan head phantom with a subdural hematoma. (b) The gray level distribution of the Shepp and Logan phantom. (From [Kak84].)*
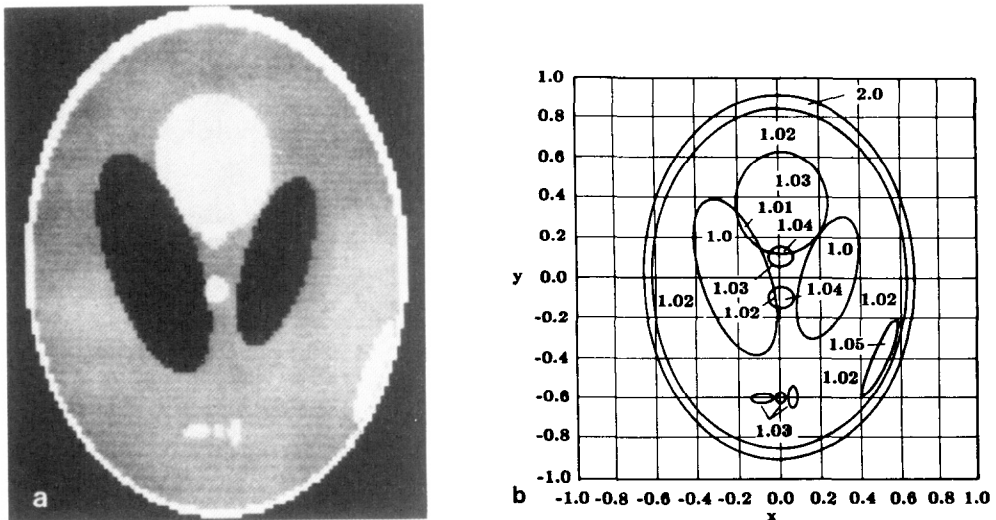
7.4.1 Modeling the Forward Projection Process

In (1), projection data were modeled by

$$p_i = \sum_{j=1}^{N} w_{ij}f_j, \qquad i = 1, 2, \cdots, M. \tag{19}$$

This is a good model for the projection process if for w_{ij}'s we use the theoretically dictated values—which, as mentioned before, is hard to do for various reasons.

To seek alternative methods for modeling the projection process, the relationship between a continuous image and the discrete projection data can be expressed by the following general form

$$p_i = R_i f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)\delta(r_i(x, y)) \, dx \, dy \tag{20}$$

where

$$r_i(x, y) = 0 \tag{21}$$

is the equation of the ith ray and R_i is the projection operator along the ray. The integral on the right-hand side serves as the definition of the projection operator.

Now suppose we assume that in an expansion for the image $f(x, y)$, we use basis functions $b_j(x, y)$ and that a good approximation to $f(x, y)$ is obtained by using N of them. This assumption can be represented mathematically by

$$f(x, y) \approx \hat{f}(x, y) \equiv \sum_{j=1}^{N} g_j b_j(x, y) \tag{22}$$

where g_j's are the coefficients of expansion; they form a finite set of numbers which describe the image $f(x, y)$ relative to the chosen basis set $b_j(x, y)$.

Substituting (22) in (20), we can write for the forward process

$$p_i = R_i f(x, y) \approx R_i \hat{f}(x, y) = \sum_{j=1}^{N} g_j R_i b_j(x, y) = \sum_{j=1}^{N} g_j a_{ij} \tag{23}$$

where a_{ij} represents the line integral of $b_j(x, y)$ along the ith ray. This equation has the same basic form as (1), yet it is more general in the sense that g_j's aren't constrained to be image gray level values over an array of points. Of course, the form here reduces to (1) if for b_j's we use the following pixel basis that is obtained by dividing the image frame into N identical subsquares; these are referred to as pixels and identified by the index j for $1 \leq j \leq N$:

$$b_j(x, y) = \begin{cases} 1 & \text{inside the } j\text{th pixel} \\ 0 & \text{everywhere else.} \end{cases} \tag{24}$$

In keeping with the nature of f_j's in (1), g_j's with these basis functions represent the average of $f(x, y)$ over the jth pixel and $R_i b_j(x, y)$ represents the length of the intersection of the ith ray with the jth pixel. Although (20) implies rays of zero width, if we now associate a finite width with each ray, the elements of the projection matrix will represent the areas of intersection of these ray strips with the pixels.

