THREE-DIMENSIONAL RECONSTRUCTION IN NUCLEAR MEDICINE BY ITERATIVE LEAST-SQUARES AND FOURIER TRANSFORM TECHNIQUES

Thomas F. Budinger and Grant T. Gullberg

Donner Laboratory and Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

January 1974

Abstract

Quantitative three-dimensional distribution of isotopes in patients is determined by digital reconstruction of data from many views taken by rotating the subject at 10° intervals before the gamma camera. The superiority of these techniques over conventional tomography is demonstrated by comparisons between reconstruction algorithms such as back-projection, simultaneous iterative reconstruction, iterative least-squares, and back-projection of filtered projection. The filtered back-projection technique (convolution method) is superior in speed; however, for quantitative results that take into account both noise and attenuation, the iterative least-squares method gives the best approximation to the real source distributions. Resolution is 1.25 cm for detection of holes in 20-cm-diameter objects.

Mathematical basis and FORTRAN listings applicable to transmission and emission imaging are given, as well as phantom and patient studies.

Table of Contents

1.0 INTRODUCTION

1.1 Scope and Previous Work

This study presents application of methods of ascertaining the three-dimensional distribution of isotope concentration or density in nuclear medicine, and differs from previous three-dimensional reconstruction efforts of astrophysics, electron microscopy, and x-ray radiology in that statistically poor measurements and photon attenuation are taken into account by the algorithm. Truly quantitative nuclear medicine cannot be accomplished from single views in most cases (Budinger, 1974); thus a means of estimating isotope concentration from data taken from multiple views is needed. The methods discussed here are applicable to photon or heavy ion transmission radiography as well as emission imaging.

The methods of three-dimensional reconstruction from multiple two-dimensional views can be dividied into thirteen categories:

1. Direct matrix techniques, generalized inverse and pseudo-inverse (Sandler, 1972; Kashyap and Mittal, 1973).

2. Summation, linear superposition, back-projection, moiré, or simple transverse-section scanning (Andrews, 1936; Edholm, 1960; Kuhl and Edwards, 1963, 1966, 1968; Anger, 1967, 1974; R. G. Hart, 1968; Harper, 1968; Vainshtein, 1970; Reichmann, 1972; Gordon and Herman, 1974).

3. Algebraic reconstruction technique (Gordon et al., 1970; Schmidlin, 1972, 1973).

4. Algebraic reconstruction technique modified for noise (Herman et al., in press; Johnson et al., 1973).

5. Simultaneous iterative reconstruction technique (Gilbert, 1972a).

6. Orthogonal tangent correction (Kuhl et al., 1973).

7. Iterative least-squares technique (Goitein, 1971).

8. Summation of compensated projections (Cho et al., in press; Vainshtein, 1973).

9. Summation of filtered back-projections, convolution technique (Bracewell and Riddle, 1967; Gilbert, 1972b; Ramachandran and Lakshminarayanan, 1971; Smith et al., 1973; Peters, 1973; Chesler, 1972; Shepp, in press; Lee et al., in press.

10. Geometric mean iterative technique (Schmidlin, in press).

11. Rho filtered back-projection (Bates and Peters, 1971).

12. Fourier reconstruction (DeRosier and Klug, 1968; Crowther et al., 1970; Budinger, 1971; Lake, 1971; Peters et al., 1973; Keyes and

Simon, 1973).

13. Summation of the projections after Hilbert transform of the derivative of the projection (Radon, 1917; John, 1955; Berry and Gibbs, 1970; Cormack, 1973; Peters, 1973).

Some of these techniques have been compared for accuracy, computer time, number of views required, and ability to handle noise (Frieder and Herman, 1971; Herman, 1972; Herman and Rowland, 1972; Herman et al., 1974). An alternative classification of methods into four categories: Summation (No. 1); Use of Fourier Transforms (Nos. 9, 11, 12); Analytic Solution of Integral Equations (Nos. 9, 12, 13); and Series Expansion Approaches (Nos. 3, 4, 5, 6, 7) has been presented with a review of literature by Gordon and Herman (1974).

The various methods in general are equivalent under certain conditions of transformation in that the result of the reconstruction is related to the true object by some integral transformation within the limitations of the statistic of the measurement. The direct methods involving matrix inversion are usually discarded because the matrices are too large or the system is undetermined and the equations will be inconsistent. This is not necessarily true if the generalized inverse is used; however, no implementation has been made as yet.

Many of the algorithms are mechanisms of evaluating Radon's relation between the value of each picture element in polar coordinates $A(r, \theta)$ and the projections for all angles $P(x, \theta)$ where x denotes an element along the projection corresponding to the line integral through the section to be reconstructed. Thus Radon in 1917 and subsequently others (Berry and Gibbs, 1970 Cormack, 1973; and Peters, 1973) showed that

$$A(r,\theta) = \frac{1}{2\pi^2} \int_{-\pi/2}^{\pi/2} \int_{-\infty}^{+\infty} \frac{\partial P(x, \theta')}{\partial x} \cdot \frac{1}{r\sin(\theta-\theta') - x} \, dx \, d\theta' \qquad (1)$$

In the practical situation there is only a finite number of views and each measurement is subject to errors. Thus, over the last 15 years specialized techniques have been developed for solving the problem of estimating the distribution of some property in three-dimensional space from many views or projections at various discrete angles.

The text is divided into a discussion of the mathematical algorithms, results with phantoms and patients, and methods of handling attenuation. Aspects of computer implementation are given, along with FORTRAN listings, as Appendices. Appendix A gives proofs of the Fourier projection theorem and the relation between the back-projected image and the true image.

2.0 MATHEMATICAL TECHNIQUES

2.1 Three-Dimensional Reconstruction by Stacked Two-Dimensional Reconstructions

This section presents some of the algorithms implemented for three-dimensional reconstruction of density of isotope distribution. For computational simplicity the object is divided into planes along the axis of rotation. Each plane is reconstructed from some mathematical operation on the corresponding one-dimensional projections, and the planes are stacked to reconstitute the three-dimensional object. Thus we consider the problem of reconstructing planes or transverse sections from multiple projections (Fig. 1). This simplification is not possible for cone beams or three-dimensional fan beams. The fan beam is a diverging beam, which by source or detector collimation samples a single transverse plane. Most of the algorithms given below can be modified to handle either parallel or fan beam situations.

2.2 Relations Between Picture Elements and Projection Rays

The digital techniques of acquiring data and manipulating projections in order to obtain a two-dimensional reconstruction by any of the above methods are given in more detail. All the methods require an algorithm for determining the ray $k(\theta)$ for a particular projection θ which passes through a given picture element (Fig. 2). In addition, we need an algorithm which gives all pixels (i, j) which intersect the given ray $k(\theta)$.

The appropriate recipe for relating the coordinate (x, y) of one array rotated some angle θ from the reference array with coordinate (u, v) is usually given as

$$\begin{aligned} x &= u\cos\theta - v\sin\theta \\ y &= u\sin\theta + v\cos\theta \end{aligned} \qquad \text{with the inverse} \qquad \begin{aligned} u &= x\cos\theta + y\sin\theta \\ v &= -x\sin\theta + y\cos\theta . \end{aligned} \tag{2}$$

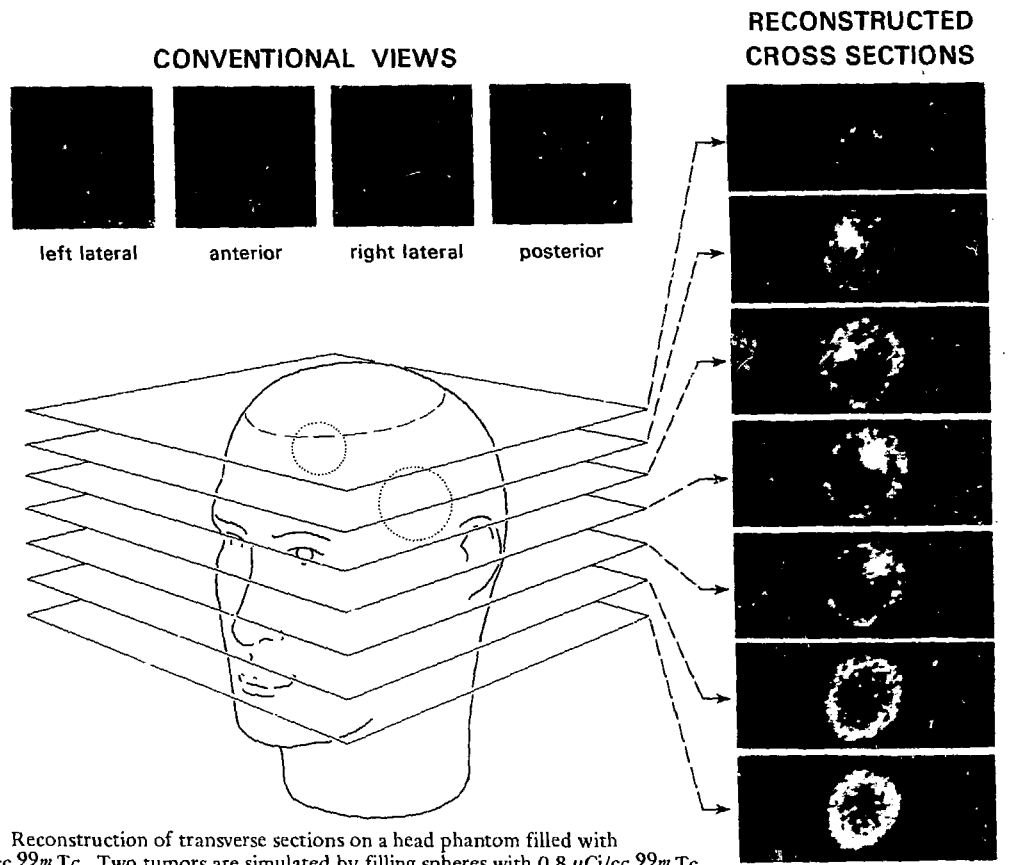


Figure 1. Reconstruction of transverse sections on a head phantom filled with 0.4 μCi/cc ^{99m}Tc. Two tumors are simulated by filling spheres with 0.8 μCi/cc ^{99m}Tc. Full three-dimensional reconstruction is effected by doing multiple two-dimensional section reconstruction and reconstituting the whole by stacking these sections.

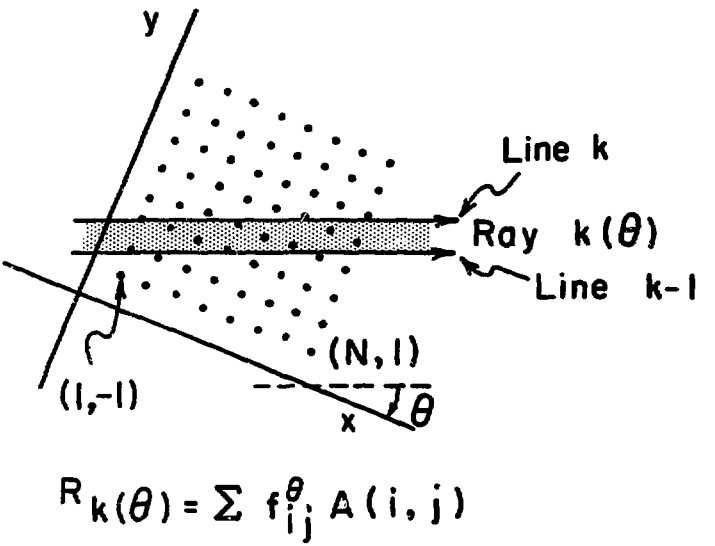


Figure 2. Ray sums are formed by adding the activity from each picture element that falls within ray lines k-1 and k.

From Eq. (2) one can derive a digital algorithm for determining which pixels of a rotated array correspond to the reference array; however, an alternate approach was taken as detailed in Appendix B. The preferred approach is based on the need to derive an algorithm which will give all picture elements lying within the boundaries of a specific projection ray as well as a recipe which gives all rays that intersect a particular pixel as a function of θ.

The projection of rays passing through the plane remains fixed relative to the detector, while the coordinate system rotates within the field of these fixed rays. This is done so that the formulation corresponds to the actual experiments wherein an object or patient is rotated in front of a fixed camera and thus differs from other formulations (e.g., see Gordon et al., 1970). The practical results of these derivations are placed in convenient terms for the digital computer as follows:

Family of lines bounding the rays

For angles $\theta = 0°, 90°, 180°$ and $270°$, the family of lines is

$$y = k + 1/2 \qquad k = 0, 1, 2 \ldots . N \tag{3}$$

For all angles other than integral multiples of $\pi/2$, the family of lines is

$$y = x\tan\theta + y_0 + k/|\cos\theta| , \qquad k = 0, 1, 2 \ldots . . n_\theta . \tag{4}$$

Where

$$y_o = \frac{N+1}{2} - \frac{L}{|\cos\theta|} - \frac{N+1}{2} \tan\theta ,$$

$$L = \begin{cases} \frac{N}{2} + \text{INT}\left[\frac{N-1}{2}\left(|\sin\theta| + |\cos\theta| - 1\right) + 1/2\right] & \text{if } [\cdot] > \text{INT}[\cdot] \\ \frac{N}{2} + \text{INT}\left[\frac{N-1}{2}\left(|\sin\theta| + |\cos\theta| - 1\right) + 1/2\right] - 1 & \text{if } [\cdot] = \text{INT}[\cdot] \end{cases}$$

$$n_\theta = \begin{cases} N + 2\text{INT}\left[\frac{N-1}{2}\left(|\sin\theta| + |\cos\theta| - 1\right) + 1/2\right] & \text{if } [\cdot] > \text{INT}[\cdot] \\ N + 2\text{INT}\left[\frac{N-1}{2}\left(|\sin\theta| + |\cos\theta| - 1\right) + 1/2\right] - 1 & \text{if } [\cdot] = \text{INT}[\cdot] \end{cases}$$

Using these equations, the minimum and maximum values for y or the j coordinates of the pixels that fall within a ray k (between lines k-1 and k) are determined for each projection angle. Then between these bounds all the i coordinates are determined by solving the respective equations for x. This gives a set $\{(i,j) \mid (i,j) \in \text{ray } k(\theta)\}$ where the coordinate pairs belong to the kth ray of protection θ. If a coordinate pair falls on the line k, then the coordinate pair is placed in ray k+1.

For each given projection angle θ, we determine the ray $k(\theta)$ that passes through a particular coordinate pair (I, J) using the following formula for the distance between the pixel represented by the coordinate pair (I, J) and the line k = 0:

$$D = \begin{cases} L + \frac{(N+1-2I)}{2} |\sin\theta| + \left(\frac{2J-N-1}{2}\right)|\cos\theta| & \text{if } \begin{matrix} 0° \leq \theta < 90° \\ 180° \leq \theta < 270° \end{matrix} \\ L + \frac{(2I-N-1)}{2} |\sin\theta| + \frac{(2J-N-1)}{2}|\cos\theta| & \text{if } \begin{matrix} 90° \leq \theta < 180° \\ 270° \leq \theta < 360° \end{matrix} \end{cases} \quad (5)$$

where L is given in Eq. B21 (Appendix B). The integer value of D+1 gives the ray number. Thus a one-statement operation for each projection θ will yield the proper ray number for a given pixel. In the case of a simple back projection on a 64 X 64 array using Eq. (5), the number of calculations is 4096 times the number of projections; alternatively, the back-projection summation can be determined by assigning the value $P_{k(\theta)}$ to each pixel through which the ray passes, which means the number of calculations is the product: number of projections times the 64 rays times number of pixels in each ray. The latter method might be more costly in time, because each ray must be bounded by a series of logical computer statements. This formulation does not take into account the fractional area of the pixel through which a ray passes. The fractional contribution each ray sum might make to a picture element varies between 0 and 1. For each pixel this weighting can be incorporated by calculating the fractional area of the pixel intersected by each ray, or the length of the ray through the pixel, or by a factor related to the distance between the ray and the pixel center. Another approach is to modify the ray width in accordance with the angle of projection relative to a square array. Incorporation of these weighting factors is costly in computer time, and is not essential if the number of rays and fineness of the array are appropriately matched to the data (Frieder and Herman, 1971). The weighting matrix discussed in the next section incorporates these weighting factors.

3.0 DETAILS OF SOME RECONSTRUCTION METHODS

3.1 Scope

Of the thirteen methods listed in the introduction, we concentrate here on the implementation of the back projection, the simultaneous iterative reconstruction technique, the iterative least-squares technique, and back-projection of filtered or compensated projections. The direct matrix approach is presented not only as an introduction to the iterative techniques, but also to give the framework for possible future work.

3.2 Direct-Matrix and Linear-Equation Methods

In this section we examine the application of linear algebra to the problem of determining the concentration or density in each element of a two-dimensional section from a number of projections.

Consider the simple problem of reconstructing the four values in a 2 × 2 array from two projections at 0° and 90°

A_1	A_2	$\Sigma = 3$
A_3	A_4	$\Sigma = 7$
$\Sigma = 4$	$\Sigma = 6$	

The feasible solutions are given by the following set of equations:

$$\begin{aligned} A_1 + A_2 &= P_1 = 3 \\ A_3 + A_4 &= P_2 = 7 \\ A_1 + A_3 &= P_3 = 4 \\ A_2 + A_4 &= P_4 = 6 \end{aligned} \tag{6}$$

This system of equations has an infinite number of solutions because Eqs. (6)

are not independent (the rank of the coefficient matrix and the augmented matrix is 3). The impossibility of a unique direct solution can be seen from the following attempt to solve the system Eqs.(6). In matrix notation the system is given as

$$F \cdot A = P, \tag{7}$$

where

$$F = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix}, \text{ and } P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix}.$$

The matrix F can be considered a weighting matrix based on the geometry. Recall from matrix algebra the explicit solution of Eq. (7) is

$$A = F^{-1} \cdot P, \tag{8}$$

where the elements of F^{-1} are related to the matrix F as follows:

If

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & \cdots & \\ f_{31} & \cdots & \cdots & \\ f_{41} & \cdots & & f_{44} \end{bmatrix}; \text{ then } F^{-1} = \frac{1}{\text{Det } F} \begin{bmatrix} f'_{11} & f'_{21} & f'_{31} & f'_{41} \\ f'_{12} & f'_{22} & \cdots & \\ f'_{13} & \cdots & \cdots & \\ f'_{14} & \cdots & & f'_{44} \end{bmatrix}$$

where f'_{ij} is the cofactor of f_{ij}.

Note for the example above Det F = 0; thus, there is no unique solution to Eqs. (6). This is an example of four equations (inconsistent) and four unknowns with no solution by Eq. (8). It is possible to solve for a 2 X 2 square array with only two projections by either an iterative scheme or by changing the projection

angles; in this case, angles 0° and 45° will suffice. Consider the system of equations for views at 0° and 45°

$$\begin{aligned} A_1 \quad + A_3 \quad &= P_1 \\ A_2 \quad + A_4 \quad &= P_2 \\ A_1 \quad &= P_3 \\ A_2 + A_3 \quad &= P_4 \end{aligned} \tag{9}$$

The solution of this system is given by Eq. (8), where now the inverse matrix can be evaluated

$$F^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 \\ 1 & 1 & -1 & -1 \end{bmatrix}$$

The extension of this problem to real data and large arrays involves serious complications; for example, suppose there are sufficient data that the values in a section of say N X N picture elements can be determined by a linear system of simultaneous equations. At first glance, it would seem for a 64 X 64 array that 4096 simultaneous equations must be solved. Most investigations stop at this revelation and proceed to other methods discussed below; however, it is important to investigate this problem in more detail before giving in to the notion that such a large matrix inversion is intractable. Indeed, as will be seen, the direct-matrix method involves a matrix size equal to approximately

$$(N \times N) \cdot (\text{no. of projections}) \cdot (\text{no. of elements per projection}) \ .$$

Consider one projection composed of the ray sum or line integrals $\{ P_{k\theta} \}$, $k = 1, n_\theta$. At 0°, 90°, or integral multiples of $\pi/2$, each ray of width unity

intersects a row or column of the section (Fig. B1). However, for projections at some other angles, the rays will not intersect each picture element entirely that is, part of the ray width will encompass a picture element. Thus some weighting factor f_{ij} is necessary to describe the contribution of a particular pixel A(i, j) to the ray sum $P_{k\theta}$. The weighting factor f_{ij} can also be used to account for attenuation. The system of linear equations for one projection at 0° is

$$\begin{aligned}
P_{1\theta_1} &= f_{11}\,A(1,1) + f_{12}\,A(1,2) + \ldots + f_{1N}\,A(1,N)\ , \\
P_{2\theta_1} &= f_{21}\,A(2,1) + f_{22}\,A(2,2) + \ldots + f_{2N}\,A(2,N)\ , \\
&\vdots \qquad\qquad\qquad\qquad\qquad \vdots \\
P_{N\theta_1} &= f_{N1}\,A(N,1) + f_{N2}\,A(N,2) + \ldots + f_{NN}\,A(N,N)\ .
\end{aligned} \tag{10}$$

For projections at angles of integral multiples of $\pi/2$ for a square array where the ray is equal to the width of a picture element, all weighting factors are 1; however, for $P_{k\theta_2}$, say at 3° from $P_{k\theta_1}$, the weighting factors will be less than one, and the terms of Eq. (10) will vary in accordance with the elements through which the rays pass, which is dependent on θ. One can generalize to all projections

$$\begin{aligned}
P_1 &= f_{11}\,A(1,1) + f_{12}\,A(1,2) + \ldots + f_{1}\,w\,A(N,N)\ , \\
P_2 &= f_{21}\,A(1,1) + f_{22}\,A(1,2) + \ldots + f_{2}\,w\,A(N,N)\ , \\
&\vdots \qquad\qquad\qquad\qquad\qquad \vdots \\
P_m &= f_{m1}\,A(1,1) + f_{m2}A(1,2) + \ldots + f_{mw}\,A(N,N)\ ,
\end{aligned} \tag{11}$$

where m is the total number of the rays for all projections, and w is equal to N^2.

Written in matrix notation, the previous equation can be expressed as

$$\begin{bmatrix} P_1 \\ P_2 \\ \cdot \\ \cdot \\ \cdot \\ P_m \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1w} \\ f_{12} & f_{22} & & f_{2w} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ f_{m1} & f_{m2} & \cdots & f_{mw} \end{bmatrix} \begin{bmatrix} A(1,1) \\ A(1,2) \\ \cdot \\ \cdot \\ \cdot \\ A(N,N) \end{bmatrix} . \qquad (12)$$

In matrix notation $P = F \cdot A$, and in the case where $m = w = N^2$ as before, we solve for $[A(i,j)]$ by inverting the matrix F

$$\begin{aligned} A(1,1) &= P_1 f^*_{11} + P_2 f^*_{12} + \cdots + P_m f^*_{1w} \\ A(1,2) &= P_1 f^*_{21} + P_2 f^*_{22} + \cdots + P_m f^*_{2w} \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \\ &\cdot \\ A(N,N) &= P_1 f^*_{w1} + P_2 f^*_{w2} + \cdots + P_m f^*_{mw} \end{aligned} \qquad (13)$$

$$A = F^{-1} \cdot P$$

where the elements of the inverse matrix F^{-1} are $[f^*_{ij}]$.

If one can calculate the elements of the matrix F^{-1}, then the solution will be a simple multiplication and addition of these elements with all the ray sums for the projections. In the example above, this involves for each element, $(N \times N)^2$ multiplications and $(N \times N)^2$ addition operations. In practice on a computing machine with 1 μsec per complete operation, this means

$$1 \times 10^{-6} \text{ sec} \times [(64 \times 64)^2 + (64 \times 64)^2] = 33 \text{ sec}$$

The long computing time plus the need to store 1.678×10^7 values for F^{-1} have prompted a search for alternate approaches. Further, the problem of measurement errors and insufficient angular measurements to satisfy Eq. (13) have resulted in the 13 algorithms cited above and detailed below or in other reviews (Frank, 1973; Gordon and Herman, 1974).

If there are potentially serious measurement errors, the problem can be formulated by requiring that an estimate of the array A in a transverse section be a minimum to a least-squares function

$$\mathcal{R}(A) = \sum_{\theta} \sum_{k=1}^{n_\theta} \frac{\left(P_{k\theta} - R_{k\theta}\right)^2}{\sigma_{k\theta}^2} \tag{14}$$

where the picture element values $A(i,j)$ satisfy the relationship

$$R_{k\theta} = \sum_{(i,j)\ \epsilon ray(k,\theta)} f_{ij}^{\theta} A(i,j) \tag{15}$$

and $\sigma_{k\theta}^2$ is the variance of the measured projection $P_{k\theta}$.
If $\mathcal{R}$ of Eq. (14) is minimized after incorporating Eq. (15) we have a solution for A in matrix form (cf. Appendix C for details)

$$\hat{A} = (F^T \Phi^{-1} F)^{-1} F^T \Phi^{-1} P, \tag{16}$$

where Φ^{-1} is the inverse of the covariance matrix and F is an $m \times N^2$ matrix composed of the weighting factors such as the fraction of the area of a particular picture element through which the ray passes as before;

and m is the total number of rays for all projections.

For the situation where we are limited in the number of views, the matrix $(F^T \Phi^{-1} F)$ is likely to be singular, thus threatening the existence of a solution to Eq. (16). This seemingly intractable problem might find for its solution the generalized inverse F^G of the matrix F, which in the formulism of Boullion and Odell (1971) gives the solution (see Appendix C for example)

$$\hat{A} = [(\Phi^{-1})^{1/2} F]^G (\Phi^{-1})^{1/2} P \quad . \tag{17}$$

Once the generalized inverse has been determined, the estimate $\hat{A}$ can be made by direct matrix multiplication as in Eq. (17). The generalized inverse is a function of the geometry of the object (imaging) space, the spatial change of the impulse response, ray divergence, if any, and photon attenuation. Thus in principle for a given imaging situation using projections at fixed but not necessarily equal angles, the generalized inverse matrix can be derived and used for digital or electronic hard-wired multiplication of the projection data.

The seemingly intractable problem of large matrix manipulations and insufficient number of projections available to give a unique solution, have led to iterative schemes for the approximation to a solution. To illustrate techniques developed further in Sections 3.3, 3.4, and 3.5, we solve Eq. (5) by an iterative approximation method whereby the value for each element A_i is guessed, and then modified by comparing the estimated projection value to the measured value. We start with the measured projections

A_1	A_2	$\Sigma = 3$
A_3	A_4	$\Sigma = 7$
$\Sigma = 4$	$\Sigma = 6$	

If we estimated each element had the mean value of 10/4, we would note that $A_1' + A_2' = 5$, which is 5/3 greater than the measured value, thus we make a second estimate at the value for A_1 and A_2 of 3/5 (10/4) = 6/4; this gives an array with the first row modified as

A_1 6/4	A_2 6/4	$\Sigma = 3$
A_3 10/4	A_4 10/4	$\Sigma = 5$
$\Sigma = 4$	$\Sigma = 4$	

Clearly the values of A_3 and A_4 need to be increased, because their sum deviates from the measured value by 5/7. After adjusting these values by 7/5 (10/4) = 14/4, we have

6/4	6/4	$\Sigma = 3$
14/4	14/4	$\Sigma = 7$
$\Sigma = 5$	$\Sigma = 5$	

The sums of the vertical rows need adjustment to coincide with the measured values; thus after the first iteration, we have

6/5	9/5	$\Sigma = 3$
14/5	21/5	$\Sigma = 7$
$\Sigma = 4$	$\Sigma = 6$	

which gives one solution. Even with the situation complicated by the noise of measurement, an approximation to the original distribution giving rise to the projections can be made by the iterative approach suggested above and other

algorithms outlined in this section. Before examining these schemes, let us review the simplest method of reconstruction.

