

Analytic and Iterative Reconstruction Algorithms in SPECT*

Philippe P. Bruyant, PhD

Nuclear Spectroscopy and Image Processing Research Group, Biophysics Laboratory, Claude Bernard University, Lyon, France

Images of the inside of the human body can be obtained non-invasively using tomographic acquisition and processing techniques. In particular, these techniques are commonly used to obtain images of a γ -emitter distribution after its administration in the human body. The reconstructed images are obtained given a set of their projections, acquired using rotating gamma cameras. A general overview of analytic and iterative methods of reconstruction in SPECT is presented with a special focus on filter backprojection (FBP), conjugate gradient, maximum likelihood expectation maximization, and maximum a posteriori expectation maximization algorithms. The FBP algorithm is faster than iterative algorithms, with the latter providing a framework for accurately modeling the emission and detection processes.

Key Words: iterative reconstruction algorithms; analytic reconstruction algorithm; SPECT

J Nucl Med 2002; 43:1343–1358

The basic principle of nuclear medicine imaging is the following: a γ -emitter-labeled pharmaceutical is administered to a subject and an external device, the gamma camera, detects the radioactivity stemming from the body, from 1 or several angles of views. The image obtained at 1 angle of view is the projection of the 3-dimensional (3D) distribution onto the 2-dimensional (2D) detector plane. Because of the projection operation, no information regarding the depth at which disintegrations occur is available; moreover, activities stemming from separate structures may overlap each other on the detector plane, and the contrast may be low. With only 1 projection image, it is impossible to determine the activity distribution because an infinite number of distributions can yield the same projection. It is as difficult as to find 2 values knowing only their sum. However, the overlap observed in the projections depends on the relative positions of the detector and of the structures inside the body. So, more information about the

relative positions can be obtained by acquiring projections over a large number of angles of view around the subject. The basic idea of SPECT is to obtain, as accurately as possible, an image of the γ -emitter distribution in any slice of the body, using projections of this image acquired by a rotating gamma camera from several angles of view.

Although many different algorithms for SPECT exist (1), this article presents the basics of the more widely used algorithms: filtered backprojection (FBP), conjugate gradient (CG), maximum likelihood expectation maximization (MLEM), and maximum a posteriori expectation maximization (MAPEM) algorithms. It is not an exhaustive presentation of reconstruction algorithms and, in particular, it does not deal with the 2D Fourier reconstruction (2,3), the algorithms using linograms (4,5), Chebyshev polynomials (6), nonlinear backprojection (7), or those used when projections are acquired with a pinhole collimator (8). We shall not specifically address the reconstruction with fanbeam collimators, because in this case the projection data are usually rebinned to yield parallel projections (9) and, hence, the same reconstruction algorithm can be used for both sets of projections. Also, physical phenomena such as attenuation, scattering, and so forth are not under consideration here. Presentations of image reconstruction, without mathematics, can be read before reading this article (10,11). The appendices at the end of this article contain some important equations, very useful to better understand the principles described in the body of the text. In-depth, mathematic presentations of principles of CT can be found (1,12,13).

PRESENTATION

The 3D dataset of γ -emitter distribution is obtained in SPECT by piling up many slices usually reconstructed independently. For this reason, we will consider in the following the reconstruction of 1 slice only. Data acquisition is schematically illustrated in Figure 1. The detector rotates around the object of interest and allows one to observe the pattern of γ -emission in the field of view for many angles. We define $g(s, \theta)$ as the number of scintillations detected at any location s along the detector when the detector head is at an angular position θ . We also define the quantity $f(x, y)$ as the estimated number of photons emitted at any point (x, y) of the transverse slice in the field of view. This unknown quantity is assumed to

Received Jan. 14, 2002; revision accepted Jun. 10, 2002.

For correspondence or reprints contact: Philippe P. Bruyant, PhD, Division of Nuclear Medicine, University of Massachusetts Medical School, 55 Lake Ave. N., Worcester, MA 01655.

E-mail: Philippe.Bruyant@umassmed.edu

*NOTE: FOR CE CREDIT, YOU CAN ACCESS THIS ACTIVITY THROUGH THE SNM WEB SITE (http://www.snm.org/education/ce_online.html) THROUGH OCTOBER 2003.