In SART, superior reconstructions are obtained by using a model of the forward projection process that is more accurate than what can be obtained by the choice of pixel basis functions—this is done by using bilinear elements which are the simplest higher order basis functions. The basis functions obtained from bilinear elements are pyramid shaped, each with a support extending over a square region the size of four pixels. It can be shown that the g_j's appearing in (22) for the case of bilinear elements are the sample values of the image function $f(x, y)$ on a square lattice. It can further be shown that whereas the pixel basis leads to a discontinuous image representation, the bilinear elements allow a continuous form of $\hat{f}(x, y)$ to be regenerated for computation. However, finding the exact ray integrals across such bilinear elements [as called for by $R_i b_j(x, y)$ in (23)] for a large number of rays is a time-consuming task and we will use an approximation.

Rather than try to find separately the individual coefficients a_{ij} for a particular ray, we approximate the overall ray integral $R_i \hat{f}(x, y)$ by a finite sum involving a set of M_i equidistant points $\{\hat{f}(s_{im})\}$, for $1 \leq m \leq M_i$ [Lyt80] (see Fig. 7.6):

$$p_i \approx \sum_{m=1}^{M_i} \hat{f}(s_{im}) \Delta s. \tag{25}$$

Fig. 7.6: *The ray-sum equations for a set of equidistant points along a straight line cut by the circular reconstruction region. (From [Kak84].)*
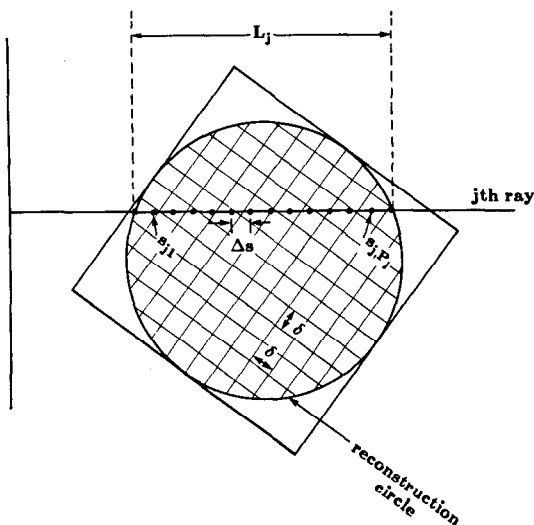
The value $\hat{f}(s_{im})$ is determined from the values g_j of $f(x, y)$ on the four neighboring points of the sampling lattice, i.e., by bilinear interpolation. We write

$$\hat{f}(s_{im}) = \sum_{j=1}^{N} d_{ijm} g_j \qquad \text{for } m = 1, 2, \cdots, M_i. \tag{26}$$

The coefficient d_{ijm} is therefore the contribution that is made by the jth image sample to the mth point on the ith ray. Combining (25) and (26), we obtain an approximation to the ray integral p_i as a linear function of the image samples g_j:

$$p_i = \sum_{m=1}^{M_i} \sum_{j=1}^{N} d_{ijm} g_j \Delta s \tag{27}$$

$$= \sum_{j=1}^{N} \sum_{m=1}^{M_i} d_{ijm} g_j \Delta s \qquad \text{for } 1 \le i \le J \tag{28}$$

$$= \sum_{j=1}^{N} a_{ij} g_j \tag{29}$$

where the coefficients a_{ij} represent the net effect of the linear transformations. They are determined as the sum of the contributions from different points along the ray:

$$a_{ij} = \sum_{m=1}^{M_i} d_{ijm} \Delta s. \tag{30}$$

Therefore, a_{ij} is proportional to the sum of contributions made by the jth image sample to all the points on the ith ray. It is important to the overall accuracy of the model that for $m = 1$ and for $m = M_i$, i.e., for the first and last points of the ray within the reconstruction circle, the weights are adjusted so that $\sum_{j=1}^{N} a_{ij}$ equals the actual physical length L_i.

One certainly has latitude about selecting the step size Δs; setting it equal to half the spacing of the sampling lattice provides a good trade-off between the accuracy of representation and computational cost.

7.4.2 Implementation of the Reconstruction Algorithm

As mentioned before, the results of SART implementation will be shown on 128 × 128 matrices using 100 projections, each with 127 rays. In the model of (29), this corresponds to $N = 16,384$ picture elements and an overall number of rays $I = 12,700$. Note that the system of equations is underdetermined by about 25%, but then the reconstruction circle covers only about 75% of the area of the square sampling lattice.

With the a_{ij}'s determined by the method just described, the reader will now be taken through a series of steps that are part of the SART implementation.

First, it will be shown that even with the superior forward projection modeling by the use of bilinear elements, one doesn't want to carry out a sequential implementation of the reconstruction algorithm.