3.3 Linear superposition or back-projection

The simplest and most rapid method of reconstituting a two-dimensional distribution from multiple one-dimensional projections is to merely project the views back to a common object region as depicted in Fig. 3. This technique

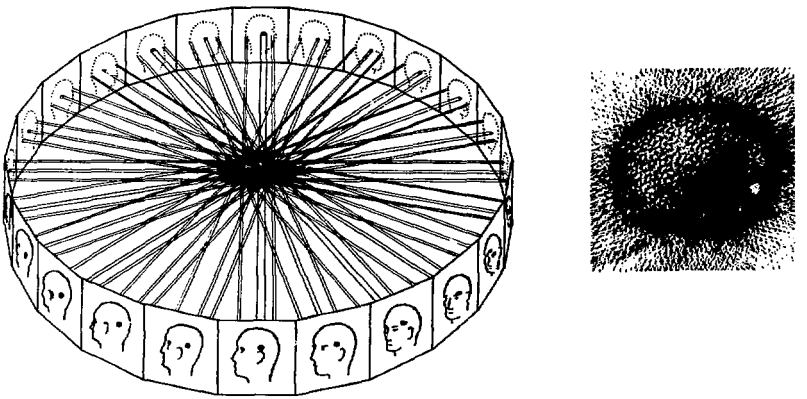


Figure 3. A transverse section is created by projecting the profiles from various views back through an image array. The technique is also known as superposition, summation, or simple transverse section scanning.

is basically that of conventional tomography or laminography implemented by analog methods of moving the imaging system relative to the object. This technique has been explored extensively in nuclear medicine applications since about 1956 under the name "transverse section scanning" (Kuhl and Edwards, 1964). Kuhl and co-workers used rectilinear scanners to obtain photopeak events as a function of distance along the projected line $P_{k\theta}$ in Fig. 4.

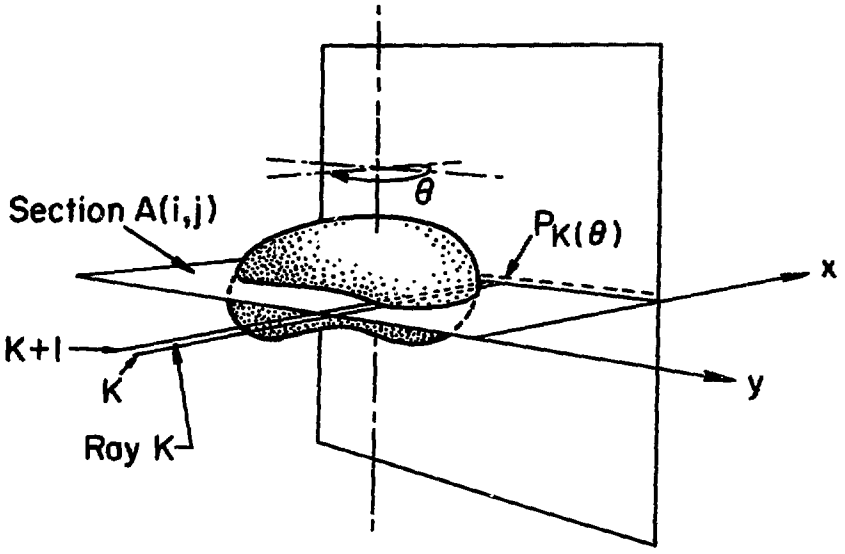


Figure 4. Relation between the section ray sums and a projection.

The strings of data for the multiple views were superposed on film by changing the angle of display relative to the film corresponding to the change in angle of view from one scan to the next. The result is the superposition of the projections. The first proposal for simple "transverse section scanning" using the gamma camera employed an optical technique for superpositioning the multiple camera views (Anger, 1967). The digital implementation of this technique was reported first by Kuhl (1966) for nuclear medicine, and more recently by Hart (1968) and Vainshtein (1970), the Russian crystalographer for electron microscopy. R. Hart's technique is similar to circular tomography. Harper (1968) presented a feasible method of three-dimensional image synthesis where any plane can be viewed by inserting an opaque screen in the field of back-projected three-dimensional images.

Though this technique is very simple, it cannot give the true radionuclide concentration even for an infinite number of projections. The resulting reconstruction will not equal the true image because each point in an image reconstructed using back-projection (Fig. 3) will be formed by the superposition of a set of straight lines corresponding to each projected ray from the true object. The superposition of a continuous set of lines around the point is equivalent to the rotation over a circumference of $2\pi r$ for the two-dimensional case and around a sphere of $4\pi r^2$ for the three-dimensional case. Thus the blurring function is $1/r$ or $1/r^2$ respectively, and the relation between the true object and the back-projected object is simply

$$\text{Back-projection} = \text{True} * 1/|r| \tag{18}$$

where * denotes a convolution. †

The operation of back-projection or linear superposition is described mathematically for a continuous series of projections $P(x, \theta)$ as

$$B(r,\phi) = \int_0^{\pi} P[r\cos(\phi-\theta),\ \theta]\, d\theta, \tag{19}$$

---

† Footnote: Proof of Eq. (18)
The projection theorem (see Appendix A and Section 3.7) gives the relation between the fourier transform $\tilde{A}(R, \theta)$ of the image and the projections

$$P(x, \theta) = \int_{-\infty}^{\infty} \tilde{A}(R, \theta)\, e^{i\, 2\pi R \cdot x}\, dR.$$

Using Eq. (19) we have,

$$B(r, \phi) = \int_0^{2\pi} \int_0^{\infty} R^{-1}\, \tilde{A}(R,\theta) \exp\left[i2\pi\, r \cdot R\cos(\phi-\theta)\right] R\, dR\, d\theta$$

This can be rewritten

$$B(r, \phi) = \mathcal{F}^{-1}\{R^{-1}\, \tilde{A}(R, \theta)\} = \mathcal{F}^{-1}\{R^{-1}\} * A(r, \phi)$$

The inverse transform of R^{-1} is r^{-1} as detailed in Appendix A.

where B is the back-projected image (Fig. 5). In almost every practical

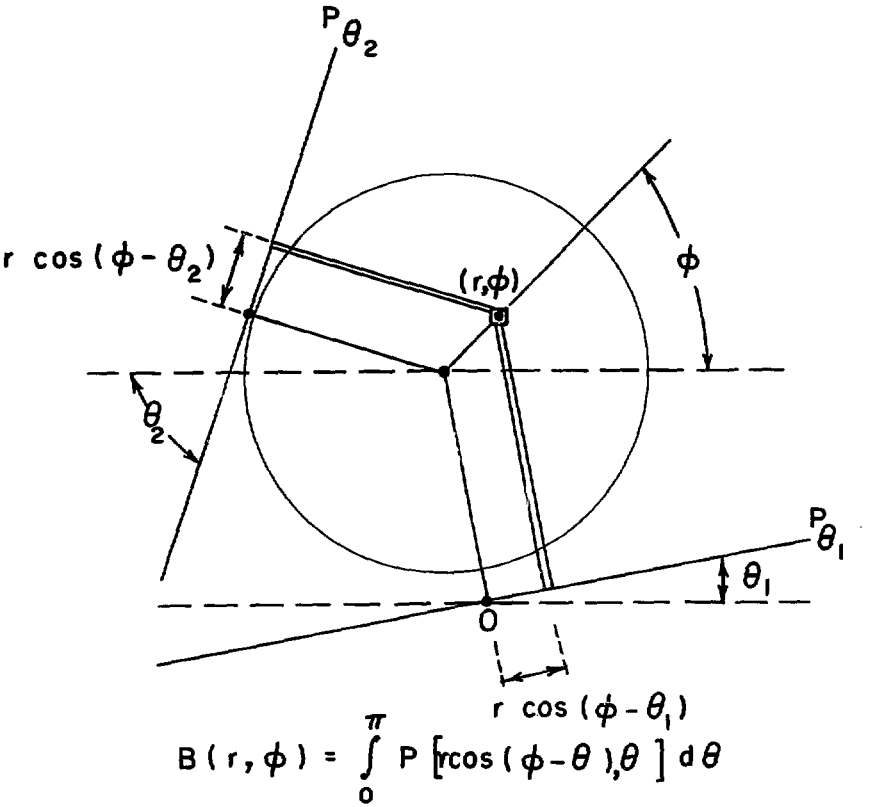


Figure 5. The value of a picture element (r,ϕ) after back-projection of rays from projections at multiple angles.

situation, we are dealing with a finite number of projections and for convenience of digital computation we use Cartesian coordinates. Thus, we describe the back-projection as

$$B'(i,j) = \sum_{\theta} P_{k(\theta)} \qquad (20)$$

where, for each element (i, j), we sum the contribution of each ray $k(\theta)$[†] which passes through the element. We let the total density or concentration, for the section or array being reconstructed, be estimated by

$$T = \sum_{k} P_{k\theta} \tag{21}$$

for any single projection. After back-projecting, the total density T' for the array is

$$T' = \sum_{i=1}^{n} \sum_{j=1}^{n} B'(i,j) \quad . \tag{22}$$

A normalization factor is derived for reducing the value of each picture element so that the reconstructed array total density corresponds to the estimated total given in Eq. (21). Thus the corrected back-projected image is

$$B(i,j) = B'(i,j) \cdot \frac{T}{T'} \qquad \text{for all } i,j\,. \tag{23}$$

A more exact background correction involves modifying the values by subtracting from each pixel the mean density or concentration multiplied by the number of views minus one (Vainshtein, 1971; Gilbert, 1972b). Thus,

$$B(i,j) = B'(i,j) - \frac{T(\text{no. of views} - 1)}{\text{no. of pixels}} \quad . \tag{24}$$

[†] The subscript $k(\theta)$ denotes a particular ray that passes through the particular pixel (i, j) that falls within the ray path. This unconventional notation is used to signify we are dealing with specific ray sums.

3.4 Algebraic reconstruction technique (ART)

A simple method of approaching a solution for the undetermined system of linear equations was outlined in Section 3.2. Gordon et al. (1970) applied this method to the reconstruction of a 50 x 50 digitized image from computed projections. The excellent results obtained with only a few views encouraged them and others to pursue techniques of iterative solution of the projection equations. The EMI scanner (Fischgold, 1973) employes a form of ART. The simple algorithm consists of guessing at a value for all the picture elements A(i, j), and then modifying each element along each ray by a factor that compensates for the discrepancy between the measured ray sum $P_{k(\theta)}$ and the calculated ray sum $R_{k(\theta)}$.

$$A^{n+1}(i,j) = A^{n}(i,j) \cdot \frac{P_{k(\theta)}}{R_{k(\theta)}} \tag{25}$$

If the calculated ray sum is the same as the measured value, it implies that the guessed values are correct for a particular projection; however, for another projection there might be a large discrepancy, thus the picture elements of the last view which lie in the ray for the new view will be modified according to the discrepancy between the new ray sum and the measured value. Thus, each ray from each projection is examined and values of A(i, j) falling within that ray are changed iteratively for all the projections for 5 to 10 iterations. Equation (25) is called multiplicative ART. Another method of correcting the discrepancy between the measured projections consists of adding the difference between the measured ray sum and the estimated ray sum. This is called the additive form of ART.

$$A^{n+1}(i,j) = \max \{A^{n}(i,j) + (P_{k(\theta)} - R_{k(\theta)})/N_{k(\theta)},\ 0\} \tag{26}$$

Here $N_{k(\theta)}$ is the number of pixels lying along the particular ray $k(\theta)$ which passes through pixel (i,j).

There are two important modifications of ART. One consists of setting to zero those values in the array that are clearly zero because they correspond to a ray sum that was observed as zero. This effecively bounds the data and is an important boundary condition for any of the iterative techniques. A third version of this technique known as ART3 incorporates noise and has been used effectively in transmission studies of phantoms and simulations with added noise (Herman, 1973; Johnson, et al., 1973).

3.5 Simultaneous iterative reconstruction technique (SIRT)

The simultaneous iterative reconstruction technique was developed by Gilbert (1972a) and differs from ART in that at each iteration the densities $A^n(i,j)$ are altered by using data from all of the projections simultaneously. Thus

$$A^{n+1}(i,j) = \max\left\{ A^n(i,j) + \frac{\sum_{\theta} P_{k(\theta)}}{\sum_{\theta} L_{k(\theta)}} - \frac{\sum_{\theta} R^n_{k(\theta)}}{\sum_{\theta} N_{k(\theta)}} \quad , \quad 0 \right\} , \tag{27}$$

where (i,j) represents the pixel which is an element of ray $k(\theta)$; $L_{k(\theta)}$ is the length of ray $k(\theta)$; $P_{k(\theta)}$ is the measured projected density of ray $k(\theta)$; $R_{k(\theta)}$ is the projected density of ray $k(\theta)$ after iteration n, i.e.,

$$R^n_{k(\theta)} = \sum_{(i,j) \in \text{ray } k(\theta)} A^n(i,j);$$

and $N_{k(\theta)}$ is the number of points in ray $k(\theta)$. This algorithm was used in our comparative studies. Gilbert also gave a multiplicative algorithm which is

$$A^n(i,j) = \max\left\{\frac{\sum_\theta P_{k(\theta)} \quad \sum_\theta N_{k(\theta)}}{\sum_\theta L_{k(\theta)} \quad \sum_\theta R_{k(\theta)}} \cdot A(i,j) \ , \ 0\right\} \ . \tag{28}$$

After each iteration the total array is normalized such that for all (i, j),

$$A^{n+1}(i,j) = T \ A^{n+1}(i,j)/T' \quad , \tag{29}$$

where

$$T = \sum_{k=1}^{n_1} P_{k1} \quad ,$$

and

$$T' = \sum_i \sum_j A^{n+1}(i,j) \quad ,$$

and where $A^{n+1}(i,j)$ are the values before the normalization. This normalization can be thought of as a type of damping as described in the next section for the least-squares algorithm.

For our implementation of SIRT, we choose the line length, $L_{k(\theta)}$, to be the maximum length of all lines that subtend the array between the lines k - 1 and k. The length of these line segments is determined by consideration of Fig. E-2 in Appendix E where the implementation of SIRT, along with a Fortran listing, is given.

3.6 Least-squares iterative technique

A least-squares iterative technique orginally proposed for determination of density distribution using proton or heavy-ion or conventional x-ray transmission scanning (Goitein, 1971; Boyd et al., 1974) has particularly appropriate attributes for emission studies. We emphasize this technique in

nuclear medicine because it accommodates noisy data, errors in data accumulation, and can be modified to handle attenuation errors due to emission studies. The derivation of this algorithm is based on minimizing the error between the measured projections and the estimated projections at the nth iteration in a least-squares fashion. First, note that

$$f^{\theta}_{i_o j_o} A^{n+1}(i_o, j_o) = R^{n}_{k(\theta)} - \sum_{\substack{(i,j) \in \text{ray } k(\theta) \\ \text{not } (i_o, j_o)}} f^{\theta}_{ij} A(i, j) \qquad \theta = 1, \ldots, M$$

where f^{θ}_{ij} is a weighting factor for geometry and attenuation. The requirement we impose is that

$$\mathcal{R}\left(A^{n+1}(i_o, j_o)\right) = \sum_{\theta} \frac{\left[P_{k(\theta)} - R^{n}_{k(\theta)}\right]^2}{\sigma_k{}^2(\theta)} \tag{30}$$

be a minimum. Therefore, we are improving the densities given in the previous iteration in a least-squares sense. The notation $k(\theta)$ is to indicate that k is chosen such that (i_o, j_o) is an element of ray $k(\theta)$. Equation (30) can be rewritten as

$$\mathcal{R}\left(A^{n+1}(i_o, j_o)\right) = \sum_{\theta} \left[P_{k(\theta)} - f^{\theta}_{i_o j_o} A^{n+1}(i_o, j_o) - \sum_{\substack{(i,j) \in \text{ray } k(\theta) \\ \text{not } (i_o, j_o)}} f^{\theta}_{ij} A^{n}(i, j)\right]^2 / \sigma^2_{k(\theta)} \tag{31}$$

Differentiating this with respect to $A^{n+1}(i_\bullet, j_\bullet)$ and setting equal to zero, we have

$$\frac{d\mathcal{R}}{dA^{n+1}(i_o, j_o)} = \sum_{\theta} 2 f^{\theta}_{i_o j_o} \left[P_{k(\theta)} - f^{\theta}_{i_o j_o} A^{n+1}(i_o, j_o) - \sum_{\substack{(i,j) \in \text{ray } k(\theta) \\ \text{not } (i_o, j_o)}} f^{\theta}_{ij} A^{n}(i, j)\right] / \sigma^2_{k(\theta)} = 0 .$$

Solving for $A^{n+1}(i_o, j_o)$, we have the following sequence of equations:

$$A^{n+1}(i_o,j_o) = \left\{ \sum_\theta f^\theta_{i_o j_o} \left[P_{k(\theta)} - \sum_{\substack{(i,j)\in \text{ray } k(\theta) \\ \text{not } (i_o,j_o)}} f^\theta_{ij} A^n(i,j) \right] / \sigma^2_{k(\theta)} \right\} \Big/ \sum_\theta \left[f^\theta_{i_o j_o} / \sigma_{k(\theta)} \right]^2$$

$$= \left\{ \sum_\theta f^\theta_{i_o j_o} \left[P_{k(\theta)} - R^n_{k(\theta)} + f^\theta_{i_o j_o} A^n(i_o, j_o) \right] / \sigma^2_{k(\theta)} \right\} \Big/ \sum_\theta \left[f^\theta_{i_o j_o} / \sigma_{k(\theta)} \right]^2$$

$$= A^n(i_o,j_o) + \left\{ \sum_\theta f^\theta_{i_o j_o} \left[P_{k(\theta)} - R^n_{k(\theta)} \right] / \sigma^2_{k(\theta)} \right\} \Big/ \sum_\theta \left[f^\theta_{i_o j_o} / \sigma_{k(\theta)} \right]^2$$

$$\Delta^n A(i_o,j_o) = \left\{ \sum_\theta f^\theta_{i_o j_o} \left[P_{k(\theta)} - R^n_{k(\theta)} \right] / \sigma^2_{k(\theta)} \right\} \Big/ \sum_\theta \left[f^\theta_{i_o j_o} / \sigma_{k(\theta)} \right]^2 \quad . \tag{32}$$

3.6.1 Damping Factor

If we now use Eq. (32) to correct the densities for each iteration we will find that the densities do not converge, but oscillate, because Eq. (32) corrects the previous density $A^n(i_o,j_o)$ based on the previous ray sum $R^n_{k(\theta)}$. A simple example of this can be illustrated by the following 4×4 array

$\frac{1}{4}P$	$\frac{1}{4}P$	P
$\frac{1}{4}P$	$\frac{1}{4}P$	P
P	P	

where $\frac{1}{4}P$ is the estimated pixel density for some iteration n and P is the measured projection. Now, if we assume $f^\theta_{ij} = 1$ for all θ, i, j and $\sigma^2_{k\theta} = P_{k\theta}$, then $\Delta^n A(i,j)$ can be evaluated

$$\Delta^n A(i,j) = \left[\sum_\theta (P_{k(\theta)} - R^n_{k(\theta)}) / P_{k(\theta)} \right] / \left[\sum_\theta 1 / P_{k(\theta)} \right]$$

$$= [(P - 1/2\,P) / P + (P - 1/2\,P) / P] / [1/P + 1/P]$$

$$= P/2 \ .$$

This then gives a new density $A^{n+1}(i,j) = 3P/4$ and $\Delta^{n+1}A(i,j)$:

$$\Delta^{n+1}A(i,j) = [(P - 3/2\,P) /P + (P - 3/2\,P) /P] /[1/P + 1/P]$$

$$= -P/2 \ .$$

Therefore, we have $A^{n+2}(i,j) = P/4$. If we continue this we would have the alternating sequence for each pixel density:

$$P/4 \ , \ 3P/4 \ , \ P/4 \ , \ \ldots .$$

$$A^{n}(i,j) \ , \ A^{n+1}(i,j) \ , \ A^{n+2}(i,j) \ , \ \ldots$$

So a damping factor is required which will be a function of all changes for each pixel. Therefore, once $\Delta^{n}A(i,j)$ has been evaluated for all i,j, then a damping factor δ must be determined such that

$$A^{n+1}(i_o,j_o) = A^{n}(i_o,j_o) + \delta\ \Delta^{n}\ A(i_o,j_o) \ , \tag{33}$$

where

$$\Delta^{n}A(i_o,j_o) = \Big[\sum_{\theta} f^{\theta}_{i_o j_o} \left(P_{k(\theta)} - R^{n}_{k(\theta)}\right) / \sigma^{2}_{k(\theta)}\Big] \Big/ \sum_{\theta} \left(f^{\theta}_{i_o j_o} / \sigma_{k(\theta)}\right)^{2} . \tag{34}$$

If we require that

$$\mathcal{R}(\delta) = \sum_{\theta}\sum_{k} \frac{\left(P_{k\theta} - R^{n+1}_{k\theta}\right)^{2}}{\sigma^{2}_{k\theta}} \tag{35}$$

be a minimum, then we are also choosing a damping factor in a least-squares sense. We can rewrite Eq. (35) as

$$\mathcal{R}(\delta) = \sum_{\theta}\sum_{k}\left\{P_{k\theta} - \sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\left[A^{n}(i,j) + \delta\Delta^{n}A(i,j)\right]\right\}^{2}/\sigma_{k\theta}^{2} \quad .$$

Differentiating $\mathcal{R}$ with respect to δ and setting the derivative equal to zero, we have the following equation:

$$\frac{d\mathcal{R}(\delta)}{d\delta} = \sum_{\theta}\sum_{k} 2\left\{P_{k\theta} - \sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\left[A^{n}(i,j) + \delta\Delta^{n}A(i,j)\right]\right\}\sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\Delta^{n}A(i,j)/\sigma_{k\theta}^{2}$$

This implies that

$$\sum_{\theta}\sum_{k}\left[P_{k\theta} - \sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta} A^{n}(i,j)\right]\cdot \sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\Delta^{n}A(i,j)/\sigma_{k\theta}^{2}$$

$$- \delta \sum_{\theta}\sum_{k}\left[\sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\Delta^{n}A(i,j)\right]^{2}/\sigma_{k\theta}^{2} = 0 \quad .$$

Therefore,

$$\delta = \frac{\sum_{\theta}\sum_{k}\left(P_{k\theta} - R_{k\theta}^{n}\right)\left[\sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\Delta^{n}A(i,j)\right]/\sigma_{k\theta}^{2}}{\sum_{\theta}\sum_{k}\left[\sum_{(i,j)\epsilon ray(k,\theta)} f_{ij}^{\theta}\Delta^{n}A(i,j)\right]^{2}/\sigma_{k\theta}^{2}} \qquad (36)$$

Now, if one were to apply the damping factor to the previous example, the sequence of densities would be

$$\frac{P}{4} \ , \ \frac{3P}{8} \ , \ \frac{7P}{16} \ , \ \ldots\ldots \rightarrow \frac{P}{2}$$

$$A^n(i,j)\ ,\quad A^{n+1}(i,j)\ ,\quad A^{n+2}(i,j),\ldots\ldots$$

3.6.2 Statistical noise

For our work, we have chosen the standard deviation $\sigma_{k\theta}$ to be the square root of the counts for each projected ray. Using the previous notation we can express this as $\sigma_{k\theta}{}^2 = P_{k\theta}$, and then Eqs (32) and (36) reduce to

$$\Delta^n A(i_o,j_o) = \left[\sum_\theta f^\theta_{i_o j_o}\left(1 - R^n_{k(\theta)}/P_{k(\theta)}\right)\right] \Big/ \sum_\theta \left(f^{\theta 2}_{i_o j_o}/P_{k(\theta)}\right) \quad ,$$

and

$$\delta = \frac{\sum_\theta \sum_k \left(1 - R^n_{k\theta}/P_{k\theta}\right) \left[\sum_{(i,j)\epsilon ray(k,\theta)} f^\theta_{ij}\, \Delta^n A(i,j)\right]}{\sum_\theta \sum_k \left[\sum_{(i,j)\epsilon ray(k,\theta)} f^\theta_{ij}\, \Delta^n A(i,j)\right]^2 / P_{k(\theta)}} \tag{37}$$

3.7 Filtered back-projection or convolution techniques

Recall the relationship between the true image and the image obtained from the linear superposition or back-projection of an infinite number of views Eq. (18):

$$B(x,y) = \iint \frac{A(x',y')}{\{(x-x')+(y-y')\}^{1/2}}\, dx'\, dy' = A * r^{-1} \tag{38}$$

We seek a technique whereby $\frac{1}{r}$ can be deconvoluted from $B(x,y) = B(r,\phi)$ From the convolution theorem

$$\mathcal{F}\{B(r,\phi)\} = \mathcal{F}\{A(r,\phi)\} \cdot \mathcal{F}\{r^{-1}\} \quad ; \tag{39}$$

where

$$\mathcal{F}\{r^{-1}\} = 2\pi \int r^{-1} J_o(2\pi R r)\, r\, dr \tag{40}$$

$$= R^{-1}$$

where R is the reciprocal space radius or the measure in frequency space. Thus, the true image is related to the back-projection image as

$$A(r,\phi) = \mathcal{F}^{-1}\left\{\mathcal{F}\{A(r,\phi)\}\right\} = \mathcal{F}^{-1}\left\{|R|\,\mathcal{F}\{B(r,\phi)\}\right\} \tag{41}$$

A similar result is obtained by Bates and Peters (1971) using a perhaps more rigorous derivation that we circumvented by use of the identity Eq. (40). This shortcut does not recognize the real situation in which there are finite bounds on the domain of integration where Eq. (40) does not hold. We will return to this problem later. For the present assume we have an infinite number of projections and the data are not band-limited. The operations of Eq. (41) involve the following steps:

(1) Obtain a series of projections.

(2) Derive the back-projected image by simple linear superposition (Appendix D).

(3) Fourier transform the two-dimensional image.

(4) Multiply the Fourier coefficients by the spatial frequency radius.

(5) Fourier transform (invert) the result of (4) to obtain the true image.

This procedure can be done optically (Peters, 1973) or digitally. Two-dimensional Fourier transforms can be accomplished in less than one minute for 64×64 arrays on small 16-bit computers (Budinger and Harpootlian, 1973), but for 128×128 arrays, much more time is involved. The fast Fourier transform algorithm limits the array size to integer powers of 2, thus we cannot do 80×80 or 100×100 arrays by this method unless zeroes are added to expand the array to 128×128. This is not a serious limitation for nuclear medicine, but becomes important for transverse-section radiography using transmission where the resolution is four or more times better than for emission studies. Thus we seek a method which is computationally more convenient. Methods of

modifying the projection vectors before back-projecting are both convenient in terms of computer space, and very fast.

The true value of each pixel is related to the Fourier coefficients as

$$A(r,\phi) = \int_0^{2\pi}\int_0^{\infty} \tilde{A}(R,\theta) \exp\left[i2\pi R r \cos(\phi-\theta)\right] R \, dR \, d\theta \quad . \tag{42}$$

Note that $\tilde{A}(R,\theta) = \tilde{A}(-R,\theta + \pi)$, thus Eq. (41) can be written as

$$A(r,\phi) = \int_{-\infty}^{+\infty}\int_0^{\pi} |R| \, \tilde{A}(R,\theta) \exp\left[i2\pi R r \cos(\phi-\theta)\right] dR \, d\theta \quad . \tag{43}$$

If we define

$$\mathcal{F}^{-1}\{|R| \, \tilde{A}(R,\theta)\} = \tilde{P}(r,\theta) \quad , \tag{44}$$

then the true pixel values become

$$A(r,\phi) = \int_0^{\pi} \tilde{P}\left(r\cos(\phi-\theta), \vartheta\right) d\theta \quad . \tag{45}$$

But, recall the back-projection operation Eq. (19), which indicates the operation of Eq. (45) is linear superposition of projections $\tilde{P}$. What are the physical interpretations of $\tilde{P}$, P, and $\tilde{A}$? $\tilde{A}(R, \theta)$ is the Fourier component at the reciprocal space position (R, θ). The projection theorem† equates the inverse Fourier transform of $\tilde{A}$ with respect to the real space values of the projection normal to the line, θ = constant. Thus, it is easy to see that P is the projection value associated with $\tilde{A}$, and $\tilde{P}$, therefore, is the result of modifying the projection by a ramp filter since

$$\mathcal{F}^{-1} \{|R| \mathcal{F}(P)\} = \tilde{P} \quad . \tag{46}$$

With these considerations, the relation between the projections and the true image can be deduced. Namely, the true image can be reconstructed by back-projecting the projections after they have been modified in accordance with Eq. (46).