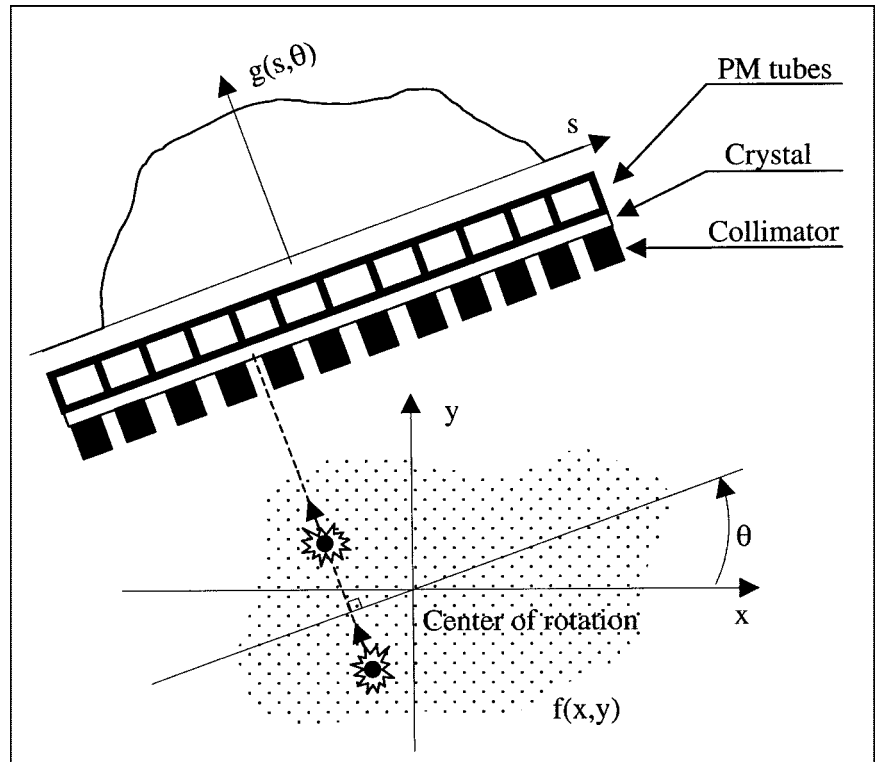


FIGURE 1. Principle of tomographic acquisition and geometric considerations. At each angle, data are projection of radioactivity distribution onto detector. Note that location of any scintillation onto crystal allows one to find out direction of incident photon (dashed line) but not to know distance between detector and emission site of photon.

be proportional to the tracer concentration we are interested in. The function g is the projection of f onto the crystal as allowed by the collimator. This means that $g(s, \theta)$ is the sum of the radioactive counts recorded in any time interval at point s when the detector is at angle θ . The collimator defines the kind of projection (e.g., an orthogonal projection for parallel collimators) and determines the direction of the incident photon for any scintillation in the crystal. The difficulty is that photons emitted at different depths, but along the same direction, can potentially produce scintillations at the same location in the crystal; thus, the distance between the emission site and the

scintillation site is unfortunately unknown. As a consequence, the amount of information brought by only 1 projection is insufficient to obtain an image of the tracer distribution in the organ of interest, because a large number of radioactive distributions can generate the same pattern in a single projection. Fortunately, the number of possible solutions can be reduced in acquiring projections for many distinct angular positions of the detector. Moreover, as the number of projections increases, the possible solutions are increasingly alike.

At the end of the acquisition process, each point of the detector, for each angular position, contains the number of