A sequential implementation can be carried out by using the update formula of (4), reexpressed here in terms of SART symbols:

$$\vec{g}^{(k+1)} = \vec{g}^{(k)} + \vec{a}_i \frac{p_i - \vec{a}_i^T \vec{g}^{(k)}}{\vec{a}_i^T \vec{a}_i} \tag{31}$$

where $\vec{a}_i$ denotes the ith row vector of the array a_{ij}. As described before, the estimate $\vec{g}^{(k)}$ of the image vector is updated after each ray has been considered. We set the initial estimate $\vec{g}^{(0)}$ to zero, and we say that one iteration of the algebraic reconstruction technique is completed when all I rays, i.e., all I ray-sum equations, have been used exactly once. Owing to reasons discussed in Section 7.1, for sequential processing the projection data are ordered in such a manner that the angle between the projections considered successively is kept large; for the reconstructions shown here that were obtained with sequential updating, this angle was 73.8°.

Fig. 7.7(a) illustrates the reconstruction of the test image for one iteration of the sequential implementation. In order to avoid streak artifacts in the final image, the correction terms for the first few projections are de-emphasized relative to those for projections considered later on. The image has been thresholded to the gray level range 0.95–1.05 to illustrate the finer detail. Note that even the larger structures are buried in the salt and pepper noise present when no form of relaxation or smoothing is used. Fig. 7.7(b) shows a line plot through the three small tumors of the phantom (the profile shown is along the line $y = -0.605$). We observe that the amplitude variations of the noise largely exceed the density differences characterizing these structures.

Fig. 7.7: *Reconstruction from one iteration of sequential ART. (a) Image. (b) Line plot through the three small tumors (for $y = -0.605$). (From [And84a].)*
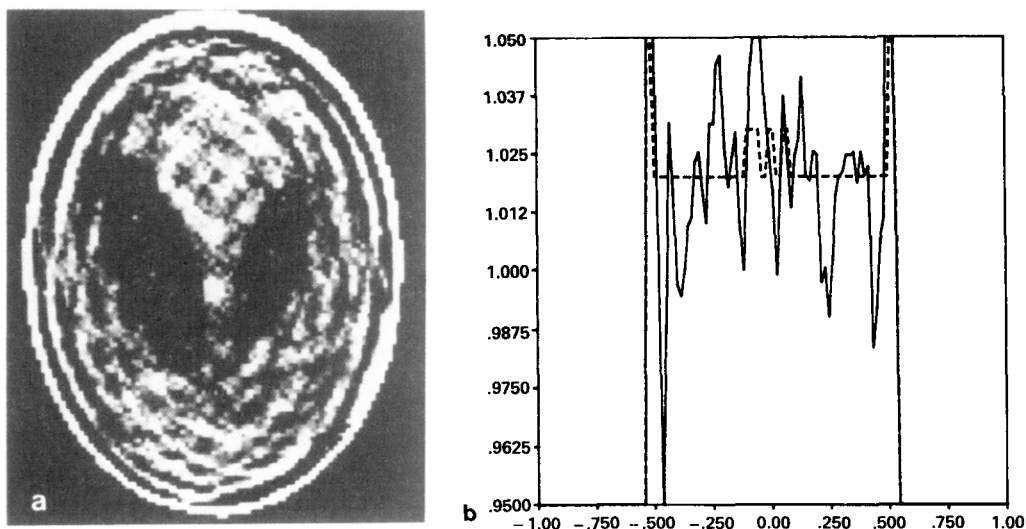
It will now be shown that superior results are obtained if instead of sequentially updating pixels on a ray-by-ray basis we simultaneously apply to a pixel the average of the corrections generated by all the rays in a projection. Stated in a bit more detail, this is what we want to do: For the first ray in a projection we compute as before the corrections to be made at every pixel. Instead of actually applying these corrections, we store them in a separate array to be called the correction array (the size of which is the same as that of the image array). Then we take up the next ray and add the pixel updates generated by this ray to the correction array. And then the next ray, and so on. After we are through all the rays in a projection, we add the correction array (or some fraction thereof) to the image array. This entire process is repeated with every projection. Fig. 7.8(a) illustrates the reconstruction obtained with this method. The precise formula that was used in the reconstruction in Fig. 7.8 for updating the pixel values can be stated as follows:

$$g_j^{(k+1)} = g_j^{(k)} + \frac{\sum_i \left[a_{ij} \dfrac{p_i - \vec{a}_i^T \vec{g}^{(k)}}{\sum_{j=1}^{N} a_{ij}} \right]}{\sum_i a_{ij}} \tag{32}$$

Fig. 7.8: *Reconstruction from one iteration of S.A.R.T. (a) Image. (b) Line plot through the three small tumors (for y = −0.605). (From [And84a].)*

where the summation with respect to *i* is over the rays intersecting the *j*th image element for a given scan direction.