---

† Fourier transform of the projection gives the components along the section in Fourier space normal to the projection:
A(x, y, z) is a three-dimensional distribution, and the two-dimensional projection is defined as

$$P(x, y) = \int A(x, y, z)\, dz \quad .$$

The three-dimensional Fourier transform is

$$\tilde{A}(X, Y, Z) = \iiint A(x, y, z) \exp[-i2\pi(x \cdot X + y \cdot Y + z \cdot Z)]\, dx\, dy\, dz \quad ;$$

for Z = 0 we have

$$\tilde{A}(X, Y, Z) = \iint \left\{ \int A(x, y, z) dz \right\} \exp[-i2\pi(x \cdot X + y \cdot Y)]\, dx\, dy$$

$$\equiv \iint P(x, y) \exp[-i2\pi(x \cdot X + y \cdot Y)]\, dx\, dy \quad .$$

Q. E. D.

See Appendix A for the two-dimensional projection theorem.

There are two ways of implementing Eqs. (44) and (45). One is known as the convolution technique and the other as the filter technique. Both are equivalent as can be seen by the following:

The Fourier transform of the function $|R|$ is not defined unless one imposes an upper bound R_m which is the maximum meaningful frequency which can be reconstructed:

$$g(r) = \int_0^{R_m} |R| \exp(i2\pi R r)\, dR \quad . \tag{47}$$

Integration by parts gives

$$= \frac{R_m}{\pi r} \sin(2\pi R_m r) - \frac{1}{2(\pi r)^2} [1 - \cos(2\pi R_m r)] \tag{48}$$

The function $g(r)$ convoluted with the projection gives the modified projection. If the projection function is band limited to R_m, then Eq. (48) becomes identical to that derived by Bracewell and Riddle (1967):

$$[\delta(r) - R_m \operatorname{sinc}^2 (R_m r)] \quad . \tag{49}$$

If the data are sampled at equal intervals, which are integer multiples of $\frac{1}{2R_m}$, where $1/(2R_m) = a$ is the sampling interval chosen small enough to avoid aliasing. The sampling theorem requires $\frac{1}{R_m} = 2a$. Ramachandran and Lakshminarayanam (1971) arrive at a somewhat similar expression deduced

from evaluation of the integral Eq. (47) between some large values $-R_m$ to $+R_m$ (see footnote‡)

$$g(na) \begin{cases} 1/4a^2 & n = 0 \\ -1/(na)^2 & n \text{ odd} \\ 0 & n \text{ even} \end{cases} . \tag{50}$$

These techniques are equivalent to the direct application of a ramp filter to the Fourier components of the projection values, which can be seen by recalling the convolution theorem, since

---

‡ The Fourier series of the function $|R|$ between $-R_m$ and R_m is

$$|R| = \sum_{\infty}^{\infty} C_n \exp(in\pi R/R_m) ,$$

where

$$C_n = \frac{1}{2R_m} \int_{-R_m}^{R_m} |R| \exp(-\pi i Rn/R_m) \, dR .$$

For $n = 0$,

$$C_n = \frac{1}{R_m} \int_0^{R_m} R \, dr = \frac{1}{R_m} \frac{R^2}{2} \Big|_0^{R_m} = \frac{R_m}{2} ,$$

otherwise

$$C_n = \frac{R_m}{2\pi^2 n^2} \left(\exp(-\pi in) + \exp(\pi in)\right) - \frac{R_m}{\pi^2 n^2} .$$

Thus,

$$C_n = \begin{cases} R_m/2 & \text{if } n = 0 \\ -2R_m/\pi^2 n^2 & \text{if } n \text{ is odd} \\ 0 & \text{if } n \text{ is even} \end{cases}$$

Note from the sampling theorem $g(na) = g(n/2R_m) = 2R_m C_n$. This gives Eq. (50).

$$
\begin{aligned}
\tilde{P} &= g * P \ , \\
\mathcal{F}(\tilde{P}) &= \mathcal{F}(g) \cdot \mathcal{F}(P) \ , \\
\mathcal{F}(\tilde{P}) &= |R| \cdot \mathcal{F}(P) \ , \\
\tilde{P} &= \mathcal{F}^{-1}[\mathcal{F}(\tilde{P})] = \mathcal{F}^{-1}[|R| \cdot \mathcal{F}(P)] \quad . \qquad (51)
\end{aligned}
$$

Note the image reconstruction is completed by back-projection of the projections modified according to Eq. (51).

3.8 Studies with phantoms and patients

3.8.1 Scope

The methods outlined in the preceding sections and implemented in the appendix are

Back-projection	Appendix D
SIRT	Appendix E
Least-squares iterative technique	Appendix F
Filtered back-projection	Appendix G

The results of both phantom and patient studies using these four techniques are presented in this section. Attenuation is discussed in Sec. 3.9 and Appendix H.

3.8.2 Methods of data acquisition

3.8.2.1 Scintillation camera procedures

Studies were done by rotating the subject in front of the Anger scintillation camera (Anger, 1967). The 16-inch scintillation camera was used with a technetium low-energy parallel-hole collimator and a special extended collimator constructed at Donner Laboratory. The extended collimator consists of a rectangular array of aluminum tubes of 0.15 mm wall thickness and 12.7 cm long. These tubes are stacked in the natural hexagonal close-packed fashion.

The camera-collimator arrangement is positioned vertically to accommodate the rotation of the patient or phantom around a vertical axis in front of the camera. The subject is positioned as close to the collimator as possible, because the resolution deteriorates with distance from the collimator. In some trials the patient was fixed relative to the rotation axis by a head holder connected to a chair, which is rotated on a stage and stopped at intervals of 10°. This was found to be cumbersome. A more satisfactory procedure was head fixation by a mouthpiece bite arrangement shown in Fig. 6. At present there is no convenient method for rotating the patient around a fixed axis. The ideal method involves rotation of the camera around the patient, as suggested in Fig. 3.

Figure 6. The technique of rotating a patient before the scintillation camera: The patient is rotated manually with head support provided by a bite mouth piece which is mounted on the rotation stool.

3.8.2.2 Digital data management

At each angle, 50,000 to 100,000 events are collected on the Hewlett-Packard digital system HP-5407 (Budinger, 1973). These events are digitized in 64×64 arrays, and stored as a histogram for later processing. The viewing time is usually 15 or 30 seconds, and each frame is stored sequentially around 360° at 10° increments. The 36 frames are held on a disc frame file, which has a total capacity of 160, 64×64 frames. We use a slight zoom or gain on the analog-to-digital converter (ADC) so as to concentrate the digitizing process on that part of the crystal, which sees the head. Thus the space between raster points represents 5.5 mm. For the reconstruction, three rows (3×5.5 = 16.5 mm) are selected from each of the 36 frames, and these become the projections for reconstruction of slices or transverse sections. Normally eight sections are taken at 16.5-mm intervals from the head vertex to about the nasion-meatal level (8×16.5 mm = 13.2 cm).

3.8.3 Reconstruction procedures

The procedures for back-projection (Appendix D), SIRT (Appendix E), and filtered (compensated) back-projection (Appendix G) involve forming 18 projections, which are the square roots of the product of the conjugate views,

(geometric mean), π radians apart. This procedure in part compensates for attenuation and the change in impulse response with distance from the collimator. Note that the true activity of a point source in the attenuation midline of thickness T is

$$\left\{P_{k(\theta)} \cdot P_{k(\theta+\pi)}\right\}^{\frac{1}{2}} e^{\mu T/2} \qquad (52)$$

where μ is the attenuation coefficient; and the true activity of a distributed source is given by

$$\frac{[P_{k(\theta)} P_{k(\theta+\pi)}]^{\frac{1}{2}} e^{\mu T/2} f T}{2 \sinh(\mu f T/2)} , \qquad (53)$$

where δ is the effective fraction of the thickness occupied by the source. This parameter can vary from 0.1 to 0.9 without seriously affecting the result (Sorenson, 1971).

Eighteen views derived as simple conjugate means or modified by correction for attenuation are used as the input to the reconstruction program. The correction of Eqs. (52) or (53) is not adequate, because it requires a priori knowledge of the thickness and an assumption about the attenuation coefficient. A refined technique for the attenuation correction has been incorporated in the least-squares technique (Appendix H).

The complete procedure for the attenuation correction iterative least-squares technique involves use of the 18 conjugate means for 3 to 5 iterations, after which the algorithm for ascertaining an outline of the object is applied. Using this outline, the distance l_{ij}^{θ} between each pixel and the object edge along a ray is calculated, and this gives the parameter f_{ij}^{θ} defined as

$$f_{ij}^{\theta} = e^{-\mu l_{ij}^{\theta}} \qquad (54)$$

where μ is the linear attenuation coefficient. The values of f_{ij}^{θ} are then incorporated into Eq. (32) and a few more iterations are completed using all 36 views to give the solution as discussed again in the section on attenuation.

3.8.4 Hot-spot detection

A comparison of the ability of these techniques to reconstruct hot spots in an 18-cm diameter lucite disc having a hot annulus around the periphery is shown in Fig. 7. Clearly the least-squares and filtered back-projection techniques are superior to the back-projection and SIRT techniques. Ten to 20 iterations for both SIRT and the iterative least squares were made. The

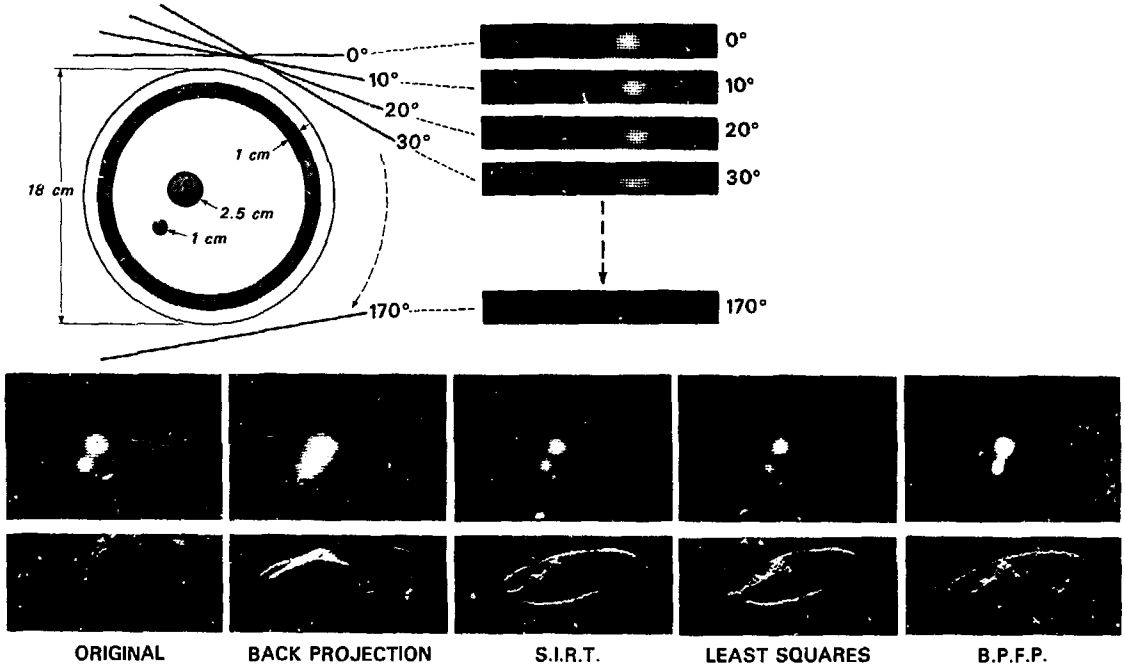


Figure 7. Comparison of four techniques for three-dimensional reconstruction—back-projection, simultaneous iterative reconstruction, least-squares, and back-projection of filtered projections (BPFP)—using a lucite phantom and ^{99m}Tc.

ability of these techniques to reconstruct the original density is made by evaluating the root mean square of the normalized difference between the true distribution and the reconstructed distribution after each iteration as

$$\text{Discrepancy} = \left\{ \frac{\sum_{i,j} [A'(i,j) - A^n(i,j)]^2}{\sum_{i,j} [A'(i,j) - A^0(i,j)]^2} \right\}^{1/2} \tag{55}$$

where $A'(i,j)$ is the true value, $A^0(i,j)$ is the value for the initial solution, and $A^n(i,j)$ is the value after the nth iteration. The results are shown in Fig. 8 for two objects.

3.8.5 Cold-spot detection

One of the problems of conventional tomography is detection of cold spots deep within hot tissue, for an example, detection of a tumor within the liver. The ability to reconstruct holes with the same resolution as hot spots is demonstrated in Fig. 9, where a simulated liver slice was reconstructed. The hole 1.25 cm in diameter can be seen above noise in the reconstructions using the techniques of iterative least squares and the filtered back-projection.

3.8.6 Patient studies

Four patient studies have been made including one adult with parietal-occipital abnormal accumulation (Fig. 10), and one probable-normal 14-year-old child who could have a craniopharyngioma (Fig. 11). The abnormal accumulation in Fig. 9 could be from hemorrhage, tumor, or granulomatous disease. In the child with suspected craniopharyngioma, the hot activity is in the region of the cavernous sinus and might be a normal finding. Quantitative brain scanning by this technique relies heavily on the ability to correct for attenuation, as alluded to earlier and explored in detail in the next section.

3.9 Attenuation

In emission studies the contribution of each volume element to the projection ray sum is not a simple additive factor, as in the case in transmission studies. Each element contributes a photon emission concentration, $\gamma/\text{sec-cm}^3$, which is attenuated by the path length between each point and the edge of the object along a projection ray. Thus the activity measured along one projection view will be significantly different from the activity measured in the conjugate view 180° from the first view if the distribution of activity is asymmetric. For transmission studies the projected ray sum is

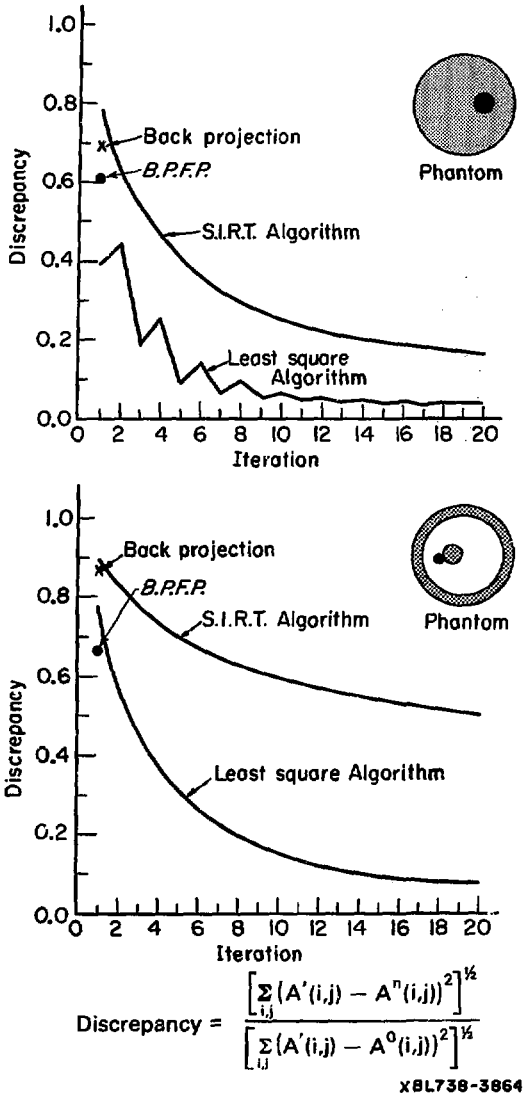


Figure 8. Comparison of the accuracy of reconstruction by back-projection, SIRT, least-squares, and filtered back-projection (BPFP) techniques for the phantoms as shown.

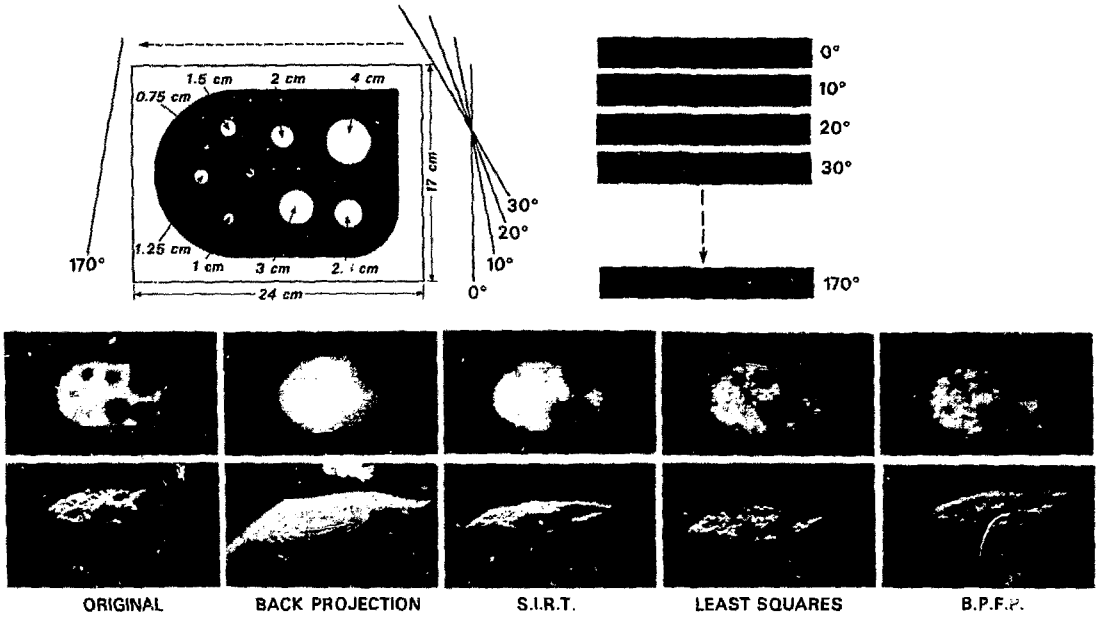


Figure 9. Comparison of the ability of the four techniques for three-dimensional reconstruction to detect small holes in a liver phantom slice filled with ^{99m}Tc (shaded region).

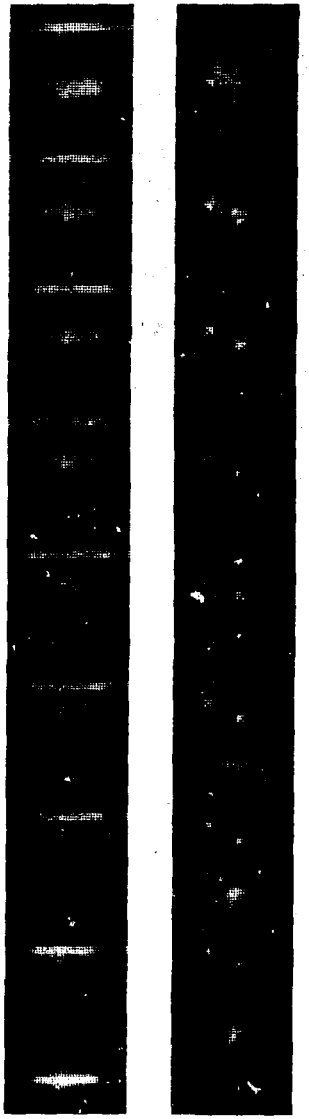


Figure 10. Patient study showing the transverse sections and on the left, the corresponding rows of data flagged from views taken at 10° increments. An abnormal accumulation of technetium is shown in the parietal-occipital area of this patient with an as yet undiagnosed pathology.

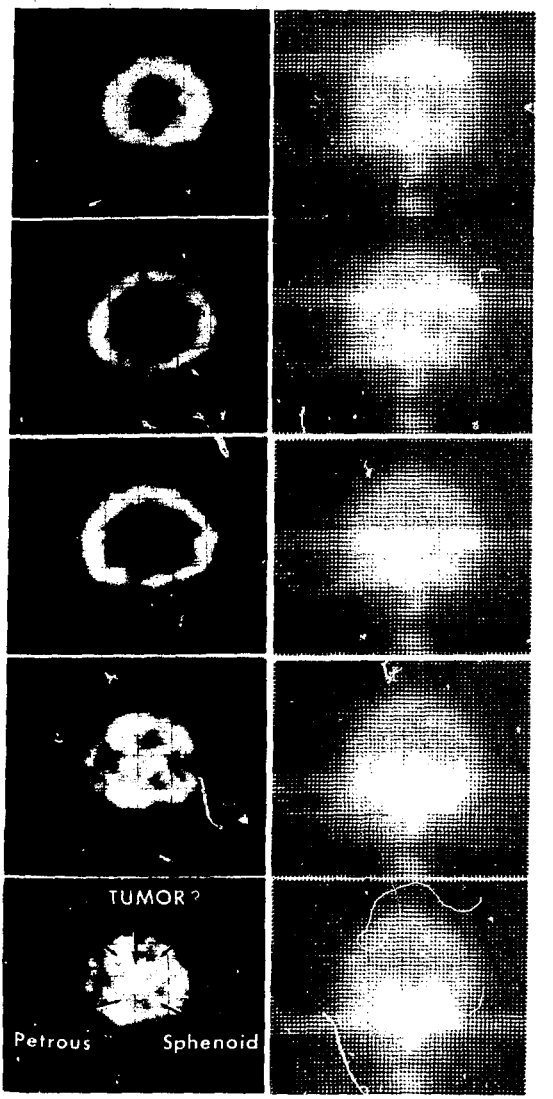


Figure 11. Patient study revealing possible abnormal accumulation of technetium in midbrain or cavernous sinus region.

$$-\log \frac{P_{k(\theta)}}{P^0_{k(\theta)}} = \sum_{i,j \in \text{ray}\, k(\theta)} \ell_{ij}\mu_{ij} , \qquad (56)$$

where ℓ_{ij} is the length of the element with attenuation coefficient μ_{ij}. However, for emission studies

$$P_{k(\theta)} = \sum_{i,j \in \text{ray}\, k(\theta)} A(i,j) \exp[-\Sigma\, \mu(\alpha,\beta)\ell(\alpha,\beta)] . \qquad (57)$$

Thus the nuclear medicine reconstruction problem is more difficult than the transmission problem. The influence of the term $\exp[-\Sigma\, \mu(\alpha,\beta)]$ depends on the attenuation coefficient μ, which unfortunately is so large for all energies used in nuclear medicine that the reconstructed images are seriously affected.

The effects of attenuation can be seen from Fig. 12; where the effects of

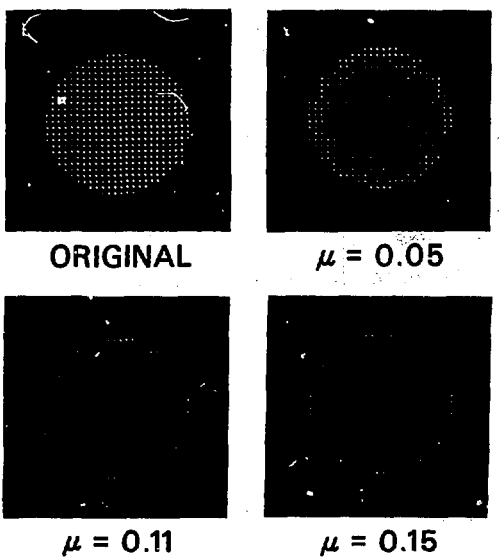


Figure 12. Comparison of the reconstructed transverse section for various photon energies (attenuation coefficients) if no attenuation compensation is made.

gamma rays of a few MeV ($\mu = 0.05$) are compared to the effects of gamma rays of 511 keV ($\mu \approx 0.11$) and 140 keV ($\mu = 0.15$). The images in the lower row show the serious artifact that will result for the usual isotopes used in nuclear medicine if attenuation is not taken into account. The source is a disc of 23-cm diameter similar in size to a section through the head. Another way of evaluating the seriousness of this problem is shown in Fig. 13 where the profiles through the reconstructed disc are given. The deviation of the reconstructed image from the true image is shown (Fig. 14) in terms of the fraction of the total number of pixels that depart by multiples of the standard deviation from the true value. Here the standard deviation is taken to be the square root of the true pixel value.

The quantitative three-dimensional reconstruction of gamma-emitter distribution in the head and other parts of the body requires compensation for attenuation. The chest presents the extreme case, and unfortunately cradles the heart, which is a most important region for application of these techniques. We envision six methods for solving this problem.

1) The simplest technique involves application of a correction matrix to the results of the reconstruction. The correction matrix consists of correction factors determined from phantom studies and assumes a fixed geometry for all studies and a constant or an assumed distribution of attenuation coefficients.

2) We would like to be free of constraints on geometry, and have adapted the second method, which entails assumption of constant attenuation coefficient and calculation of the attenuation path length [l_{ij} of Eq. (54)] between each pixel and the edge of the object along each ray. The shape of the object is estimated after a few iterations by employing the subroutine SEARCH of Appendix H. This procedure gives good results for objects with a constant linear attenuation coefficient such as the brain. A comparison of the least-squares procedure with and without attenuation is shown in Fig. 15. Without this or some other techniques discussed below, the results of the reconstruction will be nonquantitative and lead to artifacts such as the ring distribution derived from the disc section, as shown in Fig. 12.

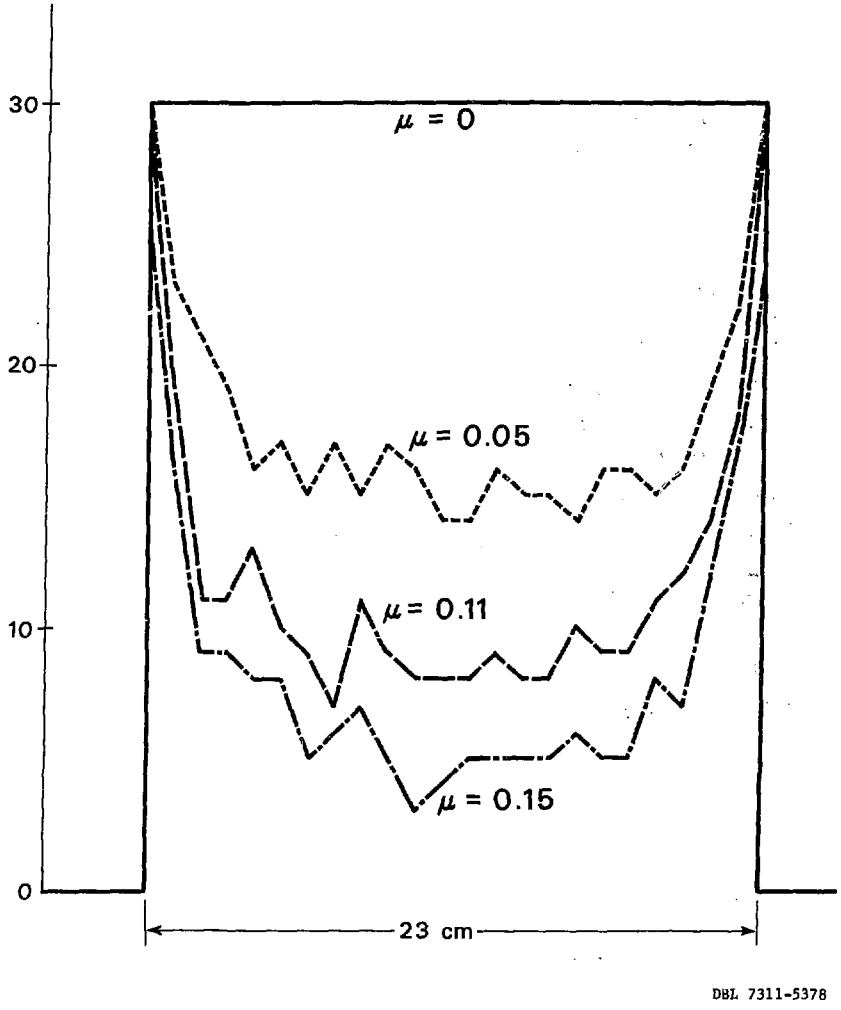
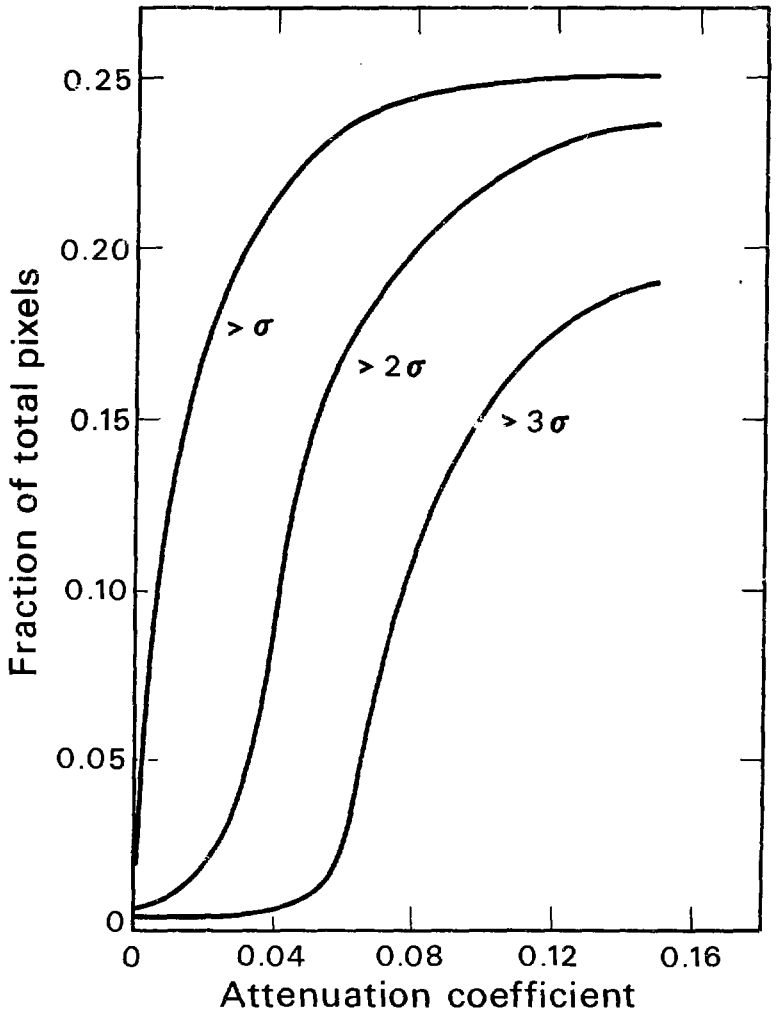


Figure 13. Profiles through sections of Fig. 12.