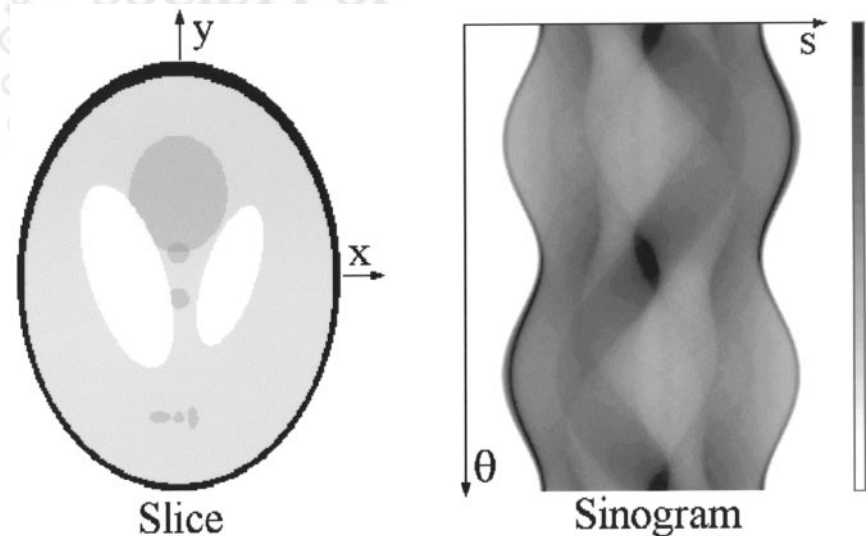


FIGURE 2. (Left) Shepp-Logan phantom slice (256×256 pixels). (Right) Corresponding sinogram, with 256 pixels per row and 256 angles equally spaced between 0° and 359° . Each row of sinogram is projection of slice at given angular position of detector.

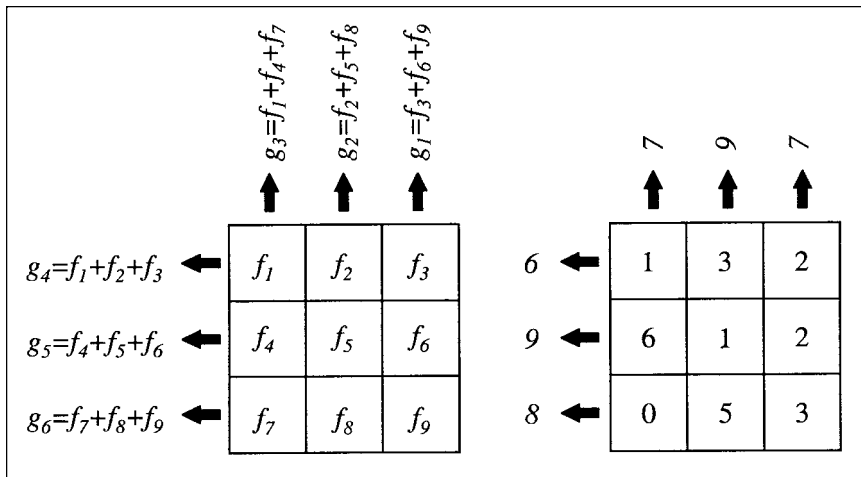


FIGURE 3. (Left) Principle of projection for one 3×3 slice at angle $\theta = 0$ and $\theta = 90^\circ$. Value in each bin is sum of values of pixels that project onto that bin. (Right) Example: $g_1 = f_3 + f_6 + f_9 = 2 + 2 + 3 = 7$. Result of projection is sinogram with 2 rows, whose values are (7, 9, 7) and (6, 9, 8).

scintillations, or counts. A common representation for the projection data g corresponding to a slice is the sinogram (Fig. 2). A sinogram is a 2D image, in which the horizontal axis represents the count location on the detector, and the vertical axis corresponds to the angular position of the detector. The corresponding number of counts is assigned to each point of the sinogram. The problem is: Given a sinogram g , what is the distribution of radioactivity f in the slice of interest? We shall first describe the projection process when a parallel collimator is used and then see how an estimate of the activity distribution in the slice can be found using the FBP, CG, MLEM, and MAPEM algorithms.

PROJECTION OPERATOR

Geometric Considerations

The collimator allows only photons whose direction is parallel to the axis of its holes to be potentially detected. For each detector angle θ , and for each location s on the detector, this direction is defined by a line D' whose equation is (Appendix 1):

$$s = x \cos \theta + y \sin \theta. \quad \text{Eq. 1}$$

Let us first consider the projection operation, giving the number of counts detected in any point of the detector line $g(s, \theta)$ as a function of the number of counts $f(x, y)$ emitted in any point of the field of view (here, we consider the ideal case, without noise, attenuation, scattering, numerical approximations, and so forth).

Radon Transform

Mathematically, the projection operator can be defined by the Radon transform (13,14). The Radon transform $g(s, \theta)$ of a function $f(x, y)$ is the line integral of the values of $f(x, y)$ along the line inclined at an angle θ from the x -axis at a distance s from the origin:

$$g(s, \theta) = \int_{-\infty}^{+\infty} f(s \cos \theta - u \sin \theta, s \sin \theta + u \cos \theta) du. \quad \text{Eq. 2}$$

Because an integral is basically a sum of values, the value $g(s, \theta)$ is the sum of the values $f(x, y)$ along D' . For this reason, $g(s, \theta)$ is called a ray-sum. The variable u defines the location of the points to be summed (the points along D').