Compared to the reconstruction of Fig. 7.7 for the sequential scheme, the simultaneous method offers a reduction in the amplitude of the noise. In addition, the noise in the reconstructed image has become more slowly
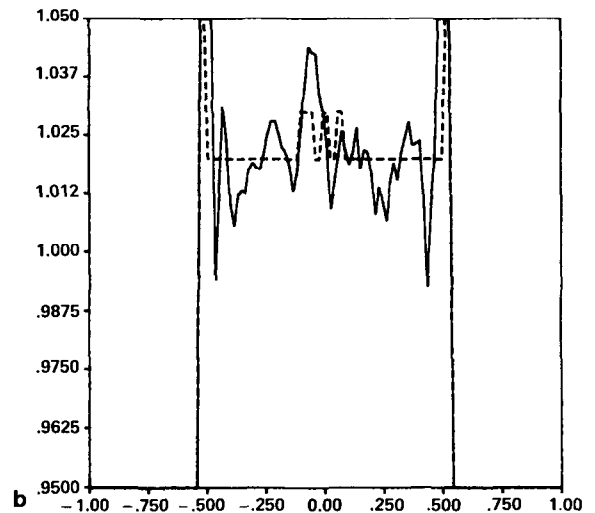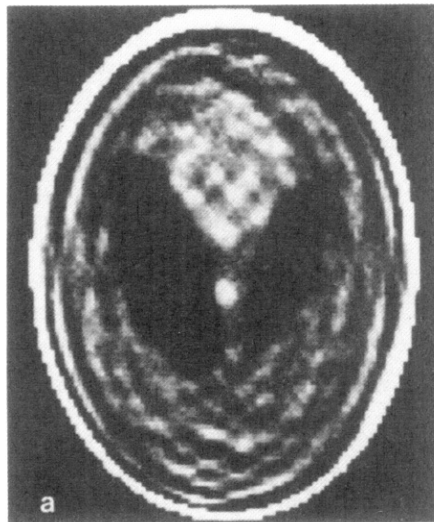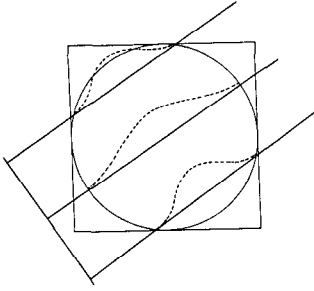
Fig. 7.9: *The longitudinal Hamming window for a set of straight rays. (From [And84a].)*

undulating compared to the previous salt and pepper appearance. This technique maintains the rapid convergence of ART-type algorithms while at the same time it has the noise suppressing features of SIRT. As with SIRT, the simultaneous implementation does require the storage of an additional array for the correction terms.

The last step, heuristic in nature, in SART consists of modifying the back-distribution of correction terms by a longitudinal Hamming window. The idea of the window is illustrated in Fig. 7.9. The uniform back-distribution according to the coefficients a_{ij} is replaced by a weighted version. This corresponds to replacing the correction term

$$a_{ij} \frac{p_i - \vec{a}_i^T \vec{g}^{(k)}}{\sum\limits_{j=1}^{N} a_{ij}} \tag{33}$$

in (32) by a weighted correction term

$$t_{ij} \frac{p_i - \vec{a}_i^T \vec{g}^{(k)}}{\sum\limits_{j=1}^{N} a_{ij}} \tag{34}$$

where the weighting coefficients t_{ij} are given by [compare with (30)]

$$t_{ij} = \sum_{m=1}^{M_i} h_{im} d_{ijm} \Delta s. \tag{35}$$

The sequence h_{im}, for $1 \leq m \leq M_i$, is a Hamming window of length M_i. Note that the length of the window varies according to the number of points M_i describing the part of the ray inside the reconstruction circle.

The weighted back-distribution of corrections emphasizes the central portions of rays in relation to portions closer to the periphery. Fig. 7.10 illustrates a reconstruction of the test image after one iteration with the longitudinal window in conjunction with the simultaneous scheme previously described. We see an improvement over the reconstructions of Figs. 7.7 and 7.8: the noise is practically gone and all the structures can be fairly well distinguished. If we hadn't applied the corrections in a simultaneous scheme but incorporated the longitudinal Hamming window only for the sequential implementation, we would have arrived at the noisy reconstruction illustrated in Fig. 7.11.