DBL 7311-5377

Figure 14. Fraction of pixels that deviate by 1, 2, and 3 standard deviations from the true values in a reconstruction if attenuation is not taken into account.

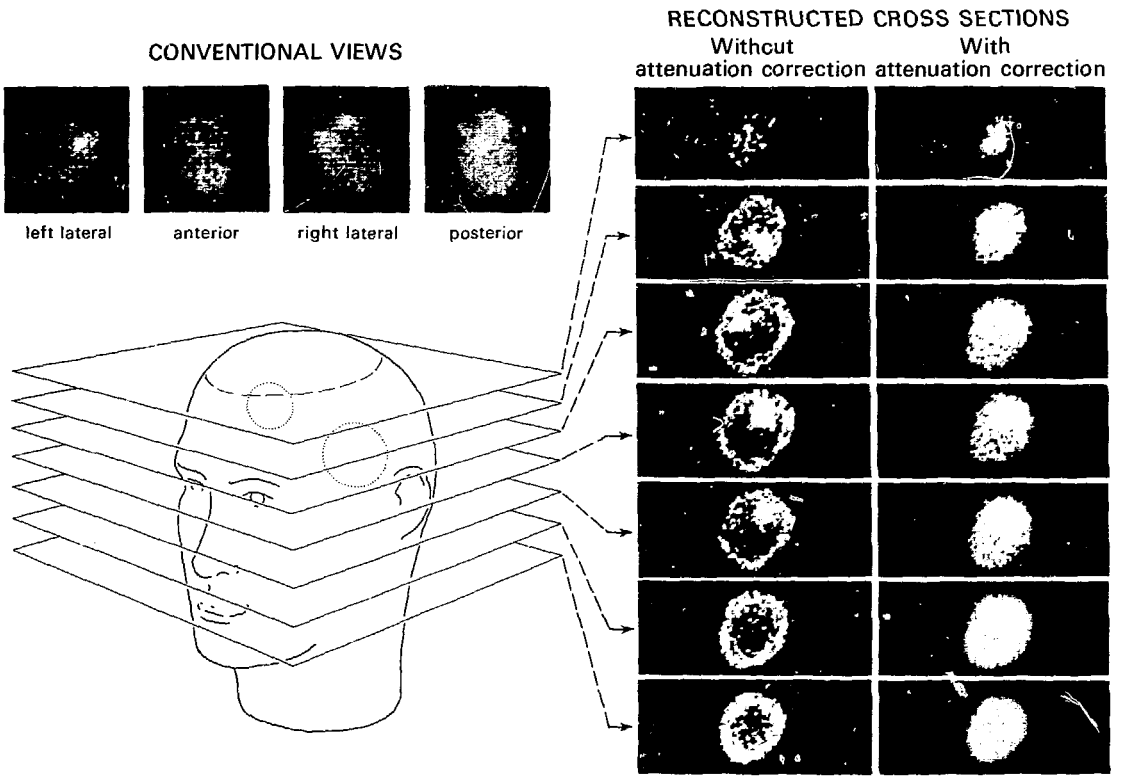


Figure 15. A comparison of transverse sections with and without attenuation compensation using the least-squares technique.

3) Another method of attenuation correction that can be applied to the last iteration of the iterative techniques or to the results of the filtered back-projection involves recalculating the projected data that would have occurred if there were no attenuation. The outline of the region of interest is automatically determined by a simple computer search routine (Appendix H). The correction is made for each ray by multiplying the geometric mean of conjugate views by a factor,

$$e^{\mu T} \cdot (f\mu T/2)/\sinh(f\mu T/2),$$

where μ is the linear attenuation coefficient, T is the thickness of the body section along the conjugate view ray, and f is some factor that varies between 0.2 and 0.8, depending upon the fractional distribution of isotope. A large change in this estimated parameter does not affect the solution significantly.

4) A fourth technique involves iteration between the algorithm for determination of the value of each pixel and the algorithm for determining f_{ij}^{θ}, which is the correction factor for attenuation. For example, ART could be used to determine the estimate of the concentration in a section for a few iterations, then the concentration fixed for a few iterations where f_{ij}^{θ} values are determined. We have not yet pursued this interesting approach, which was suggested by Dr. Richard Gordon.

5) The true distribution of attenuation coefficients can be determined by transmission measurements, as suggested by Eq. (56). The usual technique involves measuring the ratio of transmitted-to-incident photons. Thus, to estimate the distribution of bone and soft tissue in the chest, a transmission study would be done before the emission study with the Anger camera. The source could be ^{57}Co or ^{99m}Tc and, with proper tuning, ^{241}Am.

6) The last technique involves use of multiple isotopes where advantage is taken of the known different absorption coefficients of various tissues for different photon energies. For example, if ^{210}Pb (40 keV), ^{241}Am (60 keV) and ^{99m}Tc (140 keV) were used, we can determine the distribution of tissue such as lung, bone, and soft tissue by noting

$$-\log \frac{I}{I_0}(^{210}\mathrm{Pb}) = \mu'_{\ell}\ell_l + \mu'_s\ell_s + \mu'_b\ell_b = P'_{k(\theta)},$$

$$-\log \frac{I}{I_0}(^{241}\mathrm{Am}) = \mu''_l\ell_l + \mu''_s\ell_s + \mu''_b\ell_b = P''_{k(\theta)}, \qquad (58)$$

$$-\log \frac{I}{I_0}(^{99m}\mathrm{Tc}) = \mu'''_l\ell_l + \mu'''_s\ell_s + \mu'''_b\ell_b = P'''_{k(\theta)},$$

and where μ_l, μ_s, and μ_b refer to known attenuation coefficients for lung, soft tissue, and bone respectively; and the primes denote the coefficient appropriate to the various energies. The system of equations, Eq. 58, can be applied to each ray sum and from this the distribution of lung, soft tissue, and bone can be determined using the algorithms of Secs. 3.3 to 3.7.

4.0 NUMBER OF VIEWS REQUIRED

If a reconstructed image is to be uniformly resolved to a resolution d of a completely unsymmetrical object, the number of discrete views must be at least

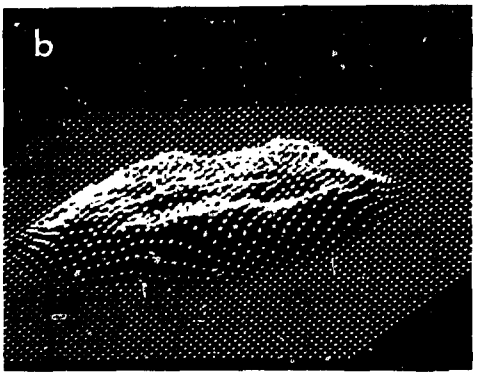
$$n \approx \pi D/d, \tag{59}$$

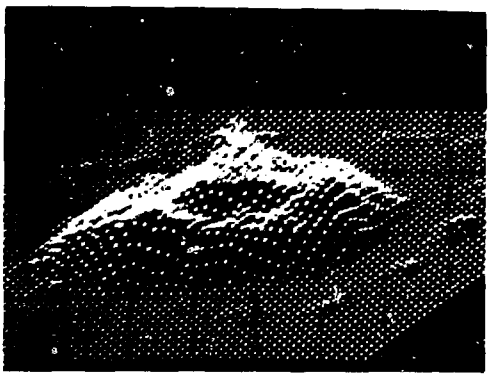
where D is the dimension of the object (Crowther et al., 1970; Klug and Crowther, 1972). Thus for a resolution of 1.5 cm in imaging a head 20 cm in diameter, we need 42 views. In practice, only 20 views are necessary for the class of objects of importance to nuclear medicine. An explanation for this discrepancy is that 42 projections would be required for an object that has no symmetry and thus no regional correlation. This is not true for any image, as there is great departure from complete randomness just by the fact that a recognizable image exists. Thus it is not surprising to find that the number of views required for reconstructing a two-dimensional distribution with a resolution distance of 1.5 cm are far fewer than theoretically prescribed for images of no symmetry. Another way of understanding the reason for the discrepancy is that in the class of objects of concern, many different objects are essentially identical. The resolution and, to a great extent, appearance of artifacts are related to how close the axis of rotation is to a center of symmetry. For example, multiple views of a right cylinder taken around an axis that is displaced from the center of rotation will give a reconstruction that is distorted and contains "clutter" outside the object region (Peters, 1973). Only a single view of the same right cylinder is necessary if the cylinder is in the assumed center of rotation for the reconstruction. However, no a priori assumption can be made regarding the topology of a cross section.

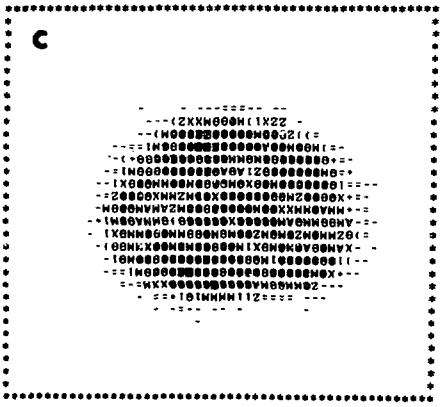
5.0 DISPLAY OF RESULTS

The distribution of intensity that represents isotope concentration can be displayed readily on the HP-5407 (Budinger and Harpootlian, 1973) using either eight levels of gray on a CRT with resolution of 64×64 or even 256×256. Concentration relationships can also be shown by the isometric or "projection" view as shown in Fig. 16.

For hard copy from the computer printer, we have worked out an over-printing routine for the CDC machines that gives levels of gray and has been very useful in our detailed program development. The subroutine that gives images, as shown in Fig. 16(c), is explained and listed in Appendix I.







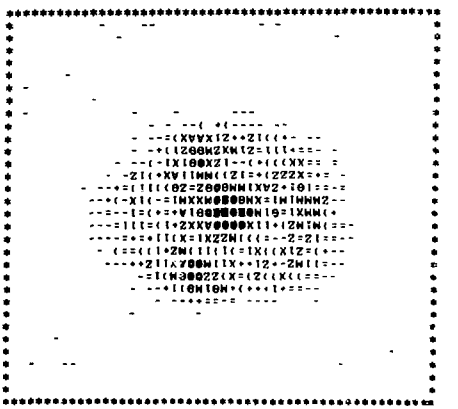


Figure 16. Techniques of displaying transverse section images after reconstruction: (a) is a display of 64 × 64 image on a CRT with eight levels of gray; (b) is the isometric or profile display of the above figures; and (c) overprinting technique for computer high-speed printer display.

6.0 SUMMARY AND FUTURE DIRECTIONS

Methods of three-dimensional isotope distribution reconstruction from multiple two-dimensional views are similar to those employed in astrophysics, electron microscopy, and radiology; however, photon attenuation must be taken into account. Successful quantitation of the three-dimensional distribution of isotopes has been achieved using the iterative least-squares technique, but not the Fourier technique. The iterative least-squares technique is superior to other techniques because it handles noise and has been successfully modified to incorporate attenuation. The method has been implemented on the CDC 6600/7600 and on the small computer HP-2100A in Fortran. In our first implementation, approximately 20 min of processing per section are required by the small machine. The ART techniques do not account for noise, but can reconstruct a section in approximately 1 min without attenuation correction. Fourier transform techniques are approximately 80 times faster than the least-squares method, but do not handle noise or attenuation.

Patient studies for isotope distribution in the head, heart, and liver can be accomplished by rotating the patient before a scintillation camera in 10° increments. The study time is approximately 30 min, and the doses are no greater than routine studies of 100 to 400 mrad.

Application of these techniques to the heart and other organs involves gating the camera or the computer to overcome motion, as has been done in preliminary studies (Fig. 17). Generalized techniques of motion extraction (Schmidlin et al., 1973; Budinger and Harpootlian, 1973) are also applicable to this tractable problem. In order to implement these techniques to heart work, a parallel-holed collimator that can handle the high energies of ^{81}Rb (446-511 keV) should be used, and one of the suggested techniques for attenuation correction should be employed.

Transmission scanning for the direct determination of attenuation co-efficients such as is done on the EMI scanner (Fischgold, 1973; Cho et al., 1973; Robb, et al., 1973) cannot be done using the gamma camera in counting mode because about 10^7 counts per picture element or ray sum are needed to determine a change in tissue density of 0.5% to 1%. The camera cannot operate over about 7×10^3 counts/sec overall or 20 counts/sec/pixel. However, it is possible to combine transmission imagery with emission to give a comparison of density distribution to isotope concentration, if the requirements are not greater than distinguishing between bone, muscle, fat, water, air, (lung = 0.2-0.6 sq. gr.).

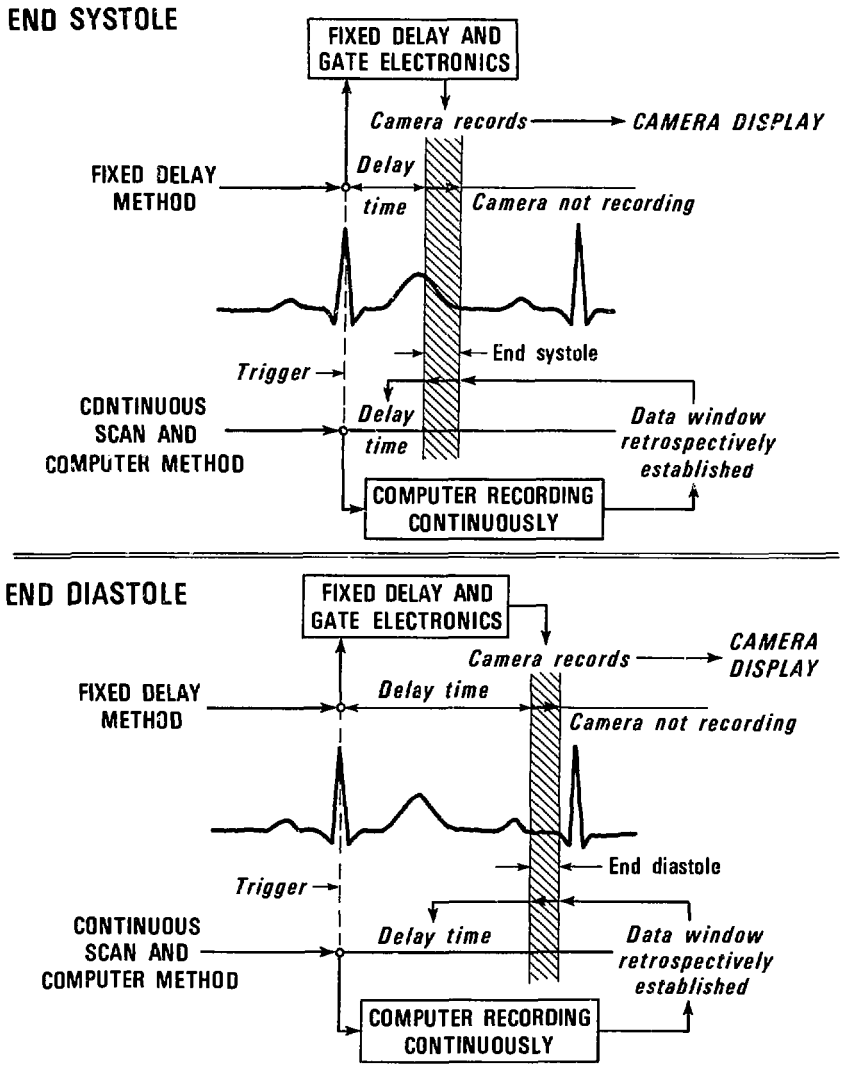


Figure 17. Regions of the EKG that are used to select data for demonstrating images of end-systole and end-diastole.

The concept of a fan beam is under development, and it is possible to reconstitute fan-beam data, either transmission or emission, to the parallel beam for implementation of the algorithm used in this report.

Proton radiography (Steward and Koehler, 1974) or radiography with heavier ions (Benton, et al., 1973) such as ^{4}He, ^{16}O, or ^{40}Ne has the potential of resolving density differences of 0.5 to 1 part in 1000; thus with these techniques the small differences in density between normal and cancerous or infected tissues can give a new dimension to clinical diagnostic medicine.

ACKNOWLEDGMENTS

This work was supported by the United States Atomic Energy Commission. We appreciate the encouragment from Drs. Hal Anger, Hans Bichsel, James Born, Robert Glaeser, John Lawrence, James McRae, and Cornelius Tobias.

Appendix A - Theorems for Fourier techniques

The back-projected image for a continuum of projections is equal to the true image convoluted with $1/r$.

$$B(r,\phi) = A(r,\phi) * 1/r.$$

Proof: The back-projected image has the following relationship (see Fig. 5):

$$B(r,\phi) = \int_0^{\pi} P(r\cos(\phi-\theta), \theta)d\theta. \tag{A1}$$

By the projection theorem we know that

$$P(x,\theta) = \int_{-\infty}^{\infty} \tilde{A}(R,\theta)\, e^{i2\pi Rx}\, dR, \tag{A2}$$

giving

$$B(r,\phi) = \int_0^{\pi}\int_{-\infty}^{\infty} \tilde{A}(R,\theta)\, \exp[2\pi irR\cos(\phi-\theta)]\, dR\, d\theta. \tag{A3}$$

Now we can rewrite Eq. (A3) as

$$B(r,\phi) = \int_0^{2\pi}\int_{-\infty}^{\infty} R^{-1}\, \tilde{A}(R,\theta)\exp[2\pi irR\cos(\phi-\theta)]\, R\, dR\, d\theta. \tag{A4}$$

Equation (A4) is the inverse Fourier transform of $R^{-1}\,\tilde{A}(R,\theta)$, which we can write as

$$\begin{aligned} B(r,\phi) &= \mathcal{F}^{-1}_{(2)}\{R^{-1}\,\tilde{A}(R,\theta)\} \\ &= \mathcal{F}^{-1}_{(2)}\{R^{-1}\} * A(r,\phi) \end{aligned} \tag{A5}$$

Now the inverse Fourier transform of R^{-1} is

$$\mathcal{F}^{-1}_{(2)}\{R^{-1}\} = \int_0^{\infty}\int_0^{2\pi} R^{-1}\exp[2\pi irR\cos(\phi-\theta)]\, Rd\theta\, dR \tag{A6}$$

$$= \int_0^{\infty}\int_0^{2\pi} \exp[2\pi irR\cos(\phi-\theta)]\, d\theta dR \tag{A7}$$

Expanding the integrand as a power series, we have

$$\mathcal{F}^{-1}_{(2)}\{R^{-1}\} = \int_0^\infty \int_0^{2\pi} \sum_{k=0}^{\infty} \frac{(2\pi i r R)^k}{k!} \cos^k(\phi-\theta)\, d\theta\, dR$$

$$= \int_0^\infty \sum_{k=0}^{\infty} \frac{(2\pi i r R)^k}{k!} \int_0^{2\pi} \cos^k(\phi-\theta)\, d\theta\, dR$$

Now

$$\int_0^{2\pi} \cos^k(\phi-\theta)\, d\theta = \int_0^{2\pi} \frac{(e^{i(\phi-\theta)} - e^{-i(\phi-\theta)})^k}{2^k} d\theta,$$

and expanding the term $(e^{i(\phi-\theta)} - e^{-i(\phi-\theta)})$ as a power series, we can write

$$\int_0^{2\pi} \cos^k(\phi-\theta)\, d\theta = \int_0^{2\pi} \sum_{n=0}^{k} \frac{k!}{2^k n!\,(k-n)!} e^{in(\phi-\theta)} e^{i(n-k)(\phi-\theta)}\, d\theta$$

$$= \sum_{n=0}^{k} \frac{k!}{2^k n!\,(k-n)!} e^{(2n-k)i\phi} \int_0^{2\pi} e^{(k-2n)i\theta}\, d\theta,$$

where $$\int_0^{2\pi} e^{(k-2n)i\theta}\, d\theta = \begin{cases} 0 & \text{if } 2n \neq k \\ 2\pi & \text{if } 2n = k \end{cases}$$

Therefore,

$$\mathcal{F}^{-1}_{(2)}\{R^{-1}\} = 2\pi \int_0^\infty \sum_{k=0}^{\infty} \frac{(2k)!}{2^{2k} k!\,(2k-k)!} \frac{(2\pi i r R)^{2k}}{(2k)!}\, dR$$

$$= 2\pi \int_0^\infty \sum_{k=0}^{\infty} \frac{(2\pi i r R)^{2k}}{2^{2k}(k!)^2}\, dR$$

$$= 2\pi \int_0^\infty \sum_{k=0}^{\infty} (-1)^k \frac{(2\pi r R)^{2k}}{2^{2k}(k!)^2}\, dR$$

$$= 2\pi \int_0^\infty J_0(2\pi r R)\, dR \qquad \text{(A8)}$$

Next let's investigate Eq. (A8). If we use the identity (Watson, 1966)

$$J_0(t) = \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} e^{it \sin\theta} d\theta, \tag{A9}$$

we can express $J_0(2\pi rR)$ as

$$J_0(2\pi rR) = \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} e^{i2\pi rR \sin\theta} d\theta . \tag{A10}$$

Let $w = r \sin\theta$ and $dw = r \cos\theta \, d\theta = \sqrt{r^2 - w^2}\, d\theta$, then

$$J_0(2\pi rR) = \frac{1}{\pi} \int_{-r}^{r} \frac{e^{i2\pi Rw}}{\sqrt{r^2 - w^2}} dw . \tag{A11}$$

Equation (A10) implies that

$$\mathcal{F}\{J_0(2\pi rR)\} = f(w)$$

where

$$f(w) = \begin{cases} \dfrac{1}{\pi\sqrt{r^2 - w^2}} & \text{if} \quad |w| \leq r \\ 0 & \text{elsewhere} \end{cases} .$$

Now we know that $2 \int_0^{\infty} J_0(2\pi rR)\, dR = \mathcal{F}\{J_0(2\pi rR)\}_{w=0} = f(0)$,

which implies that

$$2\pi \int_0^{\infty} J_0(2\pi rR)\, dR = \frac{2\pi}{2\pi} \cdot \frac{1}{r} = \frac{1}{r}$$

Therefore, we can express Eq. (A5) as

$$B(r,\phi) = \frac{1}{r} * A(r,\phi) \qquad \text{Q.E.D.}$$

The proof of the projection theorem for two-dimensional space is given in Peters (1973) and is repeated here using our notation.

Projection Theorem for R^2: The Fourier transform of the projection gives the components along the section [Fig. A-1(a)] in Fourier space normal to the projection, i.e.,

$$P(x', \theta) = \int_{-\infty}^{+\infty} \tilde{A}(R, \theta) \exp(2\pi i x' R)\, dr$$

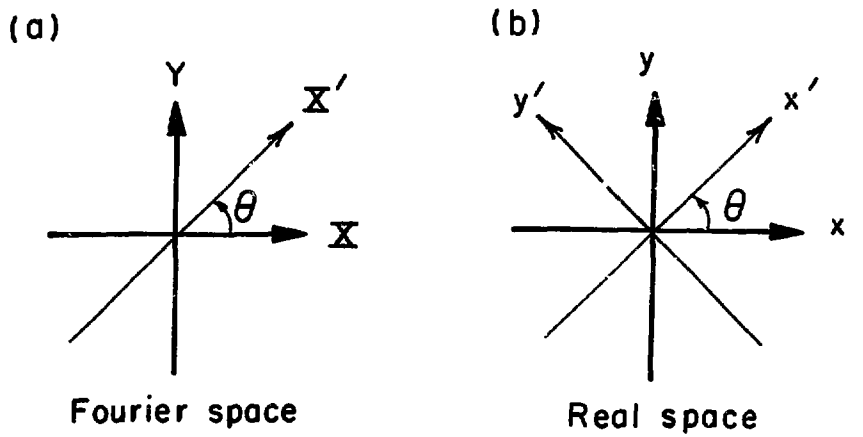


Figure A-1. Relation between components in Fourier space and projections in real space.

Proof: First, note in Fig. A-1(b) that the projections $P(x', \theta)$ can be expressed as

$$P(x', \theta) = \int_{-\infty}^{\infty} A(x, y)\, dy'$$

where the coordinate system (x', y') is rotated at an angle θ. Now expressing $A(x, y)$ in terms of its Fourier transform, we have

$$P(x', \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{A}(X, Y)\, e^{2\pi i (Xx + Yy)}\, dX\, dY\, dy' \quad .$$

Using the equations for rotation given in Eq. (2) of the text, we have

$$P(x', \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{A}(X,Y) \exp\left\{2\pi i \left[X(x'\cos\theta - y'\sin\theta) + Y(x'\sin\theta + y'\cos\theta)\right]\right\} dX\, dY\, dy'$$

Rearranging, we have

$$P(x', \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{A}(X,Y) \exp\left\{2\pi i\left[(X\cos\theta + Y\sin\theta)x' + (-X\sin\theta + Y\cos\theta)y'\right]\right\} dX\, dY\, dy'$$

$$P(x', \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{A}(X,Y) \exp\left[2\pi i x' (X\cos\theta + Y\sin\theta)\right] \exp\left[2\pi i y' (X\sin\theta - Y\cos\theta)\right] dy' dX\, dY \qquad \text{(A-12)}$$

Then integrating with respect to y', we have

$$P(x', \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{A}(X,Y) \exp\left[2\pi i x' (X\cos\theta + Y\sin\theta)\right] \delta(X\sin\theta - Y\cos\theta)\, dX\, dY \quad .$$

Next, let $U = Y\cos\theta$, which implies $dU = \cos\theta\, dY$ or $dY = \sec\theta\, dU$. Substituting these relations in Eq. (A12), we have

$$P(x', \theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \tilde{A}(X, U\sec\theta) \exp\left[2\pi i x' (X\cos\theta + U\tan\theta)\right] \delta(X\sin\theta - U) \sec\theta\, dU\, dX$$

$$= \int_{-\infty}^{\infty} \tilde{A}(X, X\tan\theta) \exp\left[2\pi i x' (X\cos\theta + X\sin\theta\tan\theta)\right] \sec\theta\, dX$$

$$= \int_{-\infty}^{\infty} \tilde{A}(X, X\tan\theta) \exp(2\pi i x' X\sec\theta) \sec\theta\, dX$$

If we let $R = X\sec\theta$, then $dR = \sec\theta\, dX$ giving us

$$P(x', \theta) = \int_{-\infty}^{\infty} \tilde{A}(R\cos\theta, R\sin\theta) \exp(2\pi i x' R)\, dR$$

or

$$P(x', \theta) = \int_{-\infty}^{\infty} \tilde{A}(R,\theta) \exp(2\pi i x' R)\, dR \quad . \qquad \text{Q. E. D.}$$

Appendix B. The Relationship Between the Array Pixels and the Projection Rays

Figure B-1 illustrates the relationship between the N × N array of pixels and the family of lines which bound the projection rays. Note that the projection rays remain fixed corresponding to a fixed camera, and the coordinate system rotates relative to these fixed rays. The development of the equations relating the array pixels and projection rays is separated into four categories corresponding to the rotated angle θ. Subsection V generalizes these equations for all angles.