An important question that remains to be answered is: What happens when we go through iterations with, say, the simultaneous implementation; meaning that after we have made a reconstruction by going through all the projections once, we go through them all once again using the reconstruction of Fig. 7.10 as our initial solution; and then continue iterating in like fashion?
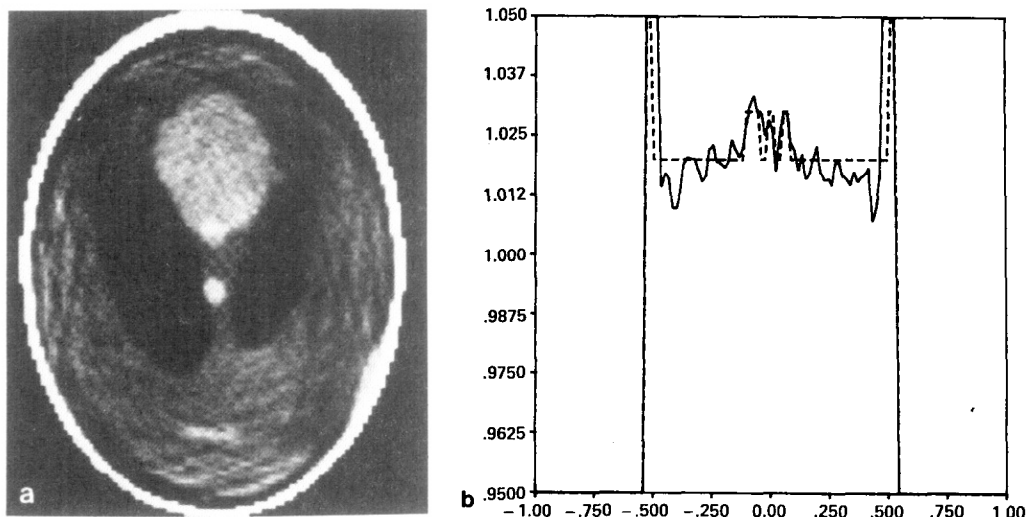
Fig. 7.10: *Reconstruction from one iteration of SART with a longitudinal Hamming window. (a) Image. (b) Line plot through the three small tumors (for y = −0.605). (From [And84a].)*

In Figs. 7.12 and 7.13, we have shown the reconstructions obtained with two and three iterations, respectively. As is evident from the reconstructions, we do gain more contrast, although at the cost of increased salt and pepper noise. All reconstructions shown represent the raw output from the algorithms with no postprocessing applied to suppress noise.

For the purpose of comparison, we have included in Fig. 7.14 the reconstruction obtained by using a convolution-backprojection algorithm. Comparing this with Fig. 7.10, we see that the SART reconstruction with one iteration is quite similar, although with further iterations, as displayed in Figs. 7.12 and 7.13, we see an increased amplitude of the salt and pepper noise, which is probably an indication of remaining inconsistencies in the model used for the forward projection process.

7.5 Bibliographic Notes

The earliest expositions on algebraic reconstruction were by Gordon *et al.* [Gor70], [Gor71], [Gor74], Herman *et al.* [Her71], [Her73], [Her77], and Budinger and Gullberg [Bud74]. The reader is also referred to the book by Herman [Her80] for an exhaustive treatment of the subject.

When binary values are chosen for the weights w_{ij} in (16) in ART, i.e., w_{ij} is set equal to 1 if the center of the jth pixel falls within the strip of the ith ray and 0 if not, it becomes necessary to adjust the width of each ray according to the orientation of the projection [Gor74], [Her73], [Opp75].

Attempts have been made to reduce the salt and pepper noise associated with ART-type reconstructions by increasing the number of rays per view [Smi77]. When the number of rays per view is increased, many pixels are
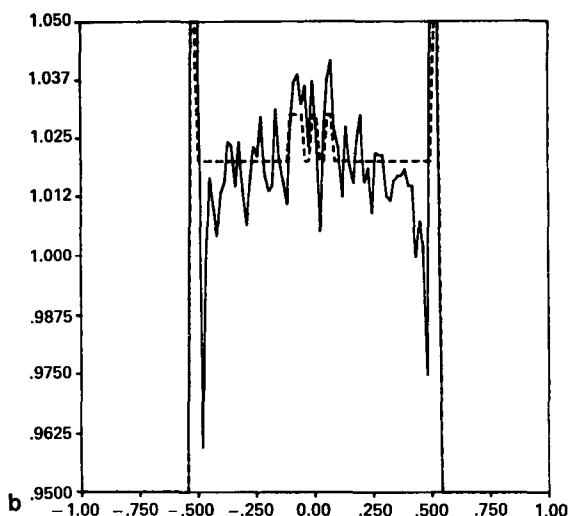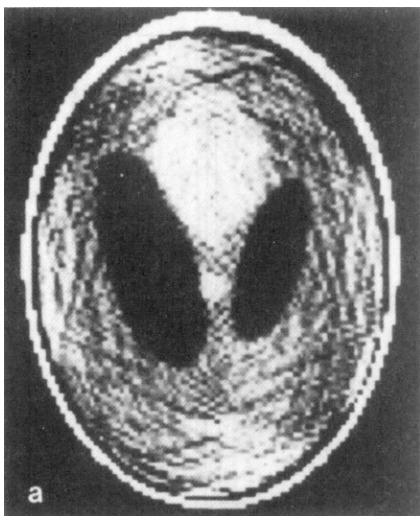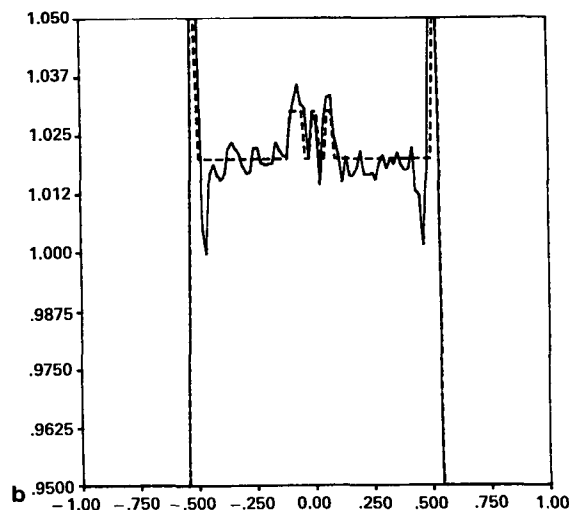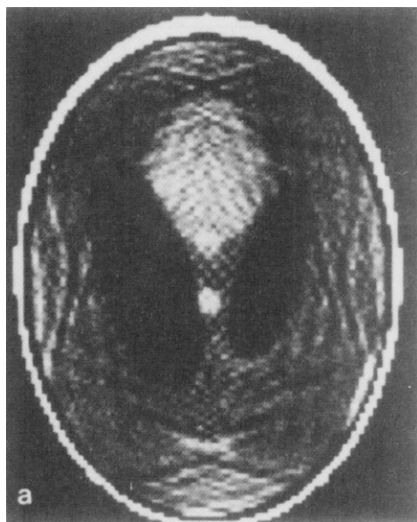
Fig. 7.11: *Reconstruction from one iteration of sequential ART with a longitudinal Hamming window. (a) Image. (b) Line plot through the three small tumors (for y = −0.605). (From [And84a].)*

Fig. 7.12: *Reconstruction from two iterations of SART with a longitudinal Hamming window. (a) Image. (b) Line plot through the three small tumors (for y = −0.605). (From [And84a].)*

intersected by several rays in each projection. This results in the averaging of possible errors committed in the correction procedure such as the one given by (4). Common practice is to have a system with about four times as many equations as unknown pixel values [Her80], [Her78], [She74]. The computational cost, however, is increased directly with the number of rays processed. An additional method has been to use a relaxation factor $\lambda < 1$ [Gor74], [Her80], [Her76], [Her78], [Hou72], [Swe73] which, although reducing the salt and pepper noise, increases the number of iterations required for convergence.

The SART algorithm was first reported in [And84a]. In contrast with the bilinear elements used for SART, the pixel basis is common to much
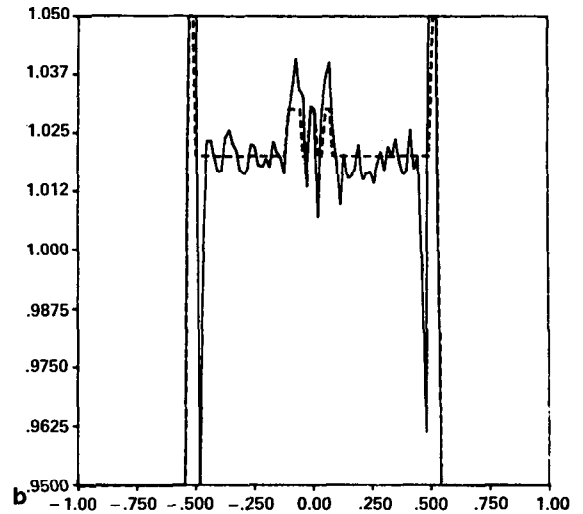
Fig. 7.13: *Reconstruction from three iterations of SART with a longitudinal Hamming window. (a) Image. (b) Line plot through the three small tumors (for y = – 0.605). (From [And84a].)*

Fig. 7.14: *Convolution-back-projection reconstruction of the test image. (a) Image. (b) Line plot through the three small tumors (for y = – 0.605). (From [And84a].)*

literature published on algebraic techniques [Din79], [Gil72], [Gor74], [Gor70], [Her80], [Her76], [Her78], [Her73], [Hou72], [Opp75], [She74].