I. $\theta = 0°$

For $\theta = 0°$ the family of lines which define the projection rays is

$$y = k + 1/2 \qquad k = 0, 1, 2, \cdots, N. \tag{B1}$$

II. $0° < \theta < 90°$

A. The family of lines

For $0 < \theta < 90°$ the family of lines which define the projection rays are

$$y = (\tan\theta)x + y_0 + k/\cos\theta, \quad k = 0, 1, 2, \cdots, n_\theta, \tag{B2}$$

where $\tan\theta$ is the slope of the family of lines, y_0 is the y intercept for the line $k = 0$, $1/\cos\theta$ is the increase in the y intercept for each succeeding line and n_θ is the total number of rays necessary to cover the N×N array. This varies from N to $\sqrt{2}$ N if the width of each ray is identical to the distance between pixels.

Before developing the equation for y_0 and n_θ, first compare Fig. B-1(a) for $\theta = 0°$ to Fig. B-1(b) for $0° < \theta < 90°$ and notice that additional lines are added so that the N×N array is bound above and below by lines. The necessity for adding additional lines is that the distance between the points (N, 1) and (1, N) and the line x_c has increased as the coordinate system is rotated. The distance between x_c and (N, 1) is developed in the following sequence of equations:

$$\text{Distance between } x_c \text{ and } (N,1) = \frac{N-1}{\sqrt{2}} \sin(\theta+45°)$$

$$= \frac{N-1}{\sqrt{2}} (\sin\theta\cos 45° + \cos\theta \sin 45°)$$

$$= \frac{N-1}{2} (\sin\theta + \cos\theta). \tag{B3}$$

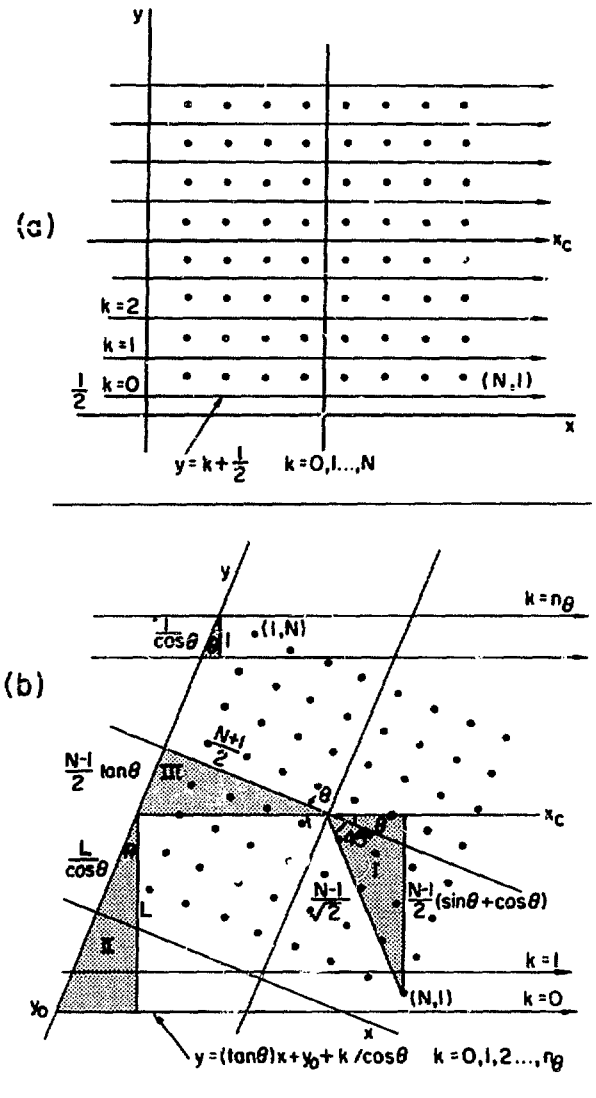


Figure B-1. Relationship between the picture elements and coordinate system for the derivation of the family of lines, which delineate the parallel rays.

Due to symmetry, Eq. (B3) holds for the point (1, N) as well as (N, 1). When $\theta = 45°$ this distance will be at a maximum.

From Eq. (B3) we can develop an expression for the distance between the point (N, 1) at $\theta = 0°$ and the point (N, 1) at $0° < \theta < 90°$.

$$\underline{\text{Distance between}}\ (N,1)\big|_{\theta = 0°}\ \underline{\text{and}}\ (N,1)\big|_{0° < \theta < 90°}$$

$$= (\underline{\text{Distance between}}\ x_c\ \underline{\text{and}}\ (N,1)\big|_{0° < \theta < 90°}) - (\underline{\text{Distance between}}\ x_c\ \underline{\text{and}}\ (N,1)\big|_{\theta = 0°})$$

$$= \frac{N-1}{2}(\sin\theta + \cos\theta) - (\frac{N-1}{2}) = \frac{N-1}{2}(\sin\theta + \cos\theta - 1). \qquad \text{(B4)}$$

Now at $\theta = 0°$ the first line corresponding to $k = 0$ is at $y = 1/2$ and the distance between it and the point (N, 1) is 1/2. Therefore the distance expressed in Eq. (B4) can increase by 1/2 before more lines are necessary. To determine the distance between the line $k = 0$ at $\theta = 0°$ and the line $k = 0$ at $0° < \theta < 90°$, one needs to add 1/2 to Eq. (B4) and take the largest integer less than this since the family of lines are separated by integral widths. Therefore,

$$\underline{\text{Distance between line}}\ k = 0\big|_{0° < \theta < 90°}\ \underline{\text{and line}}\ k = 0\big|_{\theta = 0°}$$

$$= \text{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right]. \qquad \text{(B5)}$$

For the special case when $[\ \bullet\ ] = \text{INT}[\ \bullet\ ]$ in Eq. (B5), we adopt the rule that if a point lies on one of the family of lines, let's say the line corresponding to k, then the point (pixel) belongs to the ray k + 1. Therefore when $[\ \bullet\ ] = \text{INT}[\ \bullet\ ]$ a ray will be added for the point (1, N) but not for the point (N, 1). Therefore, from Eq. (B5), we see that the total number of additional rays necessary to cover an array for $0° < \theta < 90°$ is given in the following equation:

$$\underline{\text{Additional rays}} = \begin{cases} 2\ \text{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right] & \text{if } [\ \bullet\ ] > \text{INT}[\ \bullet\ ] \\ 2\ \text{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right] - 1 & \text{if } [\ \bullet\ ] = \text{INT}[\ \bullet\ ] \end{cases} \qquad \text{(B6)}$$

Adding Eq. (B6) to N which is the number of rays for $\theta = 0°$ we get the total number of rays n_θ for the angle $0° < \theta < 90°$:

$$n_\theta = \begin{cases} N + 2\,\mathrm{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right] & \text{if } [\,\bullet\,] > \mathrm{INT}[\,\bullet\,] \\ N + 2\,\mathrm{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right] - 1 & \text{if } [\,\bullet\,] = \mathrm{INT}[\,\bullet\,] \end{cases} \tag{B7}$$

For $\theta = 0°$ the distance from x_c to the line k = 0 is N/2. Therefore, the distance to the line k = 0 for $0° < \theta < 90°$ is N/2 plus the increased integral ray widths such that the point (N, 1) is bounded below by a line. Using Eq. (B5) with provision for the placement of a point if it lies on a line, we have

Distance between line k = 0 $|_{0° < \theta < 90°}$ and x_c equals

$$L = \begin{cases} \frac{N}{2} + \mathrm{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right] & \text{if } [\,\bullet\,] \quad \mathrm{INT}[\,\bullet\,] \\ \frac{N}{2} + \mathrm{INT}\left[\frac{N-1}{2}(\sin\theta + \cos\theta - 1) + 1/2\right] - 1 & \text{if } [\,\bullet\,] = \mathrm{INT}[\,\bullet\,] \end{cases} \tag{B8}$$

Now with Eq. (B8) we can use triangles II and III (Fig. B-1(b)) to develop an expression for the y intercept, y_0, for the line k = 0. The line segments on the y axis corresponding to triangles II and III have values $L/\cos\theta$ and $\frac{N+1}{2}\tan\theta$, respectively. Thus the y intercept, y_0, for the line k = 0, has the equation

$$y_0 = \frac{N+1}{2} - \frac{L}{\cos\theta} - \frac{N+1}{2}\tan\theta\,. \tag{B9}$$

B. The set of rays $\{k(\theta)\}$ corresponding to a given pixel (I, J).

For each pixel in the N×N array with coordinates (I, J), an expression can be developed which immediately determines the ray for which it is an element. First, notice in the following figure that we already have an expression for the distance between the point (N, 1) and the line x_c and the distance between the line k = 0 and the line x_c as given in Eq. (B3) and (B8) respectively.

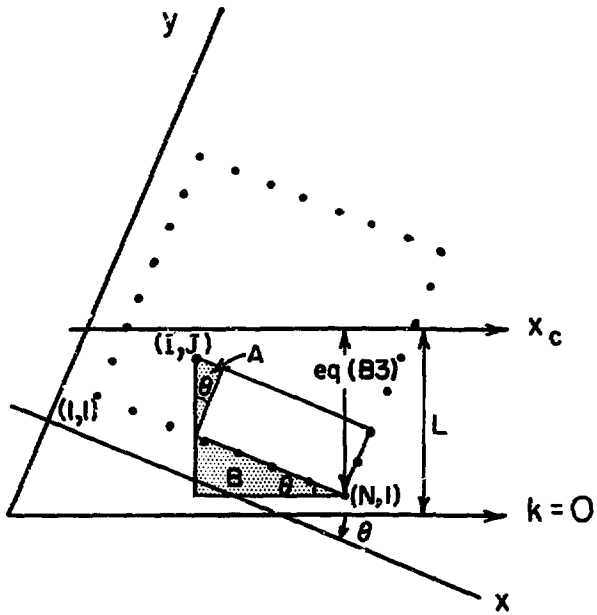


Figure B-2. Distance between the first line, of the family of lines delineating the rays, and a line through the center of the array for projection angles between 0° and 90°.

For the point corresponding to the coordinates (I, J), the leg adjacent to θ for triangle A will have the value J-1, which implies that the hypotenuse has the value $(J-1)/\cos\theta$ and the leg opposite θ has the value $(J-1)\tan\theta$. Therefore, the hypotenuse of triangle B has the value $N-I-(J-1)\tan\theta$, which implies that the leg opposite θ has the value $[N-I-(J-1)\tan\theta]\sin\theta$. Hence,

Distance between (I, J) and x_c

$$= \left| \frac{N-1}{2}(\sin\theta + \cos\theta) - (J-1)/\cos\theta + [(N-I) - (J-1)\tan\theta]\sin\theta \right|.$$

Combining terms we have

$$\left| \frac{N+1-2I}{2}\sin\theta + \frac{2J-N-1}{2}\cos\theta \right|. \qquad \text{(B10)}$$

Using Eq. (B10) the distance D between (I, J) and the line k = 0 is

$$D = L + \frac{N+1-2I}{2}\sin\theta + \frac{2J-N-1}{2}\cos\theta. \qquad \text{(B11)}$$

From Eq. (B11) we can calculate the particular ray for a projection θ that goes through the point (I, J) as

$$k(\theta) = \text{INT}\{D\} + 1. \tag{B12}$$

III. $\theta = 90°$

For $\theta = 90°$ the family of lines that define the projection rays is

$$x = k + 1/2 \qquad k = 0, 1, 2, \ldots, N. \tag{B13}$$

IV. $90° < \theta < 180°$

A. The family of lines

For $90° < \theta < 180°$, the family of lines which defines the projection rays is

$$y = (\tan\theta)x + y'_0 + k/|\cos\theta| \qquad k = 0, 1, 2, \cdots n_\theta, \tag{B14}$$

where $\tan\theta$ is the slope of the family of lines, y'_0 is the y intercept for the line $k = 0$, $1/|\cos\theta|$ is the increase in the y intercept for each succeeding line, and n_θ is the total number of rays.

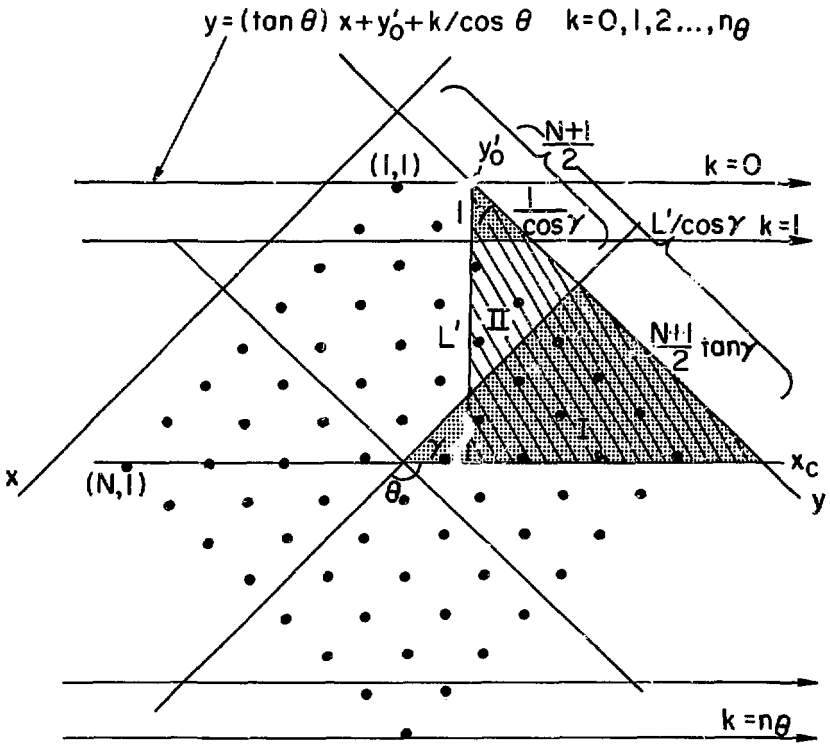
The development of the expression for y'_0 and n_θ is similar for that given for $0° < \theta < 90°$. Notice in Fig. B-3 that the family of lines increases down the plane, whereas for $0° < \theta < 90°$ the family of lines increases up the plane. Also another thing must be kept in mind; as illustrated in Fig. B-3, $\gamma = 180° - \theta$, therefore $\cos\gamma = |\cos\theta|$.

Due to symmetry, the distance, L', between x_c and the line $k = 0$ for $90° < \theta < 180°$ will have the same expression as given for L in Eq. (B),

$$L' = \begin{cases} \frac{N}{2} + \text{INT}[\frac{N-1}{2}(\sin\theta + |\cos\theta| - 1) + 1/2] & \text{if } [\,\cdot\,] > \text{INT}[\,\cdot\,] \\ \frac{N}{2} + \text{INT}[\frac{N-1}{2}(\sin\theta + |\cos\theta| - 1) + 1/2] - 1 & \text{if } [\,\cdot\,] = \text{INT}[\,\cdot\,] \end{cases} \tag{B15}$$

and the number of rays will also have a similar expression,

$$n_\theta = \begin{cases} N + 2\,\text{INT}[\frac{N-1}{2}(\sin\theta + |\cos\theta| - 1) + 1/2] & \text{if } [\,\cdot\,] > \text{INT}[\,\cdot\,] \\ N + 2\,\text{INT}[\frac{N-1}{2}(\sin\theta + |\cos\theta| - 1) + 1/2] - 1 & \text{if } [\,\cdot\,] = \text{INT}[\,\cdot\,] \end{cases} \tag{B16}$$



XBL741-2178

Figure B-3. Relationship between the picture elements and the family of lines for projection angle between 90 and 180°.

Now using triangles I and II indicated in Fig. B-3, we an develop an expression for y'_0. For triangle I the side adjacent to γ has the value $(N+1)/2$, which implies that the side opposite γ is $[(N+1)/2)]\tan\gamma$. Also, knowing the value for L', we have $L'/\cos\gamma$ as the value for the hypotenuse of triangle III. Therefore, utilizing these line segments on the y axis, we have

$$y'_0 = \frac{N+1}{2} - \frac{L'}{\cos\gamma} + \frac{N+1}{2}\tan\gamma$$

$$= \frac{N+1}{2} - \frac{L'}{|\cos\theta|} - \frac{N+1}{2}\tan\theta. \qquad \text{(B17)}$$

B. The set of rays $\{k(\theta)\}$ corresponding to a given pixel (I, J).

As was done for $0° < \theta < 90°$ in Section IIB, we can develop an expression that gives the ray for a particular projection that goes through a pixel (I, J). The distance between the point (1, 1) and the line x_c and the distance between the line k = 0 and the line x_c is given in Eq. (B3) and (B15) respectively.

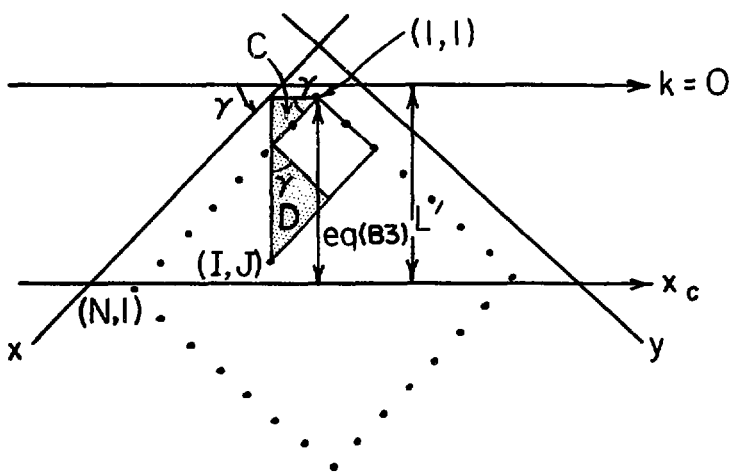


Figure B-4. Distance between the first line, of the family of lines delineating the rays, and a line through the center of the array for projection angles between 90° and 180°.

In Fig. B-4 the side adjacent to triangle D has the value J-1 which implies that the hypotenuse has the value $(J-1)/\cos\gamma$ and the side opposite γ has the value $(J-1)\tan\gamma$. Now the hypotenuse of triangle C is $I-1-(J-1)\tan\gamma$; therefore the side opposite γ has the value $[I-1-(J-1)\tan\gamma]\sin\gamma$. This gives

Distance between (I, J) and x_c

$$= \left|\left[I-1-(J-1)\tan\gamma\right]\sin\gamma + (J-1)/\cos\gamma - \frac{N-1}{2}(\sin\gamma + \cos\gamma)\right|$$

$$= \left|\frac{(2I-N-1)}{2}\sin\gamma + \frac{(2J-N-1)}{2}\cos\gamma\right|, \tag{B18}$$

which implies that the distance D' between (I, J) and line k = 0 is

$$D' = L' + \frac{(2I-N-1)}{2}\sin\theta + \frac{(2J-N-1)}{2}\left|\cos\theta\right| \tag{B19}$$

From Eq. (B19) we can determine the ray corresponding to the pixel (I, J) for a given angle θ,

$$k(\theta) = \mathrm{INT}\{D'\} + 1. \tag{B20}$$

V. $\theta \geq 180°$

Figure B-5 illustrates the corresponding relationships for angles greater than 180°. Therefore, from the equations given in sections II and IV we can summarize the equations for the family of lines determining rays for all angles other than projectives for integral multiples of $\pi/2$,

$$y = (\tan\theta)x + y_0 + k/\left|\cos\theta\right|, \qquad k = 0, 1, 2, \cdots, n_\theta \tag{B21}$$

where

$$y_0 = \frac{N+1}{2} - \frac{L}{\left|\cos\theta\right|} - \frac{N+1}{2}\tan\theta,$$

and

$$L = \begin{cases} \frac{N}{2} + \mathrm{INT}\left[\frac{N-1}{2}\left(\left|\sin\theta\right| + \left|\cos\theta\right| - 1\right) + 1/2\right] & \text{if } [\,\cdot\,] > \mathrm{INT}[\,\cdot\,] \\ \frac{N}{2} + \mathrm{INT}\left[\frac{N-1}{2}\left(\left|\sin\theta\right| + \left|\cos\theta\right| - 1\right) + 1/2\right] - 1 & \text{if } [\,\cdot\,] = \mathrm{INT}[\,\cdot\,] \end{cases}$$

$$n_\theta = \begin{cases} N + 2\mathrm{INT}\left[\frac{N-1}{2}\left(\left|\sin\theta\right| + \left|\cos\theta\right| - 1\right) + 1/2\right] & \text{if } [\,\cdot\,] > \mathrm{INT}[\,\cdot\,] \\ N + 2\mathrm{INT}\left[\frac{N-1}{2}\left(\left|\sin\theta\right| + \left|\cos\theta\right| - 1\right) + 1/2\right] - 1 & \text{if } [\,\cdot\,] = \mathrm{INT}[\,\cdot\,]. \end{cases}$$

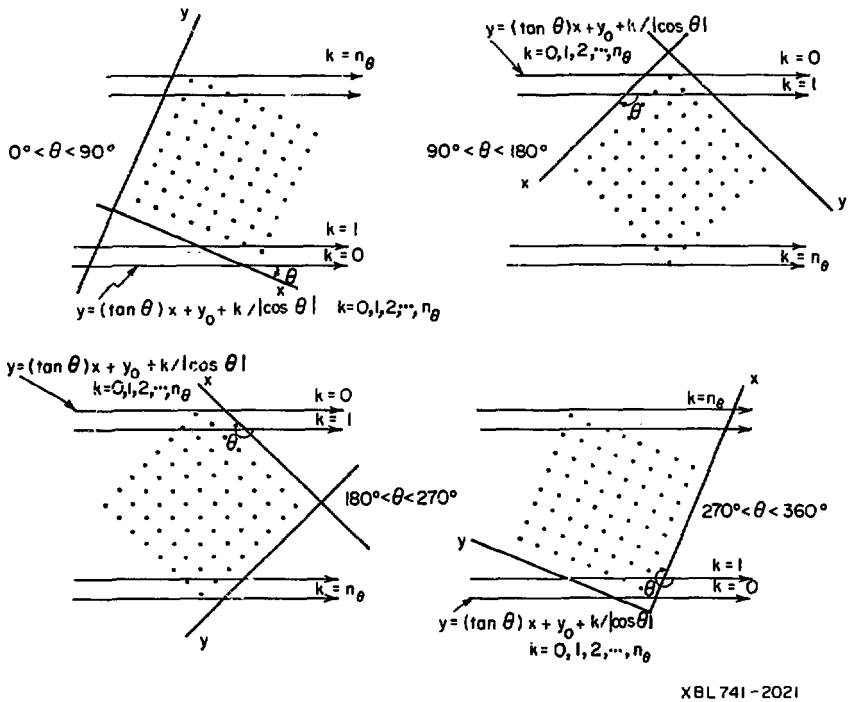


Figure B-5. Relationship between the equations for the family of lines delineating rays.

Whereas the angles $\theta = 0°$, $180°$ the family of lines is

$$y = k + 1/2 \qquad k = 0, 1, 2, \cdots, N, \tag{B22}$$

and for angles $\theta = 90°$, $270°$ the family of lines is

$$x = k + 1/2 \qquad k = 0, 1, 2, \cdots, N . \tag{B23}$$

Equations (B11) and (B12) give the ray corresponding to a pixel (I, J) for $0° < \theta < 90°$ and $180° < \theta < 270°$ whereas Eqs. (B19) and (B20) give the ray corresponding to a pixel (I, J) for $90° < \theta < 180°$ and $270° < \theta < 360°$. This can be summarized as

$$k(\theta) = \text{INT}\ \{D\},$$

where

$$D = \begin{cases} L + \frac{(N+1-2I)}{2} \left|\sin\theta\right| + \frac{(2J-N-1)}{2} \left|\cos\theta\right| & \text{if } 0° < \theta < 90° \\ & 180° < \theta < 270° \\ L + \frac{(2I-N-1)}{2} \left|\sin\theta\right| + \frac{(2J-N-1)}{2} \left|\cos\theta\right| & \text{if } 90° < \theta < 180° , \\ & 270° < \theta < 360° \end{cases} \tag{B24}$$

A Subroutine Determining the Projection Values for Each Ray

Equations (B21) - (B24) give the family of lines which defines the rays for each angle. Using these equations the following subroutine generates the projection values $R^n_{k\theta}$ by

CALL SUM(B, XR)

where B is a N×N array. The array XR contains the projections of B, which is returned upon execution of the above subroutine. These projections are denoted by XR(M, KK) where M is the index for the angle and KK is the index for the ray. It is assumed that the projection $R_{k\theta}$ has the functional relationship $R_{k\theta} = \sum_{(i,j)\in rayk(\theta)} A(i,j)$. See Appendix H for the case where $R_{k\theta} = \sum_{(i,j)\in rayk(\theta)} f^{\theta}_{ij} A(i,j)$ and f^{θ}_{ij} is some factor other than 1.

Note that the listing for SUM calls the following two subroutines:

CALL YMIN(KK, M, IY1, IY2)

CALL XMIN(KK, M, JY, IX1, IX2)

where KK is the index for the ray and M is the index for the angle. The returned values IY1 and IY2 are respectively the minimum and maximum y coordinates for pixels in the ray KK at the angle M. The returned values IX1 and IX2 are respectively the minimum and maximum x coordinates for pixels in ray KK at the angle M and having a y coordinate of JY. An example of the variables IY1, IY2, IX1 and IX2 for an 8 × 8 array are shown in Fig. B-6.

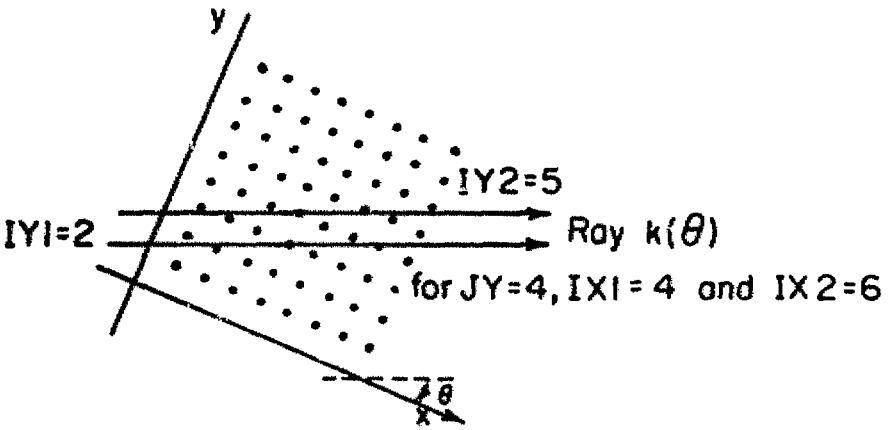


Figure B-6. Parameters calculated by a subroutine CALL XMIN.

The above subroutines require a COMMON block which contains the cosine, sine, and tangent of each projection angle and other constants which remain fixed for each iteration. These trigonometric functions can be evaluated and stored in the block COMMON by

CALL CONST

where the block COMMON has the following values

S(I) = ABS[SIN(θ_I)]

C(I) = ABS[COS(θ_I)]

T(I) = TAN(θ_I)

IR(I)= The number of rays for the angle θ_I. See Eq. (B21).

XL(I)=The variable L given in Eq. (B21).

IA(I)= The angle θ_I in degrees

XN = FLOAT (N)

XN1 = FLOAT (N-1)

XN2 = FLOAT (N1)

N = The NXN array size being reconstructed.

N1 = N+1

M1 = The number of projections

The variable IAA in SUBROUTINE CONST is the angle in degrees in which the subject is rotated for each succeeding projection.

```
      SUBROUTINE SUM(B,XR)
      DIMENSION B(46,46),XR(36,64)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      DO 18 M=1,M1
      MR=IR(M)
      IF(IA(M).EQ.0.OR.IA(M).EQ.180)52,51
   51 IF(IA(M).EQ.90.OR.IA(M).EQ.270)54,53
   52 DO 19 K=1,MR
      XR(M,K)=0.
      DO 66 I=1,N
      XR(M,K)=XR(M,K)+B(I,K)
   66 CONTINUE
   19 CONTINUE
      GO TO 18
   54 DO 61 K=1,MR
      XR(M,K)=0.
      DO 67 J=1,N
      XR(M,K)=XR(M,K)+B(K,J)
   67 CONTINUE
   61 CONTINUE
      GO TO 18
   53 DO 63 K=1,MR
      XR(M,K)=0.
      CALL YMIN(K,M,IY1,IY2)
      DO 68 J=IY1,IY2
      CALL XMIN(K,M,J,IX1,IX2)
      DO 68 I=IX1,IX2
      XR(M,K)=XR(M,K)+B(I,J)
   68 CONTINUE
   63 CONTINUE
   18 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE YMIN(KK,M,IY1,IY2)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      XK=FLOAT(KK)
      XK1=FLOAT(KK-1)
      DO 7 I=1,N
      XI=FLOAT(I)
      X=(XI-XN2*.5*(1.-T(M))+(XL(M)-XK1)/C(M))/T(M)
      IF(IA(M).LE.90.OR.IA(M).GT.180.AND.IA(M).LT.270)8,9
    8 IF(X-1.)7,10,10
    9 IF(X-XN)10,10,7
    7 CONTINUE
   10 IY1=I
      IF(IR(M)-KK)11,12,11
   12 IY2=N
      GO TO 13
   11 MI=I+1
      DO 14 J=MI,N1
      XJ=FLOAT(J)
      X=(XJ-XN2*.5*(1.-T(M))+(XL(M)-XK)/C(M))/T(M)
      IF(IA(M).LE.90.OR.IA(M).GT.180.AND.IA(M).LT.270)15,16
   15 IF(X-XN)14,17,17
   16 IF(X-1.)17,17,14
   14 CONTINUE
   17 IY2=J-1
   13 CONTINUE
      RETURN
      END

      SUBROUTINE XMIN(KK,M,JY,IX1,IX2)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      XK=FLOAT(KK)
      XK1=FLOAT(KK-1)
      XJ=FLOAT(JY)
      X1=(XJ-XN2*.5*(1.-T(M))+(XL(M)-XK1)/C(M))/T(M)
      X2=(XJ-XN2*.5*(1.-T(M))+(XL(M)-XK)/C(M))/T(M)
      IF(IA(M).LE.90.OR.IA(M).GT.180.AND.IA(M).LT.270)19,20
   19 IF(X1-XN)31,32,32
   31 IX2=INT(X1)
      GO TO 33
   32 IX2=N
   33 IF(X2-1.)34,35,35
   35 IX1=INT(X2)+1
      GO TO 30
   34 IX1=1
      GO TO 30
   20 IF(X1-1.)21,22,22
   22 IX1=INT(X1)+1
      GO TO 25
   21 IX1=1
   25 IF(X2-XN)26,26,27
   26 IX2=INT(X2)
      GO TO 30
   27 IX2=N
   30 RETURN
      END
```

```
      SUBROUTINE CONST
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      READ FMT1,N
      READ FMT2,M1
      READ FMT3,IAA
      N1=N+1
      XN=FLOAT(N)
      XN1=FLOAT(N-1)
      XN2=FLOAT(N1)
      DO 1 I=1,M1
      IA(I)=(I-1)*IAA
      AA=FLOAT(IA(I))*3.1415927/180.
      S(I)=ABS(SIN(AA))
      C(I)=ABS(COS(AA))
      T(I)=TAN(AA)
      ARG=.5*XN1*(S(I)+C(I)-1.)+.5
      IZ=INT(ARG)
      Z=FLOAT(IZ)
      IF(ARG-Z)21,22,21
   21 XL(I)=XN*.5+Z
      GO TO 2
   22 XL(I)=XN*.5+Z-1.
    2 IF(IA(I).EQ.0.OR.IA(I).EQ.90.OR.IA(I).EQ.180.OR.IA(I).EQ.270)6,8
    6 IR(I)=N
      GO TO 1
    8 IF(ARG-Z)11,12,11
   11 IR(I)=N+2*IZ
      GO TO 1
   12 IR(I)=N+2*IZ-1
    1 CONTINUE
      RETURN
      END
```

A Subroutine for Determining the Rays Corresponding to Each Pixel

Equation (B24) gives a relationship for determining the rays corresponding to each pixel with coordinates (I, J). This relationship is achieved on the digital computer by

CALL RAY (I, J, M, K)

where (I, J) are the coordinates for the pixel, M is the index for the angle, and K is the index for the ray which is returned after the above subroutine is executed.

```
      SUBROUTINE RAY(I,J,M,K)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      IF(IA(M).EQ.0.OR.IA(M).EQ.180)22,23
   23 IF(IA(M).EQ.90.OR.IA(M).EQ.270)25,35
   35 IF(IA(M).LT.90.OR.IA(M).GT.180.AND.IA(M).LT.270)24,26
   22 K=J
      GO TO 29
   24 XX=FLOAT(N-2*I+1)
      YY=FLOAT(2*J-N-1)
      GO TO 27
   25 K=I
      GO TO 29
   26 XX=FLOAT(2*I-N-1)
      YY=FLOAT(2*J-N-1)
   27 K=INT((XX*S(M)+YY*C(M))*.5+XL(M))+1
   29 RETURN
      END
```

Appendix C. Using Generalized Inverse for Three-Dimensional Reconstruction

Techniques of solving for the unknown values at each picture element (i, j) from both an adequate and inadequate number of views using direct matrix methods are detailed in this appendix.

The criterion for the reconstruction of an image from multiple projections $P_{k\theta}$ is that the best estimate, $[\hat{A}(i,j)]$, be a minimum to the least-squares function

$$\mathcal{R}(A) = \sum_{\theta} \sum_{k=1}^{n_\theta} \frac{(P_{k\theta} - R_{k\theta})^2}{\sigma_{k\theta}^2}, \tag{C1}$$

where the densities A(i, j) satisfy the functional relationship

$$R_{k\theta} = \sum_{(i,j)\in \text{ray}(k,\theta)} f_{ij}^{\theta} A(i,j), \quad \theta = 1, \cdots, M; \quad k = 1, \ldots, n_\theta, \tag{C2}$$

and $\sigma_{k\theta}$ is the standard deviation in the measured projection $P_{k\theta}$. The factor f_{ij}^{θ} represents the fraction of density that we assume is being measured by the ray $R_{k\theta}$. This factor can incorporate the expected results due to attenuation and spread of an emitting source as measured by a gamma camera. Note that the factors f_{ij}^{θ} are also a function of the angle θ.

Generalized Inverse

Equation (C1) can be expressed in matrix notation as

$$\mathcal{R}(A) = (P - FA)^T \Phi^{-1} (P - FA). \tag{C3}$$

The row matrix,

$$P^T = [P_{11} P_{21} \cdots\cdots P_{n_1} \cdots\cdots P_{1M} P_{2M} \cdots\cdots P_{n_M M}] \quad (1 \times m),$$

is $(P_{K\theta})$, a matrix of measured projections where K designates the particular ray and θ the projection. The row matrix A^T is

$$A^T = [A(1,1)\ A(2,1) \cdots\cdots A(N,1) \cdots\cdots A(1,N)\ A(2,N) \cdots\cdots A(N,N)] \quad (1 \times n^2),$$

where A(i, j) are the values for the elements of the section to be reconstructed. The matrix Φ^{-1} is the inverse of the covariance matrix for the errors in the measured projections P, where we assume that the measured projections are independent:

$$\Phi^{-1} = \begin{bmatrix} 1/\sigma_{11}^2 & & & & \\ & 1/\sigma_{21} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & 1/\sigma_{n_M M} \end{bmatrix} .$$

The matrix F is an $m \times N^2$ matrix where the row (k, θ) is composed of 0's and f_{ij}^{θ}, depending on whether the point (i, j) is an element of ray (k, θ). The variable m is the sum total of rays for all projections such that

$$m = \sum_{\theta=1}^{M} n_\theta .$$

An example of the matrix F is illustrated by the following 4×4 array with projections for 0° and 90°.

				$\Sigma =$
				P_{42}
				P_{32}
A(1,2)	A(2,2)	• • • •		P_{22}
A(1,1)	A(2,1)	A(3,1)	A(4,1)	P_{12}

$\Sigma =$ P_{11} P_{21} P_{31} P_{41}

If we assume that $f_{ij}^{\theta} = 1$ for all i, j, θ then for this particular example the matrix F is

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad \text{(C4)}$$

The best estimate $\hat{A}$ which minimizes Eq. (C3) must satisfy the equation

$$\nabla_A \mathcal{R} \Big|_{\hat{A}} = 0, \quad \text{(C5)}$$

where ∇_A is a matrix differential operator (See Deutsch, 1965). By expanding (C3) we can write (C5) as

$$\nabla_A (P^T \Phi^{-1} P - P^T \Phi^{-1} FA - A^T F^T \Phi^{-1} A + A^T F^T \Phi^{-1} FA) \Big|_{\hat{A}} = 0$$

and simplifying we have

$$-2F^T \Phi^{-1} P + 2F^T \Phi^{-1} F\hat{A} = 0.$$

Solving for $\hat{A}$ we obtain

$$\hat{A} = (F^T \Phi^{-1} F)^{-1} F^T \Phi^{-1} P , \quad \text{(C6)}$$

if the inverse matrix $(F^T \Phi^{-1} F)^{-1}$ exists. Goitein (1971) claims this is true if $(m-N^2-M+1) > 0$. However, if the matrix $F^T \Phi^{-1} F$ is singular, we can express $\hat{A}$ in terms of the generalized inverse, F^G, of the matrix F,

$$\hat{A} = [(\Phi^{-1})^{1/2} F]^G (\Phi^{-1})^{1/2} P . \quad \text{(C7)}$$

If we assume that $\Phi^{-1} = I$ and F is the matrix given in the previous example, (C4), then $\hat{A}$ can be solved by multiplying the generalized inverse F^G by the measured values P:

$$\hat{A}=\begin{bmatrix}
0.21875 & -0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 \\
-0.03125 & 0.21875 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 \\
-0.03125 & -0.03125 & 0.21875 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 \\
-0.03125 & -0.03125 & -0.03125 & 0.21875 & 0.21875 & -0.03125 & -0.03125 & -0.03125 \\
0.21875 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 \\
-0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 \\
-0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 \\
-0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & 0.21875 & -0.03125 & -0.03125 \\
0.21875 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 \\
-0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 \\
-0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 \\
-0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & 0.21875 & -0.03125 \\
0.21875 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & 0.21875 \\
-0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & 0.21875 \\
-0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 & -0.03125 & 0.21875 \\
-0.03125 & -0.03125 & -0.03125 & 0.21875 & -0.03125 & -0.03125 & -0.03125 & 0.21875
\end{bmatrix} \times P$$

This is the unweighted best estimate for the densities A(i, j). The appealing thing about the generalized inverse is that once the generalized inverse has been determined, then the estimate $\hat{A}$ can be determined by direct matrix multiplication. Also the computer storage space necessary for the matrix F^G can be reduced by noticing that the factor multiplying each projection $P_{k\theta}$ is only a function of the number of elements in the ray (K, θ). However, usually the generalized inverse for a large array is not easy to determine.

Appendix D. Subroutine for Back-Projection Algorithm

The reconstructed back-projection image can be obtained by

CALL BCKPROJ (B, P)

where B is the reconstructed array, and the array P contains the sampled projection data. Each projection value is denoted by P(M, K) where M is the index of the angle and K is the index of the ray. The ray corresponding to a particular back-projection pixel is found by the subroutine

CALL RAY(I, J, M, K),

which is listed in Appendix B.

The flow chart for the back-projection algorithm is presented in Fig. D-1.

```
      SUBROUTINE BCKPROJ(B,P)
      DIMENSION B(46,46),P(36,64)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      XT=0.
      MR=IR(1)
      DO 1 I=1,MR
      XT=XT+P(1,I)
    1 CONTINUE
      DO 20 I=1,N
      DO 20 J=1,N
      XP2=0.
      DO 30 M=1,M1
      CALL RAY(I,J,M,K)<----------cf Appendix B
      XP2=XP2+P(M,K)
   30 CONTINUE
      B(I,J)=XP2
   20 CONTINUE
      XTT=0.
      DO 80 I=1,N
      DO 80 J=1,N
      XTT=XTT+B(I,J)
   80 CONTINUE
      DO 83 I=1,N
      DO 83 J=1,N
      B(I,J)=XT*B(I,J)/XTT
   83 CONTINUE
      RETURN
      END
```

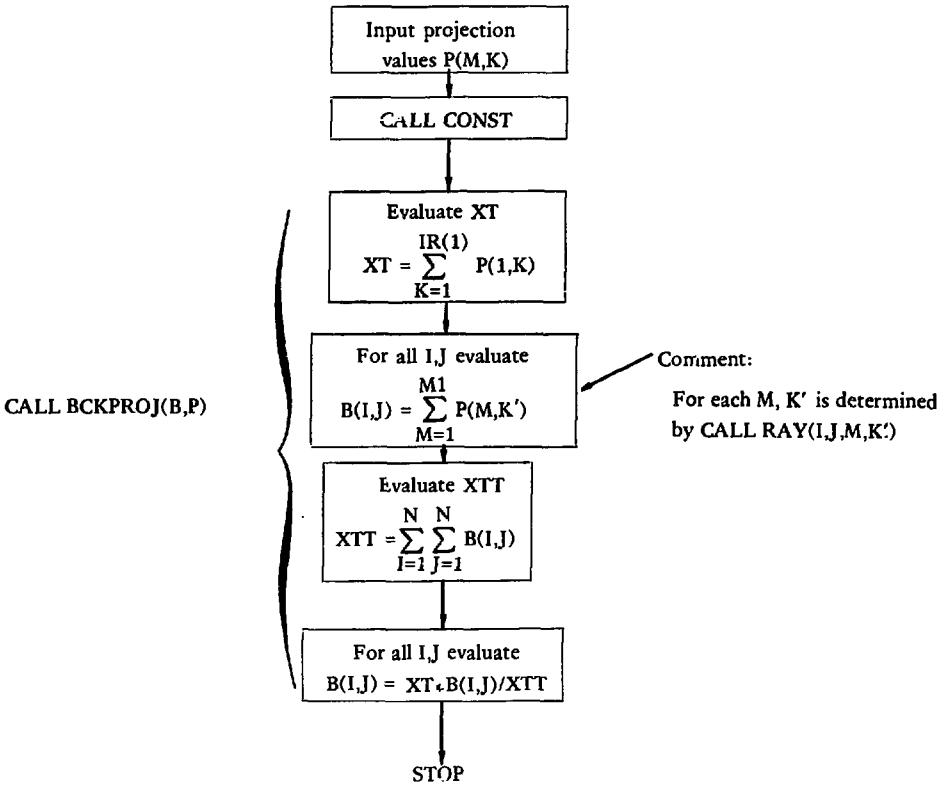


Figure D-1. Flow chart for back-projection algorithim.

Appendix E. Subroutine for SIRT Algorithm

The simultaneous iterative reconstruction technique (SIRT) is discussed in Section (3.5). The flow chart is presented in Fig. E-1. The image B is reconstructed by the SIRT algorithm by

CALL SIRT (B, P, ITER)

where B is the reconstructed array, P is the sampled projection data, and ITER is the number of iterations desired. Each projection value is denoted by P(M, K) where M is the index for the angle and K is the index for the ray. The number of pixels for each ray is stored in NN(M, K) and the length of each ray is stored in XLENGTH(M, K). Ray lengths are evaluated by

CALL XLENGH(M, K, X)

where M is the index for the angle, K is the index for the ray, and X is the ray length which is returned.

Figure E-2 gives the equations for the length of the line segments which intersect the N×N array. The variable x is the vertical distance as measured from the point (N, 1). The graphs given in the figure are for angles less than 90°, but these same results hold for angles greater than 90° and one is directed to the listing of SUBROUTINE XLENGH(M, R, X) for the implementation thereof. We have taken the length of the ray K to be the maximum length of all lines which subtend the array between the lines K-1 and K.

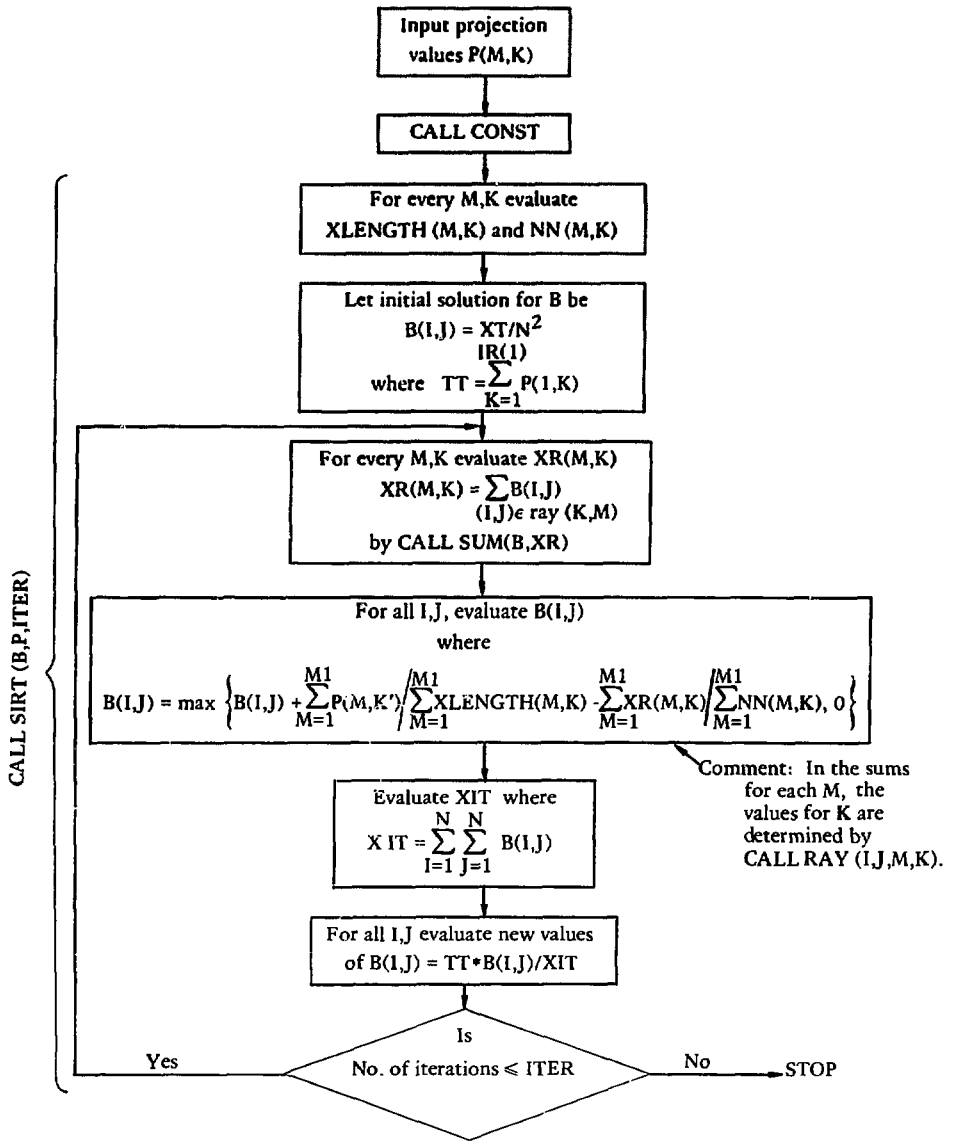
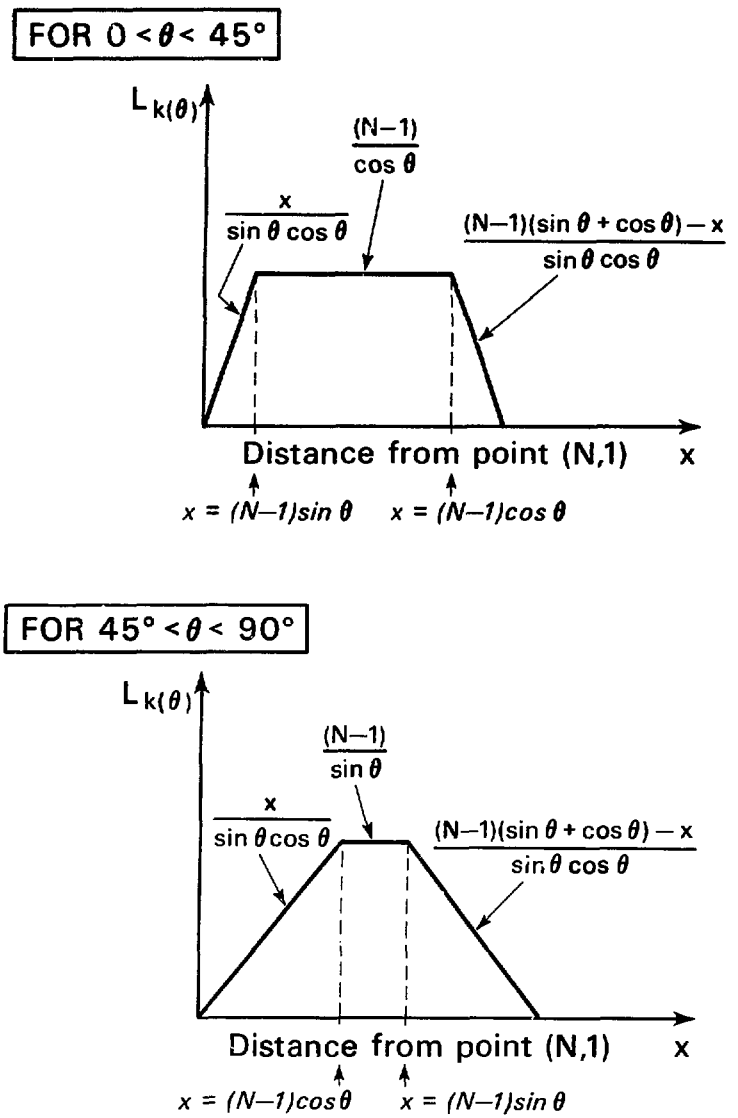


Figure E-1. Flow chart for the SIRT algorithm.



DBL 741-4629

Figure E-2. The length of ray segments.

```
      SUBROUTINE SIRT(B,P,ITER)
      DIMENSION B(46,46),XLENGTH(36,64),NN(36,64),XR(36,64),P(36,64)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      DO 11 M=1,M1
      MR=IR(M)
      DO 11 K=1,MR
      CALL XLENGH(M,K,XLENGTH(M,K))
   11 CONTINUE
      DO 1 I=1,N
      DO 1 J=1,N
      B(I,J)=1.
    1 CONTINUE
      CALL SUM(B,XR)
      DO 2 M=1,M1
      MR=IR(M)
      DO 2 K=1,MR
      NN(M,K)=INT(XR(M,K))
    2 CONTINUE
      TT=0.
      MR=IR(1)
      DO 5 I=1,MR
      TT=TT+P(1,I)
    5 CONTINUE
      DO 7 I=1,N
      DO 7 J=1,N
      B(I,J)=TT/FLOAT(N**2)
    7 CONTINUE
      K1=1
   79 CALL SUM(B,XR)
      DO 20 I1=1,N
      DO 20 J1=1,N
      IP2=0
      XL=0.
      R=0.
      N2=0
      DO 30 M=1,M1
      CALL RAY(I1,J1,M,K)
      XP2=XP2+P(M,K)
      XL=XL+XLENGTH(M,K)
      N2=N2+NN(M,K)
      R=R+XR(M,K)
   30 CONTINUE
      B(I1,J1)=B(I1,J1)+XP2/XL-R/FLOAT(N2)
      IF(B(I1,J1))43,20,20
   43 B(I1,J1)=0.
   20 CONTINUE
      XIT=0.
      DO 80 I=1,N
      DO 80 J=1,N
      XIT=XIT+B(I,J)
   80 CONTINUE
      DO 83 I=1,N
      DO 83 J=1,N
      B(I,J)=TT*B(I,J)/XIT
   83 CONTINUE
      K1=K1+1
      IF(K1-ITER)79,79,81
   81 RETURN
      END
```

```
      SUBROUTINE XLENGH(M,K,X)
      COMMON S(36),C(36),T(36),IK(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      IF(IA(M).EQ.0.OR.IA(M).EQ.90.OR.IA(M).EQ.180.OR.IA(M).EQ.270)15,17
   15 X=FLOAT(N1)
      GO TO 99
   17 IF(K.EQ.1)44,45
   44 XKK=0.
      GO TO 47
   45 XKK=FLOAT(K-1)-XL(M)+XN1*.5*(S(M)+C(M))
   47 XK=FLOAT(K)-XL(M)+XN1*.5*(S(M)+C(M))
      IF((IA(M).LT.45).OR.(IA(M).GT.135.AND.IA(M).LT.180).OR.(IA(M).GT.1
     C80.AND.IA(M).LT.225).OR.(IA(M).GT.315.AND.IA(M).LT.360))1,3
    1 XK1=XN1*S(M)
      XK2=XN1*C(M)
      IF(XK.LE.XK1)2,4
    4 IF(XKK.GT.XK2)16,6
    2 X=XK/S(M)/C(M)
      GO TO 99
    6 X=XN1/C(M)
      GO TO 99
   16 X=(XN1*(S(M)+C(M))-XKK)/S(M)/C(M)
      GO TO 99
    3 XK1=XN1*C(M)
      XK2=XN1*S(M)
      IF(XK.LE.XK1)2,20
   20 IF(XKK.GT.XK2)16,22
   22 X=XN1/S(M)
   99 RETURN
      END
```

Appendix F. Subroutine for Iterative Least-Squares Algorithm

Section (3.6) of the discussion gives the development and theory for the iterative least-squares algorithm; Fig. F-1 gives the flow chart. The computer listing given in this section assumes that each of the factors, f^{θ}_{ij}, is equal to 1. Therefore, for any emission study where the projection data is taken to be the conjugate mean of opposing views, the reconstructed image will not represent quantitatively the true image because of attenuation. However, for transmission studies where the projection data is taken to be the log of the ratio of the incident beam over the measured beam, the reconstructed image will be a true measure of the density distribution. See Appendix H for the description of the least-squares algorithm used for emission studies.

The image B is reconstructed by

CALL LESQ (B, P, ITER),

where B is the reconstructed array, P is the sampled projection data, and ITER is the number of iterations desired. Each projection value is denoted by P(M, K) where M is the index for the angle and K is the index for the ray. DEL(I, J) is the delta change for the pixel (I, J) as given by Eq. (32). The array XDEL contains the projections of the array DEL. It is assumed that the variance for each sampled projection is the value of that projection.