The error-correcting procedure of the basic ART algorithm as given by (4) is discussed in [Gor74], [Gor70], [Her80], [Her76], [Her78], [Her73], [Hou72].

As first shown by Hounsfield [Hou72], in order to improve the convergence of a sequential algebraic algorithm one should order the projections in such a manner that successive projections are well separated. This he justified on the basis of high correlation between the information in neighboring projections. Later the scheme was demonstrated to have a deeper
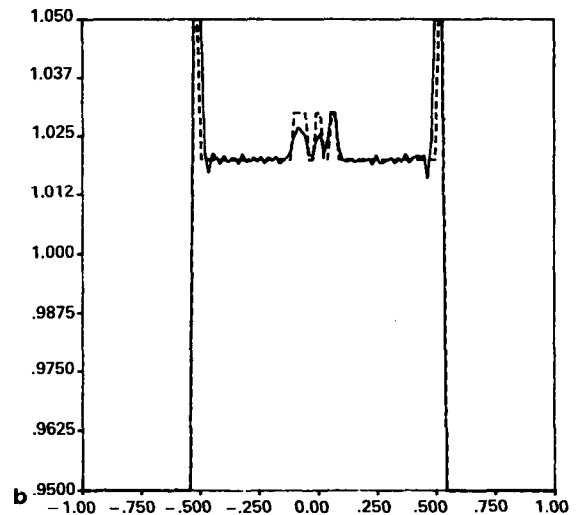
mathematical foundation as a tool for speeding up the convergence of ART-type algorithms. (The proof relies on a continuous formulation of ART, as shown by Hamaker and Solmon [Ham78].) Ramakrishnan *et al.* [Ram79] have shown how by orthogonalization of the algebraic equations we can increase the speed of convergence of a reconstruction algorithm.

The SIRT algorithm was first proposed by Gilbert [Gil72]. A simplified form of the simultaneous technique was used by Oppenheim in [Opp75]. However, the scope of the implementation as described by (32) is much wider. The method can be used advantageously in the general image reconstruction problem for curved rays with overlapping and nonoverlapping ray strips as well as in conjunction with any image representation, provided the forward process can be expressed in the form of (23).

A combination of algebraic reconstruction and digital ray tracing appears ideal for imaging lightly refracting objects [Cha79], [Cha81]. A survey of digital ray tracing and ray linking for this purpose is presented in [And82]. If a refracting object has special symmetries, then as shown by Vest [Ves75] it may be possible to reconstruct the object without ray tracing. The reader is referred to [And84b] for experimental demonstrations of how algebraic reconstruction can be combined with digital ray tracing for the cross-sectional imaging of lightly refracting objects.

7.6 References

[And82]   A. H. Andersen and A. C. Kak, "Digital ray tracing in two-dimensional refractive fields," *J. Acoust. Soc. Amer.*, vol. 72, pp. 1593-1606, Nov. 1982.

[And84a] ——, "Simultaneous algebraic reconstruction technique (SART): A superior implementation of the art algorithm," *Ultrason. Imaging*, vol. 6, pp. 81-94, Jan. 1984.

[And84b] ——, "The application of ray tracing towards a correction for refracting effects in computed tomography with diffracting sources," TR-EE 84-14, School of Electrical Engineering, Purdue Univ., Lafayette, IN, 1984.

[Bud74]   T. F. Budinger and G. T. Gullberg, "Three-dimensional reconstruction in nuclear medicine emission imaging," *IEEE Trans. Nucl. Sci.*, vol. NS-21, pp. 2-21, 1974.

[Cha79]   S. Cha and C. M. Vest, "Interferometry and reconstruction of strongly refracting asymmetric-refractive-index fields," *Opt. Lett.*, vol. 4, pp. 311-313, 1979.

[Cha81]   ——, "Tomographic reconstruction of strongly refracting fields and its application to interferometric measurements of boundary layers," *Appl. Opt.*, vol. 20, pp. 2787-2794, 1981.

[Din79]   K. A. Dines and R. J. Lytle, "Computerized geophysical tomography," *Proc. IEEE*, vol. 67, pp. 1065-1073, 1979.

[Gil72]   P. Gilbert, "Iterative methods for the reconstruction of three dimensional objects from their projections," *J. Theor. Biol.*, vol. 36, pp. 105-117, 1972.