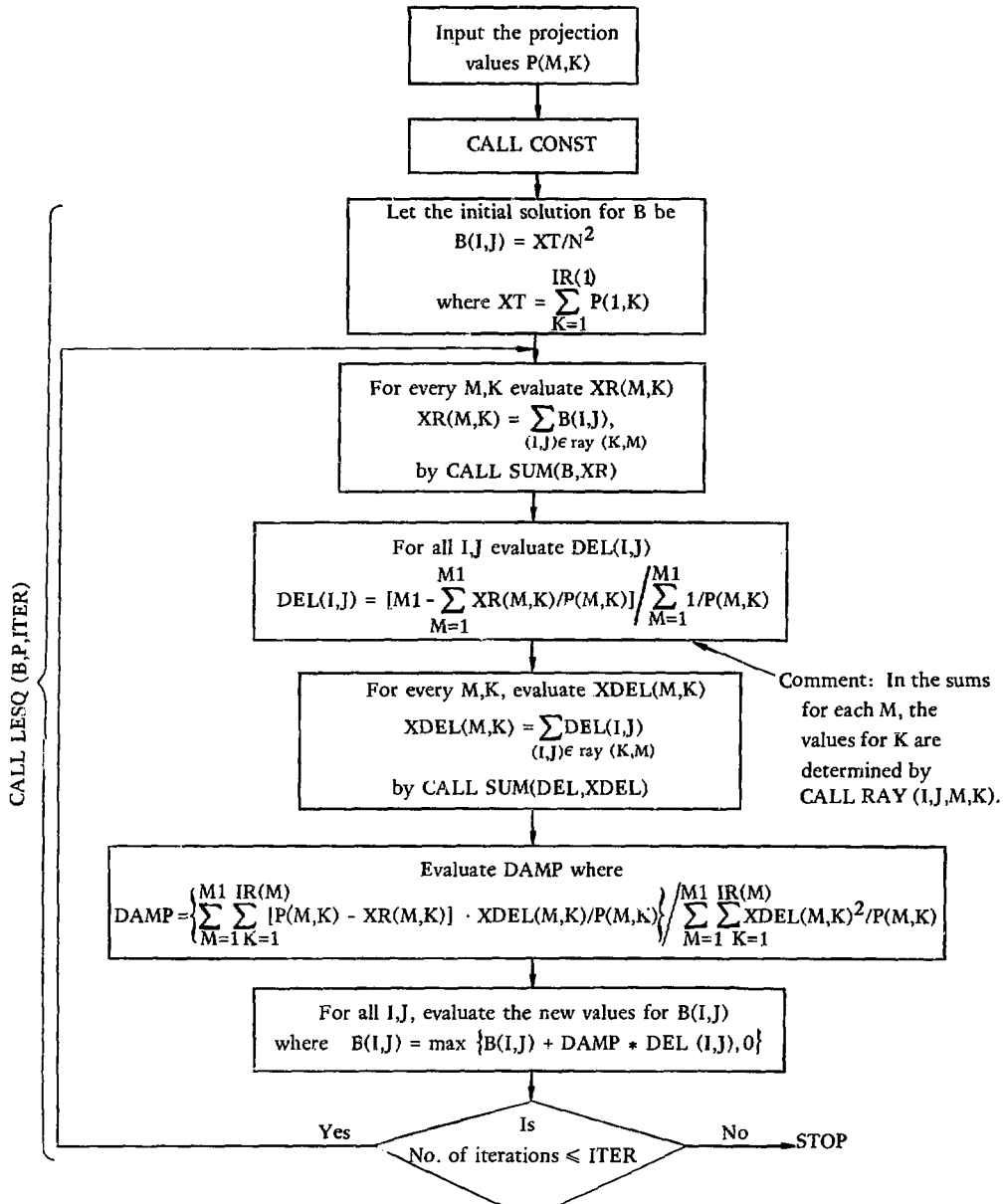


Figure F-1. Flow chart for the iterative least-squares algorithm.

```
      SUBROUTINE LESQ(B,P,ITER)
      DIMENSION B(46,46),DEL(46,46),P(36,64),XR(36,64),XDEL(36,64)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      XM1=FLOAT(M1)
      TT=0.
      MR=IR(1)
      DO 5 I=1,MR
      TT=TT+P(1,I)
    5 CONTINUE
      DO 7 I=1,N
      DO 7 J=1,N
      B(I,J)=TT/FLOAT(N**2)
    7 CONTINUE
      K1=1
   79 CALL SUM(B,XR)
      DO 20 I=1,N
      DO 20 J=1,N
      XIP2=0.
      R=0.
      DO 30 M=1,M1
      CALL RAY(I,J,M,K)
      XX=P(M,K)+1.
      XIP2=XIP2+1./XX
      R=R+XR(M,K)/XX
   30 CONTINUE
      DEL(I,J)=(XM1-R)/XIP2
   20 CONTINUE
      CALL SUM(DEL,XDEL)
      XNUM=0.
      DEM=0.
      DO 80 M=1,M1
      MR=IR(M)
      DO 80 K=1,MR
      XX=P(M,K)+1.
      XNUM=XNUM+XDEL(M,K)*(1.-XR(M,K)/XX)
      DEM=DEM+XDEL(M,K)**2/XX
   80 CONTINUE
      DAMP=XNUM/DEM
      DO 83 I=1,N
      DO 83 J=1,N
      B(I,J)=B(I,J)+DAMP*DEL(I,J)
      IF(B(I,J))10,83,83
   10 B(I,J)=0.
   83 CONTINUE
      K1=K1+1
      IF(K1-ITER)79,79,81
   81 RETURN
      END
```

Appendix G. Subroutine for Filtered Back-Projection Algorithm

The image B is reconstructed by the filtered back-projection algorithm (Fig. G-1) by

CALL FILTER(B, P)

where B is the reconstructed N×N array, and the array P contains the sampled projection data. Each projection value is denoted by P(M, K) where M is the index of the angle and K is the index of the ray.

In the listing of SUBROUTINE FILTER, one will notice that the projection data for each angle is first transferred to array A which has a fixed dimension. Therefore, for a particular projection P(M, K), K = 1, ..., IR(M), the array A has the following values

A(1)	real	0	
A(2)	imaginary	0	
A(3)	real	0	
A(4)	imaginary	0	
•			
•			
•			
A(I)	real	8	P(M, 1)
A(I+1)	imaginary	0	
A(I+2)	real	10	P(M, 2)
A(I+3)	imaginary	0	
•			
•			
•			
•			
A(I+2*IR(M)-1)	real	20	P[M, IR(M)]
A(I+2*IR(M))	imaginary	0	
A(I+2*IR(M)+1)	real	0	
A(I+2*IR(M)+2)	imaginary	0	
•			
•			
•			
A(127)	real	0	
A(128)	imaginary	0	

The array A is filled in so that the projection values of P are centered around a fixed center point of A. The array A is large enough to incorporate the values of the projection P(M, •) with the largest number of rays. Then the Fourier transform of A is taken by

CALL CFFT(MM, A, INV, SS, 1, ITER) .

The returned matrix A will now have the real and imaginary components of the Fourier transform. These components are then multiplied by the appropriate measure in Fourier space (the reciprocal space radius) and stored again in A. *

A(1)	⟶	A(1)
A(2)		A(2)
A(3)		A(3)
A(4)		A(4)
2*A(5)		A(5)
2*A(6)		A(6)
•		•
•		•
•		•
31*A(63)		A(63)
31*A(64)		A(64)
32*A(65)		A(65)
32*A(66)		A(66)
31*A(67)		A(67)
31*A(68)		A(68)
•		•
•		•
•		•
2*A(125)		A(125)
2*A(126)		A(126)
A(127)		A(127)
A(128)		A(128)

* At present this routine does not include a truncation at the maximum reasonable frequency though in our application the maximum frequency is close to the highest frequency component in the discrete Fourier transform. This truncation with a roll-off is being investigated at the time of this writing. The quantitative and noise amplification aspects of this algorithm reside in proper selection of the filter shape.

Then the inverse Fourier transform of A is obtained by

CALL CFFT(MM, A, INV, SS, -1, IFER),

where now the components of the inverse transform are stored in A. Then the components of A are mapped into the projection array P(K, M) by converting the real and imaginary parts of each term to the modulus:

A(1)	real	
A(2)	imaginary = 0	
•		
•		
•		
A(I)	real	$P(M, 1) = \sqrt{A(I)^2 + A(I+1)^2}$
A(I+1)	imaginary	
•		
•		
•		
A(I+2*IR(M)-1)	real	P(M, IR(M)) = Modulus
A(I+2*IR(M))	imaginary	
A(I+2*IR(M)+1)	real	
A(I+2*IR(M)+2)	imaginary	
•		
•		
•		
A(127)	real	
A(128)	imaginary	

After this has been done for all angles, the new projection values are then back-projected by

CALL BCKPROJ(B, P).

Remark: This technique gives good qualitative results, but is not quantitative and will seriously amplify noise because statistically weak Fourier coefficients are amplified and the ramp filter is not truncated at the maximum allowable frequency.

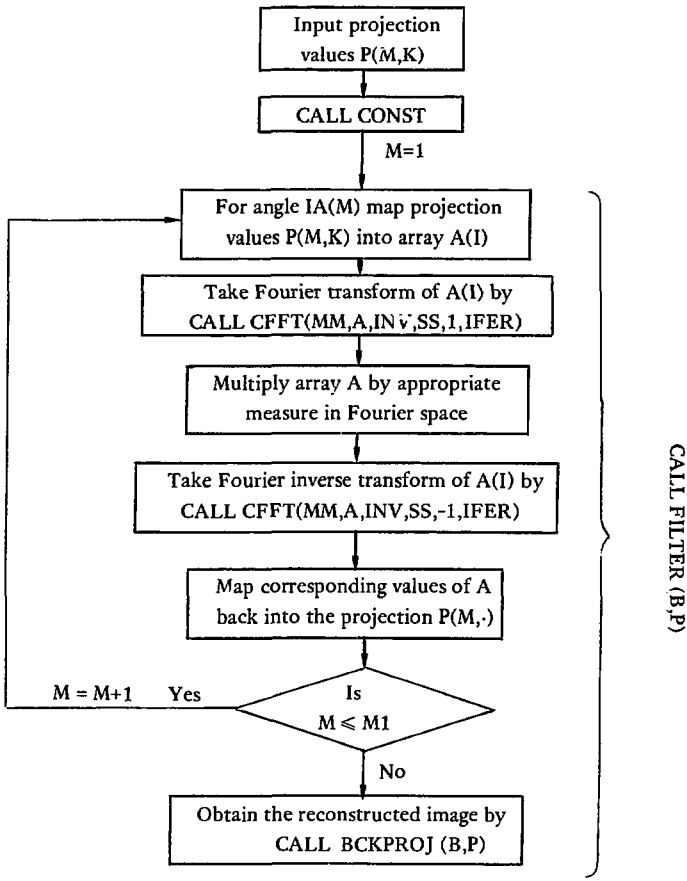


Figure G-1. Flow chart for the filtered back-projection algorithm.

```
      SUBROUTINE FILTER(B,P)
      DIMENSION B(46,46),P(36,64),MM(3),A(128),INV(128),SS(128)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      MM(1)=6
      MM(2)=0
      MM(3)=0
      ID=10
      PRINT 200
      DO 6 M=1,M1
      DO 5 I=1,128
      A(I)=0.
    5 CONTINUE
      MR=IR(M)
      IF(IA(M))310,314,310
  310 IF(IA(M)-90)315,314,315
  315 IZ=INT(.5*XN1*(S(M)+C(M)-1.)+.5)
      IU=ID-IZ-1
      GO TO 305
  314 IU=ID-1
  305 I=2*IU+1
      DO 7 K=1,MR
      A(I)=P(M,K)
      I=I+2
    7 CONTINUE
      CALL CFFT(MM,A,INV,SS,1,IFER)
      NN=128
      K=1
      DO 8 I=3,63,2
      A(I)=A(I)*FLOAT(K)
      A(I+1)=A(I+1)*FLOAT(K)
      A(NN+3-I)=A(NN+3-I)*FLOAT(K)
      A(NN-I+2)=A(NN-I+2)*FLOAT(K)
      K=K+1
    8 CONTINUE
      A(65)=A(65)*32.
      A(66)=A(66)*32.
      CALL CFFT(MM,A,INV,SS,-1,IFER)
      I=2*IU+1
      DO 9 K=1,MR
      IF(A(I))55,56,56
   55 A(I)=0.
      A(I+1)=0.
   56 P(M,K)=SQRT(A(I)**2+A(I+1)**2)
      I=I+2
    9 CONTINUE
    6 CONTINUE
      CALL BCKPROJ(B,P)
      RETURN
      END
```

```
      SUBROUTINE CFFT(M,A,INV,S,IFSET,IFERR)
      DIMENSION A(1),INV(1),S(1),N(3),M(3),NP(3),W(2),W2(2),W3(2)
      EQUIVALENCE (N1,N(1)),(N2,N(2)),(N3,N(3))
   10 IF( IABS(IFSET) - 1) 900,900,12
   12 MTT=MAXO(M(1),M(2),M(3)) -2
      IF(MTT.LT.1) GO TO 13
      MSUM=M(1)+M(2)+M(3)
      IF(MSUM.GT.15) GO TO 13
      ROOT2 = SQRT(2.)
      IF (MTT-MT ) 14,14,13
   13 IFERR=1
      RETURN
   14 IFERR=0
      M1=M(1)
      M2=M(2)
      M3=M(3)
      N1=2**M1
      N2=2**M2
      N3=2**M3
   16 IF(IFSET)20,18,18
   18 NX= N1*N2*N3
      FN = NX
      DO 19 I = 1,NX
      A(2*I-1) = A(2*I-1)/FN
   19 A(2*I) = -A(2*I)/FN
   20 NP(1)=N1*2
      NP(2)= NP(1)*N2
      NP(3)=NP(2)*N3
      DO 250 ID=1,3
      IL = NP(3)-NP(ID)
      IL1 = IL+1
      MI = M(ID)
      IF (MI)250,250,30
   30 IDIF=NP(ID)
      KBIT=NP(ID)
      MEV = 2*(MI/2)
      IF (MI - MEV )60,60,40
   40 KBIT=KBIT/2
      KL=KBIT-2
      DO 50 I=1,IL1,IDIF
      KLAST=KL+I
      DO 50 K=I,KLAST,2
      KD=K+KBIT
      T=A(KD)
      A(KD)=A(K)-T
      A(K)=A(K)+T
      T=A(KD+1)
      A(KD+1)=A(K+1)-T
   50 A(K+1)=A(K+1)+T
      IF (MI - 1)250,250,52
   52 LFIRST =3
      JLAST=1
      GO TO 70
```

From IBM program HARM cf. Budinger (1971).

```
 60 LFIRST = 2
    JLAST=0
 70 DO 240 L=LFIRST,MI,2
    JJDIF=KBIT
    KBIT=KBIT/4
    KL=KBIT-2
    DO 80 I=1,IL1,IDIF
    KLAST=I+KL
    DO 80 K=I,KLAST,2
    K1=K+KBIT
    K2=K1+KBIT
    K3=K2+KBIT
    T=A(K2)
    A(K2)=A(K)-T
    A(K)=A(K)+T
    T=A(K2+1)
    A(K2+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    T=A(K3)
    A(K3)=A(K1)-T
    A(K1)=A(K1)+T
    T=A(K3+1)
    A(K3+1)=A(K1+1)-T
    A(K1+1)=A(K1+1)+T
    T=A(K1)
    A(K1)=A(K)-T
    A(K)=A(K)+T
    T=A(K1+1)
    A(K1+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    R=-A(K3+1)
    T = A(K3)
    A(K3)=A(K2)-R
    A(K2)=A(K2)+R
    A(K3+1)=A(K2+1)-T
 80 A(K2+1)=A(K2+1)+T
    IF (JLAST) 235,235,82
 82 JJ=JJDIF    +1
    ILAST= IL +JJ
    DO 85 I = JJ,ILAST,IDIF
    KLAST = KL+I
    DO 85 K=I,KLAST,2
    K1 = K+KBIT
    K2 = K1+KBIT
    K3 = K2+KBIT
    R =-A(K2+1)
    T = A(K2)
    A(K2) = A(K)-R
    A(K) = A(K)+R
    A(K2+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    AWR=A(K1)-A(K1+1)
    AWI = A(K1+1)+A(K1)
```

```
      R=-A(K3)-A(K3+1)
      T=A(K3)-A(K3+1)
      A(K3)=(AWR-R)/ROOT2
      A(K3+1)=(AWI-T)/ROOT2
      A(K1)=(AWR+R)/ROOT2
      A(K1+1)=(AWI+T)/ROOT2
      T= A(K1)
      A(K1)=A(K)-T
      A(K)=A(K)+T
      T=A(K1+1)
      A(K1+1)=A(K+1)-T
      A(K+1)=A(K+1)+T
      R=-A(K3+1)
      T=A(K3)
      A(K3)=A(K2)-R
      A(K2)=A(K2)+R
      A(K3+1)=A(K2+1)-T
   85 A(K2+1)=A(K2+1)+T
      IF(JLAST-1) 235,235,90
   90 JJ= JJ + JJDIF
      DO 230 J=2,JLAST
   96 I=INV(J+1)
   98 IC=NT-I
      W(1)=S(IC)
      W(2)=S(I)
      I2=2*I
      I2C=NT-I2
      IF(I2C)120,110,100
  100 W2(1)=S(I2C)
      W2(2)=S(I2)
      GO TO 130
  110 W2(1)=0.
      W2(2)=1.
      GO TO 130
  120 I2CC = I2C+NT
      I2C=-I2C
      W2(1)=-S(I2C)
      W2(2)=S(I2CC)
  130 I3=I+I2
      I3C=NT-I3
      IF(I3C)160,150,140
  140 W3(1)=S(I3C)
      W3(2)=S(I3)
      GO TO 200
  150 W3(1)=0.
      W3(2)=1.
      GO TO 200
  160 I3CC=I3C+NT
      IF(I3CC)190,180,170
  170 I3C=-I3C
      W3(1)=-S(I3C)
      W3(2)=S(I3CC)
      GO TO 200
```

```
180 W3(1)=-1.
    W3(2)=0.
    GO TO 200
190 I3CCC=NT+I3CC
    I3CC = -I3CC
    W3(1)=-S(I3CCC)
    W3(2)=-S(I3CC)
200 ILAST=IL+JJ
    DO 220 I=JJ,ILAST,IDIF
    KLAST=KL+I
    DO 220 K=I,KLAST,2
    K1=K+KBIT
    K2=K1+KBIT
    K3=K2+KBIT
    R=A(K2)*W2(1)-A(K2+1)*W2(2)
    T=A(K2)*W2(2)+A(K2+1)*W2(1)
    A(K2)=A(K)-R
    A(K)=A(K)+R
    A(K2+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    R=A(K3)*W3(1)-A(K3+1)*W3(2)
    T=A(K3)*W3(2)+A(K3+1)*W3(1)
    AWR=A(K1)*W(1)-A(K1+1)*W(2)
    AWI=A(K1)*W(2)+A(K1+1)*W(1)
    A(K3)=AWR-R
    A(K3+1)=AWI-T
    A(K1)=AWR+R
    A(K1+1)=AWI+T
    T=A(K1)
    A(K1)=A(K)-T
    A(K)=A(K)+T
    T=A(K1+1)
    A(K1+1)=A(K+1)-T
    A(K+1)=A(K+1)+T
    R=-A(K3+1)
    T=A(K3)
    A(K3)=A(K2)-R
    A(K2)=A(K2)+R
    A(K3+1)=A(K2+1)-T
220 A(K2+1)=A(K2+1)+T
230 JJ=JJDIF+JJ
235 JLAST=4*JLAST+3
240 CONTINUE
250 CONTINUE
    NTSQ=NT*NT
    M3MT=M3-MT
350 IF(M3MT) 370,360,360
360 IG03=1
    N3VNT=N3/NT
    MINN3=NT
    GO TO 380
```

```
370 IGO3=2
    N3VNT=1
    NTVN3=NT/N3
    MINN3=N3
380 JJD3 = NTSQ/N3
    M2MT=M2-MT
450 IF (M2MT)470,460,460
460 IGO2=1
    N2VNT=N2/NT
    MINN2=NT
    GO TO 480
470 IGO2 = 2
    N2VNT=1
    NTVN2=NT/N2
    MINN2=N2
480 JJD2=NTSQ/N2
    M1MT=M1-MT
550 IF(M1MT)570,560,560
560 IGO1=1
    N1VNT=N1/NT
    MINN1=NT
    GO TO 580
570 IGO1=2
    N1VNT=1
    NTVN1=NT/N1
    MINN1=N1
580 JJD1=NTSQ/N1
600 JJ3=1
    J=1
    DO 880 JPP3=1,N3VNT
    IPP3=INV(JJ3)
    DO 870 JP3=1,MINN3
    GO TO (610,620),IGO3
610 IP3=INV(JP3)*N3VNT
    GO TO 630
620 IP3=INV(JP3)/NTVN3
630 I3=(IPP3+IP3)*N2
700 JJ2=1
    DO 870 JPP2=1,N2VNT
    IPP2=INV(JJ2)+I3
    DO 860 JP2=1,MINN2
    GO TO (710,720),IGO2
710 IP2=INV(JP2)*N2VNT
    GO TO 730
720 IP2=INV(JP2)/NTVN2
730 I2=(IPP2+IP2)*N1
800 JJ1=1
    DO 860 JPP1=1,N1VNT
    IPP1=INV(JJ1)+I2
    DO 850 JP1=1,MINN1
    GO TO (810,820),IGO1
810 IP1=INV(JP1)*N1VNT
    GO TO 830
```

```
  820 IP1=INV(JP1)/NTVN1
  830 I=2*(IPP1+IP1)+1
      IF (J-I) 840,845,845
  840 T=A(I)
      A(I)=A(J)
      A(J)=T
      T=A(I+1)
      A(I+1)=A(J+1)
      A(J+1)=T
  845 CONTINUE
  850 J=J+2
  860 JJ1=JJ1+JJD1
  870 JJ2=JJ2+JJD2
  880 JJ3 = JJ3+JJD3
890   IF(IFSET) 895,895,891
  891 DO 892 I = 1,NX
  892 A(2*I) = -A(2*I)
  895 RETURN
  900 MT=MAXO(M(1),M(2),M(3)) -2
      IF(MT.LT.1) GO TO 905
      MT = MAXO(2,MT)
  904 IF (MT-13)906,906,905
  905 IFERR = 1
      GO TO 895
  906 IFERR=0
      NT=2**MT
      NTV2=NT/2
  910 THETA=.7853981634
      JSTEP=NT
      JDIF=NTV2
      S(JDIF)=SIN(THETA)
      DO 950 L=2,MT
      THETA=THETA/2.
      JSTEP2=JSTEP
      JSTEP=JDIF
      JDIF=JSTEP/2
      S(JDIF)=SIN(THETA)
      JC1=NT-JDIF
      S(JC1)=COS(THETA)
      JLAST=NT-JSTEP2
      IF(JLAST - JSTEP) 950,920,920
  920 DO 940 J=JSTEP,JLAST,JSTEP
      JC=NT-J
      JD=J+JDIF
  940 S(JD)=S(J)*S(JC1)+S(JDIF)*S(JC)
  950 CONTINUE
```

```
960 MTLEXP=NTV2
    LM1EXP=1
    INV(1)=0
    DO 980 L=1,MT
    INV(LM1EXP+1) = MTLEXP
    DO 970 J=2,LM1EXP
    JJ=J+LM1EXP
970 INV(JJ)=INV(J)+MTLEXP
    MTLEXP=MTLEXP/2
980 LM1EXP=LM1EXP*2
982 IF(IFSET)12,895,12
    END
```

Appendix H. Subroutines for the Attenuation-Corrected Iterative Least-Squares Algorithm

The reconstruction of objects by using an emission source will have a density lower than the true image, due to attenuation, if the projection data are assumed to be

$$R_{k\theta} = \sum_{(i,j)\in ray(k,\theta)} A(i,j) . \tag{H1}$$

The results of this are discussed and shown in Section (3.9). To correct for this we have developed a least-squares iterative algorithm (Fig. H-1) which assumes that

$$R_{k\theta} = \sum_{(i,j)\in ray(k,\theta)} f_{ij}^{\theta} A(i,j) , \tag{H2}$$

where f_{ij}^{θ} are the attenuation factors. In order to evaluate the factors f_{ij}^{θ} the object is first reconstructed by CALL LESQ (B, P, ITER) for a few iterations, ITER, and where the projections P(M, K) are the geometric mean of opposing views. Then the shape of the object is outlined by assuming a particular threshold, XLEVAL, by

CALL SEARCH(B, BX, BOUNI, BOUNJ, XLEVAL, L)

where B is the N×N reconstructed array after implementating LESQ; BX is an N×N array used to define whether a pixel with values given by B is an interior, boundary, or exterior point; BOUNI, BOUNJ, give the x and y coordinates respectively for the boundary points; XLEVAL is a threshold value such that the boundary is determined if B(I, J) > MAX/XLEVAL, where MAX is the maximum value for B; and L is the total number of boundary points. Therefore, if the matrix B is

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 20 & 21 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 20 & 10 & 15 & 25 & 0 & 0 & 0 \\
0 & 0 & 20 & 15 & 6 & 10 & 21 & 0 & 0 & 0 \\
0 & 0 & 20 & 15 & 5 & 10 & 20 & 0 & 0 & 0 \\
0 & 0 & 0 & 21 & 22 & 20 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

then the matrix BX returned will be

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & -2 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & -2 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Note that the searching occurs first from each side, then from the top and bottom. It is assumed that the object is convex as will be the case for any brain scan.

Next the values f_{ij}^{θ} are determined by

CALL DIST(BX, BOUNI, BOUNJ, I1, J1, M, L, ATC, FACTOR, XD)

where BX is the N×N array described above; BOUNI and BOUNJ are the arrays containing the x and y coordinates respectively for the boundary points; I1 is the x coordinate and J1 is the y coordinate for the pixel whose FACTOR = f_{ij}^{θ} is desired; M is the index of the angle, L is the total number of boundary points, and ATC is the assumed attenuation coefficient; and XD is the distance of the pixel from the boundary if it is an interior point (XD = 0 if the pixel is an exterior point).

For interior points the factor, FACTOR = f_{ij}^{θ}, will be

FACTOR = EXP (-ATC*XD)

where XD is the distance of the pixel from the boundary, and where the distance is measured along a line at the angle IA(M) and in the direction of the projection. For exterior points, FACTOR will be zero since there is no density at these points for an emission study. This saves computer time in not having to determine the factors for exterior points.

Next the reconstructed array corrected for attenuation is determined by

CALL LESQC (B, PP, FACT, ITER)

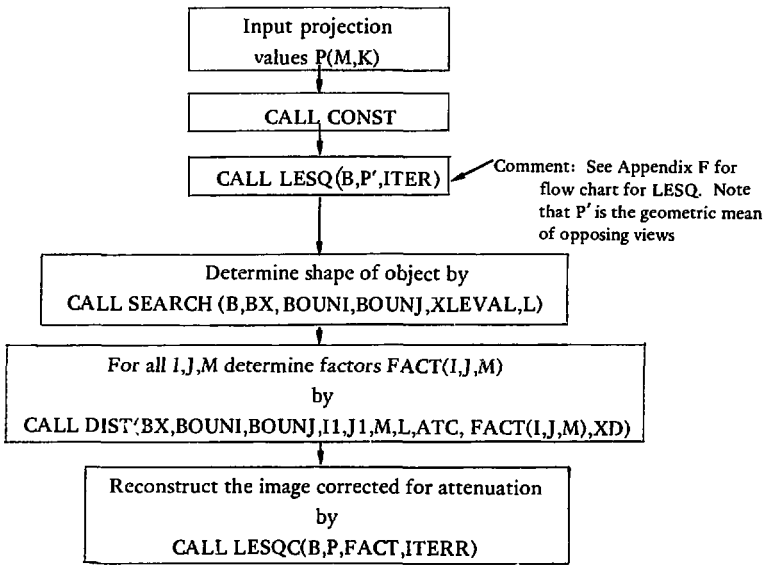
where B is the reconstructed array, PP is the sample projection data, ITER is the number of iterations, and FACT (I, J, M) is the array storing the factors f_{ij}^{θ}.

In executing this subroutine on the CDC 7600 at Lawrence Berkeley Labs, we had to use Large Core Memory in order to store the 46X46X36 array FACT; thus the dimension declaration, LARGE is used to allocate storage.

The sums for the projection data given by Eq. (H2) are obtained by

CALL SUMM (B, XR, FACT)

where B is an NXN array, XR is an array containing the projections of B, and FACT is the array storing the factors f_{ij}^{θ}. Each projection is denoted by XR(M, K) where M is the index of the angle and K is the index of the ray.



Comment: The flow chart for LESQC is similar to that given for LESQ in Appendix F. However, CALL SUMM(B,XR) replaces CALL SUM(B,XR) where in SUMM it is assumed that $XR(M,K)=\sum_{(I,J)\epsilon\ ray(K,M)} FACT(I,J,M) * B(I,J)$, and the equations for DEL and DAMP are

$$DEL(I,J) = \left\{\sum_{M=1}^{M1} FACT(I,J,M) * [1-XR(M,K)/P(M,K)]\right\} \Big/ \left\{\sum_{M=1}^{M1} FACT(I,J,M)^2/P(M,K)\right\}$$

$$DAMP = \left\{\sum_{M=1}^{M1}\sum_{K=1}^{IR(M)} XDEL(M,K) *[1-XR(M,K)/P(M,K)]\right\} \Big/ \left\{\sum_{M=1}^{M1}\sum_{K=1}^{IR(M)} XDEL(M,K)^2/P(M,K)\right\}$$

Figure H-1. Flow chart for the attenuation corrected iterative least-squares algorithm.

```
      SUBROUTINE SEARCH(B,BX,BOUNI,BOUNJ,XLEVAL,L)
      DIMENSION B(46,46),BX(46,46)
      INTEGER BOUNI(200),BOUNJ(200)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      MAX=0.
      NN=N/2+1
      L=0
      DO 11 I=1,N
      DO 11 J=1,N
      IF(B(I,J)-MAX)11,11,12
   12 MAX=B(I,J)
   11 CONTINUE
      DO 13 I=1,N
      DO 13 J=1,N
      BX(I,J)=0.
   13 CONTINUE
      DO 1 J=1,N
      I1=0
      I2=N+1
      IT1=1
      IT2=1
      DO 2 K=1,NN
      I1=I1+1
      I2=I2-1
      IF(IT1)3,3,4
    4 IF(B(I1,J)-MAX/XLEVAL)9,9,6
    6 BX(I1,J)=1.
      IT1=0
      L=L+1
      BOUNI(L)=I1
      BOUNJ(L)=J
      GO TO 9
    3 BX(I1,J)=-2.
    9 IF(IT2)10,10,7
    7 IF(B(I2,J)-MAX/XLEVAL)2,2,8
    8 BX(I2,J)=1.
      L=L+1
      BOUNI(L)=I2
      BOUNJ(L)=J
      IT2=0
      GO TO 2
   10 BX(I2,J)=-2.
    2 CONTINUE
    1 CONTINUE
      DO 21 I=1,N
      J1=0
      J2=N+1
      IT1=1
      IT2=1
      DO 22 K=1,NN
      J1=J1+1
      J2=J2-1
      IF(IT1)23,23,24
   24 IF(BX(I,J1))35,25,33
```

```
25 IF(B(I,J1)-MAX/XLEVAL)23,23,26
26 BX(I,J1)=1.
   L=L+1
   BOUNI(L)=I
   BOUNJ(L)=J1
33 IT1=0
   GO TO 23
35 IF(B(I,J1)-MAX/XLEVAL)36,36,37
37 BX(I,J1)=1.
   L=L+1
   BOUNI(L)=I
   BOUNJ(L)=J1
   IT1=0
   GO TO 23
36 BX(I,J1)=0.
23 IF(IT2)22,22,27
27 IF(BX(I,J2))30,28,31
28 IF(B(I,J2)-MAX/XLEVAL)22,22,29
29 BX(I,J2)=1.
   L=L+1
   BOUNI(L)=I
   BOUNJ(L)=J2
31 IT2=0
   GO TO 22
30 IF(B(I,J2)-MAX/XLEVAL)46,46,47
47 BX(I,J2)=1.
   L=L+1
   BOUNI(L)=I
   BOUNJ(L)=J2
   IT2=0
   GO TO 22
46 BX(I,J2)=0.
22 CONTINUE
21 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE DIST(BX,BOUNI,BOUNJ,I1,J1,M,L,ATC,FACTOR,XD)
      DIMENSION BX(46,46)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      INTEGER BOUNI(200),BOUNJ(200)
      FACTOR=1.
      XD=0.
      XI1=FLOAT(I1)
      XJ1=FLOAT(J1)
      IF(BX(I1,J1))200,201,200
  200 IF(IA(M))203,204,203
  203 IF(IA(M)-90)205,206,207
  207 IF(IA(M)-180)208,209,210
  210 IF(IA(M)-270)211,212,213
  204 I=I1
    1 I=I+1
      IF(BX(I,J1))1,999,2
    2 XD=FLOAT(I-I1)
      GO TO 999
  205 XLL=64.
      DO 3 K=1,L
      IF(I1.LT.BOUNI(K).AND.J1.LT.BOUNJ(K))4,3
    4 BI=FLOAT(BOUNI(K))
      BJ=FLOAT(BOUNJ(K))
      XLX=ABS(BJ-XJ1-(BI-XI1)*T(M))*C(M)
      IF(XLL-XLX)3,3,5
    5 XLL=XLX
      XD=SQRT((XI1-BI)**2+(XJ1-BJ)**2)
    3 CONTINUE
      GO TO 999
  206 J=J1
   11 J=J+1
      IF(BX(I1,J))11,999,12
   12 XD=FLOAT(J-J1)
      GO TO 999
  208 XLL=64.
      DO 33 K=1,L
      IF(I1.GT.BOUNI(K).AND.J1.LT.BOUNJ(K))34,33
   34 BI=FLOAT(BOUNI(K))
      BJ=FLOAT(BOUNJ(K))
      XLX=ABS(BJ-XJ1+(XI1-BI)*T(M))*C(M)
      IF(XLL-XLX)33,33,35
   35 XLL=XLX
      XD=SQRT((XI1-BI)**2+(XJ1-BJ)**2)
   33 CONTINUE
      GO TO 999
  209 I=I1
   41 I=I-1
      IF(BX(I,J1))41,999,42
   42 XD=FLOAT(I1-I)
      GO TO 999
  211 XLL=64.
```

```
      DO 53 K=1,L
      IF(I1.GT.BOUNI(K).AND.J1.GT.BOUNJ(K))54,53
   54 BI=FLOAT(BOUNI(K))
      BJ=FLOAT(BOUNJ(K))
      XLX=ABS(XJ1-BJ-(XI1-BI)*T(M))*C(M)
      IF(XLL-XLX)53,53,55
   55 XLL=XLX
      XD=SQRT((XI1-BI)**2+(XJ1-BJ)**2)
   53 CONTINUE
      GO TO 999
  212 J=J1
   61 J=J-1
      IF(BX(I1,J))61,999,62
   62 XD=FLOAT(J1-J)
      GO TO 999
  213 XLL=64.
      DO 73 K=1,L
      IF(I1.LT.BOUNI(K).AND.J1.GT.BOUNJ(K))74,73
   74 BI=FLOAT(BOUNI(K))
      BJ=FLOAT(BOUNJ(K))
      XLX=ABS(XJ1-BJ+(BI-XI1)*T(M))*C(M)
      IF(XLL-XLX)73,73,75
   75 XLL=XLX
      XD=SQRT((XI1-BI)**2+(XJ1-BJ)**2)
   73 CONTINUE
  999 FACTOR=EXP(-ATC*XD)
  201 RETURN
      END
```

```
      SUBROUTINE LESQC(B,PP,FACT,ITERR)
      DIMENSION B(46,46),DEL(46,46),PP(36,64),XR(36,64),XDEL(36,64)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      LARGE FACT(46,46,36)
      K1=1
   79 CALL SUMM(B,XR,FACT)
      DO 20 I=1,N
      DO 20 J=1,N
      XIP2=0.
      R=0.
      DO 30 M=1,M1
      CALL RAY(I,J,M,K)
      XX=PP(M,K)+1.
      XIP2=XIP2+FACT(I,J,M)**2/XX
      R=R+FACT(I,J,M)*(1.-XR(M,K)/XX)
   30 CONTINUE
      DEL(I,J)=R/XIP2
   20 CONTINUE
      CALL SUMM(DEL,XDEL,FACT)
      XNUM=0.
      DEM=0.
      DO 80 M=1,M1
      MR=IR(M)
      DO 80 K=1,MR
      XX=P(M,K)+1.
      XNUM=XNUM+XDEL(M,K)*(1.-XR(M,K)/XX)
      DEM=DEM+XDEL(M,K)**2/XX
   80 CONTINUE
      DAMP=XNUM/DEM
      DO 83 I=1,N
      DO 83 J=1,N
      B(I,J)=B(I,J)+DAMP*DEL(I,J)
      IF(B(I,J))10,83,83
   10 B(I,J)=0.
   83 CONTINUE
      K1=K1+1
      IF(K1-ITERR)79,79,81
   81 RETURN
      END
```

```
      SUBROUTINE SUMM(B,XR,FACT)
      DIMENSION B(46,46),XR(36,64)
      COMMON S(36),C(36),T(36),IR(36),XL(36),IA(36),XN,XN1,XN2,N,N1,M1
      LARGE FACT(46,46,36)
      DO 18 M=1,M1
      MR=IR(M)
      IF(IA(M).EQ.0.OR.IA(M).EQ.180)52,51
   51 IF(IA(M).EQ.90.OR.IA(M).EQ.270)54,53
   52 DO 19 K=1,MR
      XR(M,K)=0.
      DO 66 I=1,N
      XR(M,K)=XR(M,K)+FACT(I,K,M)*B(I,K)
   66 CONTINUE
   19 CONTINUE
      GO TO 18
   54 DO 61 K=1,MR
      XR(M,K)=0.
      DO 67 J=1,N
      XR(M,K)=XR(M,K)+B(K,J)*FACT(K,J,M)
   67 CONTINUE
   61 CONTINUE
      GO TO 18
   53 DO 63 K=1,MR
      XR(M,K)=0.
      CALL YMIN(K,M,IY1,IY2)
      DO 68 J=IY1,IY2
      CALL XMIN(K,M,J,IX1,IX2)
      DO 68 I=IX1,IX2
      XR(M,K)=XR(M,K)+B(I,J)*FACT(I,J,M)
   68 CONTINUE
   63 CONTINUE
   18 CONTINUE
      RETURN
      END
```

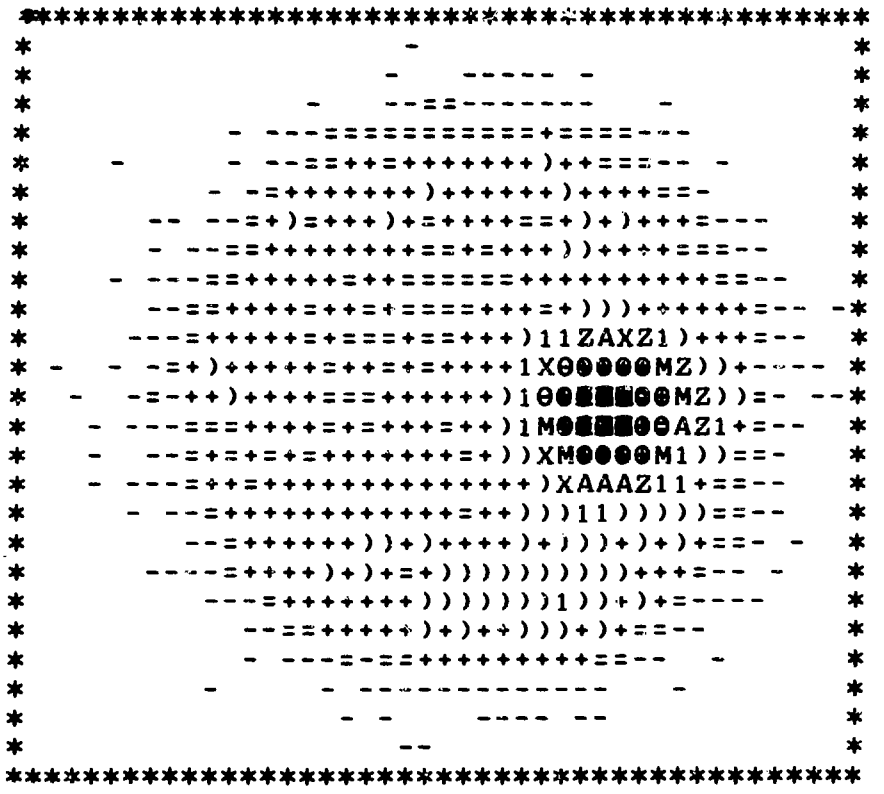
Appendix I. Subroutine for Array Imaging

The reconstructed array can be imaged, as in Fig. I-1, on the computer printout by

```
CALL ARAYPLT(B,N)
```

where B is an N×N array.

This subroutine utilizes the over-printing capability on the line printer at Lawrence Berkeley Laboratory. The line printer has ten characters per inch and six lines per inch. Therefore, the subroutine interpolates between the lines in order for the array to appear square. Some printers have eight lines per inch for which the subroutine would have to be changed accordingly. A printer with eight lines per inch is considerably more desirable. McLeod (1970) describes the particular algorithm used here.



XBL 741-228

Figure I-1. Example of overprinting technique used for CDC or IBM high-speed printer output display of images.

```
      SUBROUTINE ARAYPLT(B,N)
      DIMENSION B(46,46),LN1(128),LN2(128),LN3(128),LN4(128),LN5(128),LN
     26(128),LN7(128),LN8(128),DEN(21)
      INTEGER GRAY1(21),GRAY2(21),GRAY3(21),GRAY4(21),GRAY5(21),GRAY6(21
     2),GRAY7(21),GRAY8(21)
      DATA (GRAY1(I),I=1,21)/1H ,1H-,1H=,1H+,1H),1H1,1HZ,1HX,1HA,1HM,1HO
     2,1HC,1HO,1HO,1HO,1HO,1HO,1HC,1HO,1HO,1HO/
      DATA (GRAY2(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H-
     2,1H=,1H+,1H+,1H+,1H+,1HX,1H),1HX,1HX,1HX/
      DATA (GRAY3(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H
     2,1H ,1H ,1H ,1H ,1H ,1H',1H ,1H ,1H ,1H'/
      DATA (GRAY4(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H
     2,1H ,1H ,1H.,1H-,1H.,1H.,1H.,1H.,1H.,1H./
      DATA (GRAY5(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H
     2,1H ,1H ,1H ,1H ,1H=,1H-,1HF,1HH,1HH,1HH/
      DATA (GRAY6(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H
     2,1H ,1H ,1H ,1H ,1H ,1H ,1HC,1HB,1HB,1HB/
      DATA (GRAY7(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H
     2,1H ,1H ,1H ,1F ,1H ,1H ,1F ,1H ,1HV,1HV/
      DATA (GRAY8(I),I=1,21)/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H
     2,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1HA/
      DATA (DEN(I),I=1,21)/.0,.15,.22,.25,.29,.33,.37,.40,.42,.45,.53,.5
     26,.60,.64,.67,.79,.85,.89,.93,.97,1./
      XMAX=0.
      DO 500 I=1,N
      DO 500 J=1,N
      IF(B(I,J))501,500,500
  501 B(I,J)=0.
  500 CONTINUE
      DO 1 I=1,N
      DO 1 J=1,N
      IF(XMAX-B(I,J))33,1,1
   33 XMAX=B(I,J)
    1 CONTINUE
      IF(N.GT.100)23,24
   24 N1=N*6/10
      NN=(60-N1)/4-1
      DO 22 I=1,NN
      PRINT 101
  101 FORMAT(/)
   22 CONTINUE
   23 DO 2 I=1,128
      LN1(I)=1H
      LN2(I)=1H
      LN3(I)=1H
      LN4(I)=1H
      LN5(I)=1H
      LN6(I)=1H
      LN7(I)=1H
      LN8(I)=1H
    2 CONTINUE
      I1=(128-N)/2
      I2=I1+N-1
      III=I1-1
      II2=I2+1
```

```
      DO 12 I=II1,II2
      LN1(I)=1H*
   12 CONTINUE
      PRINT 1002,LN1
 1002 FORMAT(1X,128(A1))
      N1=N*6/10
      DO 3 K=1,N1
      J1=N-(K-1)*10/6
      J2=J1-1
      XJ=FLOAT(N)-FLOAT(K-1)*10./6.
      II=0
      DO 4 I=I1,I2
      II=II+1
      D=B(II,J1)+(B(II,J2)-B(II,J1))*(FLOAT(J1)-XJ)/FLOAT(J1-J2)
      D=D/XMAX
      DO 5 M=1,21
      IF(D-DEN(M))6,7,5
    5 CONTINUE
    6 M1=M-1
      M2=M
      T=(DEN(M2)+DEN(M1))/2.
      R=DEN(M2)-DEN(M1)
      D=D+R/2.-R*RANF(0.)
      IF(D-T)9,9,10
    9 L=M1
      GO TO 20
   10 L=M2
      GO TO 20
    7 L=M
   20 LN1(I)=GRAY1(L)
      LN2(I)=GRAY2(L)
      LN3(I)=GRAY3(L)
      LN4(I)=GRAY4(L)
      LN5(I)=GRAY5(L)
      LN6(I)=GRAY6(L)
      LN7(I)=GRAY7(L)
      LN8(I)=GRAY8(L)
    4 CONTINUE
      PRINT 1001,LN1
      PRINT 1001,LN2
      PRINT 1001,LN3
      PRINT 1001,LN4
      PRINT 1001,LN5
      PRINT 1001,LN6
      PRINT 1001,LN7
      PRINT 1000,LN8
 1000 FORMAT(1H ,128(A1))
 1001 FORMAT(1H+,128(A1))
    3 CONTINUE
      DO 13 I=II1,II2
      LN1(I)=1H*
   13 CONTINUE
      PRINT 1002,LN1
      RETURN
      END
```

BIBLIOGRAPHY

J. R. Andrews, Planigraphy. I. Introduction and history, Am. J. Roentgenol. Radium Ther. 36(5), 575-587 (1936).

H. O. Anger, The scintillation camera for radioisotope localization, in Radioisotope in der Lokalisations-diagnostik, G. Hoffmann and K. E. Sheer, Eds. (F. K. Schattauer-Verlag, Stuttgart, 1967) pp 18-21.

H. O. Anger, Tomography and other depth discrimination techniques, in Instrumentation in Nuclear Medicine, G. J. Hine and J. A. Sorenson, Eds. (Academic Press, N. Y., 1974) Vol. 2, pp. 61-100.

R. H. T. Bates and T. M. Peters, Towards improvements in tomography, New Zealand J. Sci. 14, 883-896 (1971).

E. V. Benton, R. P. Henke, and C. A. Tobias, Heavy-particle radiography, Science 182, 474-476, 1973.

M. V. Berry and D. F. Gibbs, The interpretation of optical projections, Proc. Roy. Soc. A314,143-152 (1970).

T. L. Boullion and P. L. Odell, Generalized Inverse Matrices (John Wiley & Sons, Inc. New York, 1971) 103 pp.

D. Boyd, J. Coonrod, J. Dehnert, D. Chiu, C. Lim, B. Macdonald and V. Perez-Mendez, A high-pressure xenon proportional chamber for x-ray laminographic reconstruction using fan beam geometry, in Proc. IEEE Nucl. Sci. Symp., San Francisco, November 14-16, 1973.

R. N. Bracewell and A. C. Riddle, Inversion of fan-beam scans in radio astronomy, Astrophys. J. 150(2), 427-434 (1967).

T. F. Budinger, Transfer Function Theory and Image Evaluation in Biology: Applications in Electron Microscopy and Nuclear Medicine, Ph.D. Thesis, University of California, Berkeley (1971).

T. F. Budinger, Clinical and research quantitative nuclear medicine system, in Medical Radioisotopes Scintigraphy 1972 (International Atomic Energy Agency, Vienna, 1973) Vol. 1, pp. 501-555.

T. F. Budinger, Quantitative nuclear medicine imaging: Application of computers to the gamma camera and whole-body scanner, in Recent Advances in Nuclear Medicine, J. H. Lawrence, Ed. (Grune and Stratton, Inc., N. Y., 1974) Chapter 2, in press.

T. F. Budinger and J. Harpootlian, Developments in digital computer implementation in nuclear medicine imaging, Lawrence Berkeley Laboratory Report LBL-2186, 1973.

D. A. Chesler, in Tomographic Imaging in Nuclear Medicine, G. S. Freeman, Ed. (Soc. Nucl. Med., New York, 1972) p 176.

E. H. Cho, I. Ahn, C. Bohm, and G. Huth, Computerized image reconstruction methods with multiple photon/x-ray transmission scanning, Phys. Med. Biol. (1974) in press.

A. M. Cormack, Reconstruction of densities from their projections, with applications in radiological physics, Phys. Med. Biol. 18(2), 195-207 (1973).

R. A. Crowther, D. J. DeRosier, and A. Klug. The reconstruction of a three-dimensional structure from projections and its application to electron microscopy, Proc. Roy. Soc. Lond. A317, 319-340 (1970).

D. J. DeRosier and A. Klug. Reconstruction of three dimensional structures from electron micrographs, Nature 217, 130-134 (1968).

Ralph Deutsch, Estimation Theory (Prentice Hall Inc., Englewood Cliffs, N. J., 1965) 259 pp.

P. Edholm, The tomogram-its formation and content, Acta Radiol. Suppl. #193, 1-109 (1960).

H. Fischgold, La nouvelle du jour: L' Emi-scanner. J. Radiol. Electrol. Med. Nucl. 54, 1-5 (1973).

J. Frank, Computer processing of electronmicrographs, in Advanced Techniques in Biological Electron Microscopy, J. K. Koehler, Ed. (Springer-Verlag, New York, 1973) pp 215-274.

G. Frieder and G. T. Herman, Resolution in reconstructing objects from electron micrographs, J. Theoret. Biol. 33, 189-211 (1971).

P. F. C. Gilbert, Iterative methods for the reconstruction of three-dimensional objects from projections, J. Theoret. Biol. 36, 105-117(1972a).

P. F. C. Gilbert, The reconstruction of a three-dimensional structure from projections and its application to electron microscopy II. Direct methods, Proc. Roy. Soc. (London) Ser B 182, 89-102 (1972b).

M. Goitein, Three-dimensional density reconstruction from a series of two-dimensional projections, Nucl. Instr. Methods 101(3), 509-518(1971).

R. Gordon, R. Bender, and G. T. Herman, Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and x-ray photography, J. Theoret. Biol. 29, 471-481 (1970).

R. Gordon and G. T. Herman, Three dimensional reconstruction from projections: A review of algorithms, Int. Rev. Cytol. (1973) in press.

P. V. Harper, The three-dimensional reconstruction of isotope distributions, in Fundamental Problems in Scanning, A. Gottschalk and R. N. Beck, Eds. (Charles C. Thomas, Springfield, 1968) pp 191-194.

R. G. Hart, Electron microscopy of unstained biological material: the polytropic montage, Science 159, 1464-1467 (1968).

G. T. Herman, Two direct methods for reconstructing pictures from their projections: A comparative study, Computer Graphics and Image Processing 1, 123-144 (1972).

G. T. Herman, A relaxation method for reconstructing objects from noisy x-rays, Tech. Rept. No. 63, Dept. Computer Sci., State University of New York at Buffalo (1973).

G. T. Herman, A. Lent, and S. Rowland, ART: Mathematics and applications (a report on the mathematical foundations and on the applicability to real data of the Algebraic Reconstruction Techniques), J. Theoret. Biol. (1973) in press.

F. John, Plane Waves and Spherical Means Applied to Partial Differential Equations, (Interscience, New York, 1955).

S. A. Johnson, G. T. Herman, R. A. Robb, J. F. Greenleaf, E. L. Ritman, S. L. Lee, and E. H. Wood, Dynamic three-dimensional reconstruction of beating heart from multiplanar roentgen-television images, Mayo Clinic ms report (1973).

R. L. Kashyap and M. C. Mittal, Picture reconstruction from projections, in Proc. First Int. Joint Conf. Pattern Recognition, Washington (1973) pp 286-292.

J. W. Keyes, Jr., and W. Simon, Computer techniques for radionuclide transverse section tomography and quantitative spatial (three-dimensional) imaging, in Proc. Third Symp. on Sharing of Computer Programs and Technol. in Nucl. Med., Miami, Florida, June 15-16 (1973). USAEC Report CONF-730627 (1973) pp 190-201.

A. Klug and R. A. Crowther, Three-dimensional image reconstruction from the viewpoint of information theory, Nature 238, 435-440 (1972).

D. E. Kuhl and R. Q. Edwards, Image separation radioisotope scanning, Radiology 80, 653-661 (1963).

D. E. Kuhl and R. Q. Edwards, Perforated tape recorder for digital scan data store with grey shade and numeric readout, J. Nucl. Med. 7, 269-280 (1966).

D. E. Kuhl and R. Q. Edwards, Reorganizing data from transverse section scans of the brain using digital processing, Radiology 91(5), 975-983 (1968).

D. E. Kuhl, R. Q. Edwards, A. R. Ricci, and M. Reivich, Quantitative section scanning using orthogonal tangent correction, J. Nucl. Med. 14(4), 196-200 (1973).

D. E. Lee, J. W. Keyes, and W. Simon, in Application of Optical Instrumentation in Medicine II, Proc. Soc. Photo. Optical Instr. Eng. (1973) in press.

I. D. G. McLeod, Pictorial output from a line-printer, IEEE Trans. Computers C-19, (1970) 160-162.

T. M. Peters, Image Reconstruction from Projections, Ph.D. Thesis, University of Canterbury, New Zealand (1973).

T. M. Peters, P. R. Smith, and R. D. Gibson, Computer aided transverse body-section radiography, Brit. J. Radiol. 46, 314-317 (1973).

J. Radon, Ueber die Bestimmung von Funktionen durch ihre integralwerte laengs gewisser Mannigfaltigkeiten. (On the determination of functions from their integrals along certain manifolds.), Ber. Saechs. Akad. Wiss. Leipzig, Math. Phys. Kl. 69, 262-277 (1917).

G. N. Ramachandran and A. V. Lakshminarayanan, Three-dimensional reconstruction from radiographs and electron micrographs: application of convolutions instead of Fourier transforms, Proc. Natl. Acad. Sci. U. S. 68(9), 2236-2240 (1971).

S. Reichmann, Modified theory of the development of tomographic blurring, Acta Radiol. 12, 433-451 (1972).

R. A. Robb, S. A. Johnson, M. Wondrow, and E. H. Wood, An operator-interactive, computer-controlled system for high fidelity digitization and analysis of biomedical images, in Proc. 17th Annual Tech. Meeting Soc. Photo. Optical Instr. Eng., San Diego, 7-10 August (1973).

S. S. Sandler, Direct three-dimensional analysis of electron micrograph pictures, Pattern Recognition 4(4), 353-359 (1972).

P. Schmidlin, Iterative separation of sections in tomographic scintigrams, Nucl. Med. 11, 1-16 (1972).

P. Schmidlin, Three-dimensional scintigraphy with an Anger camera and a digital computer, in Medical Radioisotope Scintigraphy 1972. (International Atomic Energy Agency, Vienna, 1973) pp 409-417.

P. Schmidlin, The method of iterative section separation in tomoscintigraphy, in Proc. Third Intern. Conf. Data Handling and Image Processing in Scintigraphy, Boston, 1973, S. M. Pizer, C. E. Metz, and G. L. Brownell, Eds. in press.

P. Schmidlin, H. Palmtag, and J. Clorius, Correction of organ motion by filtering techniques, in Proc. Third Symp. on Sharing Computer Programs and Technol. in Nucl. Med., Miami, Florida, June 15-16 (1973). USAEC Report CONF-730627 (1973) pp 53-63.

L. A. Shepp, Reconstructing soft body tissue from many x-ray transmissions. IEEE 1973 Nucl. Sci. Symp., in press.

P. R. Smith, T. M. Peters, and R. H. T. Bates, Image reconstruction from finite numbers of projections, J. Phys. A6, 361-382 (1973).

J. A. Sorenson, Methods for quantitating radioactivity in vivo by external counting measurements, Ph.D. Thesis, University of Wisconsin (1971).

V. W. Steward and A. M. Koehler, Proton radiography in the diagnosis of breast carcinoma, Radiology 110, 217-221 (1974).

B. K. Vainshtein, Finding the structure of objects from projections, Kristallografiya 15(5), 894-902 (1970). Translated in: Soviet Physics-Crystallography 15, 781-787 (1971).

B. K. Vainshtein, Three-dimensional electron microscopy of biological macromolecules, Usp. Fiz. Nauk 109(3), 455-497 (1973).

G. N. Watson, A Treatise on the Theory of Bessel Functions (Cambridge Univ. Press, 1966) 804 pp.

Index for Subroutines