[Gor70]   R. Gordon, R. Bender, and G. T. Herman, "Algebraic reconstruction techniques (ART) for three dimensional electron microscopy and X-ray photography," *J. Theor. Biol.*, vol. 29, pp. 471-481, 1970.

[Gor71]   R. Gordon and G. T. Herman, "Reconstruction of pictures from their projections," *Commun. Assoc. Comput. Mach.*, vol. 14, pp. 759-768, 1971.

[Gor74]   R. Gordon, "A tutorial on ART (algebraic reconstruction techniques)," *IEEE Trans. Nucl. Sci.*, vol. NS-21, pp. 78-93, 1974.

[Ham78]   C. Hamaker and D. C. Solmon, "The angles between the null spaces of X rays," *J. Math. Anal. Appl.*, vol. 62, pp. 1-23, 1978.

[Her71]   G. T. Herman and S. Rowland, "Resolution in ART: An experimental investigation

of the resolving power of an algebraic picture reconstruction," *J. Theor. Biol.*, vol. 33, pp. 213–233, 1971.

[Her73] G. T. Herman, A. Lent, and S. Rowland, "ART: Mathematics and applications: A report on the mathematical foundations and on applicability to real data of the algebraic reconstruction techniques," *J. Theor. Biol.*, vol. 43, pp. 1–32, 1973.

[Her76] G. T. Herman and A. Lent, "Iterative reconstruction algorithms," *Comput. Biol. Med.*, vol. 6, pp. 273–294, 1976.

[Her77] G. T. Herman and A. Naparstek, "Fast image reconstruction based on a Radon inversion formula appropriate for rapidly collected data," *SIAM J. Appl. Math.*, vol. 33, pp. 511–533, Nov. 1977.

[Her78] G. T. Herman, A. Lent, and P. H. Lutz, "Relaxation methods for image reconstruction," *Commun. A.C.M.*, vol. 21, pp. 152–158, 1978.

[Her80] G. T. Herman, *Image Reconstructions from Projections*. New York, NY: Academic Press, 1980.

[Hou72] G. N. Hounsfield, "A method of and apparatus for examination of a body by radiation such as x-ray or gamma radiation," Patent Specification 1283915, The Patent Office, 1972.

[Kac37] S. Kaczmarz, "Angenaherte auflosung von systemen linearer gleichungen," *Bull. Acad. Pol. Sci. Lett. A*, vol. 6-8A, pp. 355–357, 1937.

[Kak84] A. C. Kak, "Image reconstructions from projections," in *Digital Image Processing Techniques*, M. P. Ekstrom, Ed. New York, NY: Academic Press, 1984.

[Lyt80] R. J. Lytle and K. A. Dines, "Iterative ray tracing between boreholes for underground image reconstruction," *IEEE Trans. Geosciences and Remote Sensing*, vol. GE-18, pp. 234–240, 1980.

[Opp75] B. E. Oppenheim, "Reconstruction tomography from incomplete projections," in *Reconstruction Tomography in Diagnostic Radiology and Nuclear Medicine*, M. M. Ter Pogossian *et al.*, Eds. Baltimore, MD: University Park Press, 1975.

[Ram79] R. S. Ramakrishnan, S. K. Mullick, R. K. S. Rathore, and R. Subramanian, "Orthogonalization, Bernstein polynomials, and image restoration," *Appl. Opt.*, vol. 18, pp. 464–468, 1979.

[Ros82] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd ed. New York, NY: Academic Press, 1982.

[She74] L. S. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nucl. Sci.*, vol. NS-21, pp. 21–43, 1974.

[Sla85] M. Slaney and A. C. Kak, "Imaging with diffraction tomography," TR-EE 85-5, School of Electrical Engineering, Purdue Univ., Lafayette, IN, 1985.

[Smi77] K. T. Smith, D. C. Solmon, and S. L. Wagner, "Practical and mathematical aspects of the problem of reconstructing objects from radiographs," *Bull. Amer. Math. Soc.*, vol. 83, pp. 1227–1270, 1977.

[Swe73] D. W. Sweeney and C. M. Vest, "Reconstruction of three-dimensional refractive index fields from multi-directional interferometric data," *Appl. Opt.*, vol. 12, pp. 1649–1664, 1973.

[Tan71] K. Tanabe, "Projection method for solving a singular system," *Numer. Math.*, vol. 17, pp. 203–214, 1971.

[Ves75] C. M. Vest, "Interferometry of strongly refracting axisymmetric phase objects," *Appl. Opt.*, vol. 14, pp. 1601–1606, 1975.