Because computers can only deal with a finite number of elements in the detector and in the field of view, $g(s, \theta)$ and $f(x, y)$ are in practice functions of discrete variables—that is, the variables $s, \theta, x,$ and y have a finite number of possible values. In this presentation, the elements of the slice are the pixels, and each point of measurement on the detector, for each projection angle, is called a bin. So the number of bins equals the number of points of measurement multiplied by the number of angles. Figure 3 presents an example of a discrete projection for a 3×3 image at 2 angles. The result is a sinogram with 2 rows (corresponding to 2 projection angles) and 3 columns (corresponding to 3 points of measurements on the detector), so there are 6 bins.

From a mathematic point of view, the sets of values in the sinogram and the reconstructed slice can be considered as matrices or vectors. A matrix is a set of values organized in rows and columns; a vector is a set of values in 1 column. Readers unfamiliar with matrices should pay special attention to the easy concept of matrix product defined in Appendix 2, because it is a fundamental tool to understand image reconstruction. In Table 1 is listed the notation we will use throughout this presentation.

In the following, we will consider the sinogram and the slice as 2 vectors; the vectors are formed by stacking the rows of the matrices. In these vectors, the location of a bin is known by its index i and the location of a pixel by its index j . It can be shown that the vector g is the matrix product of matrix A and vector f (Appendix 3). The value g_i of any bin is a weighted sum of the m pixel values in the image:

$$g_i = a_{i1}f_1 + a_{i2}f_2 + \dots + a_{im}f_m = \sum_{j=1}^m a_{ij}f_j. \quad \text{Eq. 3}$$

TABLE 1
Notations

g	Vector of projection data
f	Vector of image data
A	Matrix such that $g = Af$
a_{ij}	Value of element located at i th row and j th column of matrix A
i	Projection subscript
j	Pixel subscript
g_i	Number of counts in i th bin of a projection dataset
\bar{g}_i	Mean value of g_i , assuming g_i is a Poisson random variable
f_j	Number of disintegrations in j th pixel of a slice
\bar{f}_j	Mean value of f_j assuming f_j is a Poisson random variable
m	Number of pixels
n	Number of bins

This is the discrete formula for the projection operation, and the essential point here is that the discrete projection operation can be defined as a matrix product. This product can be concisely written as $g = Af$, where A is called the forward projection operator (the adjective “forward” is usually omitted). In other words, the projection operation allows one to find the sinogram given the slice. An element of the matrix A is located in this matrix according to its row and its column. For example, the element a_{ij} is located at the i th row and j th column of the matrix. The point here is that any a_{ij} can be seen as a weighting factor representing the contribution of pixel j to the number of counts detected in bin i , or as the probability that a photon emitted from pixel j is detected in bin i . The use of zeros or ones in the matrix A can be interpreted as the binary choice “a given pixel contributes or does not contribute to the number of counts detected in any bin.” However, to get a more realistic model, we want to be able to modulate this contribution. To do so, the a_{ij} s may not be necessarily equal to 0 or 1 but can be any real values between 0 and 1. As we shall see below, the correct determination of the a_{ij} s is important because the iterative algorithms use the projection operation. The values are carefully chosen to take into account the geometry of acquisition and, more precisely, the fact that a variable

fraction of a pixel contributes to the counts detected in a given bin, depending on their relative positions and the angle of acquisition. Moreover, the camera response (that especially depends on the collimator) and physical phenomena such as attenuation and scatter can be efficiently modeled by choosing the appropriate values for the a_{ij} s. The capability to account for physical phenomena and for a nonideal imaging system is one of the advantages of iterative approaches compared with the FBP method.

Theoretically, direct methods exist to solve the system $g = Af$. A method called direct inversion consists in the determination of the invert of A , noted A^{-1} , because $f = A^{-1}g$. This method has several flaws: (a) It is computationally intensive for current computers, even for 64×64 images; (b) A^{-1} may not exist; (c) A^{-1} may be not unique; (d) A^{-1} may be ill-conditioned—i.e., small changes in the data g may produce large differences in the result f —for example, when g is noisy.

Unfortunately, in practice, the matrix A is ill-conditioned, because of the noise in the projections. Thus, the direct methods are not widely employed. We shall present now the algorithms commonly used in SPECT: the FBP, CG, MLEM, and MAPEM algorithms.

ANALYTIC RECONSTRUCTION METHODS

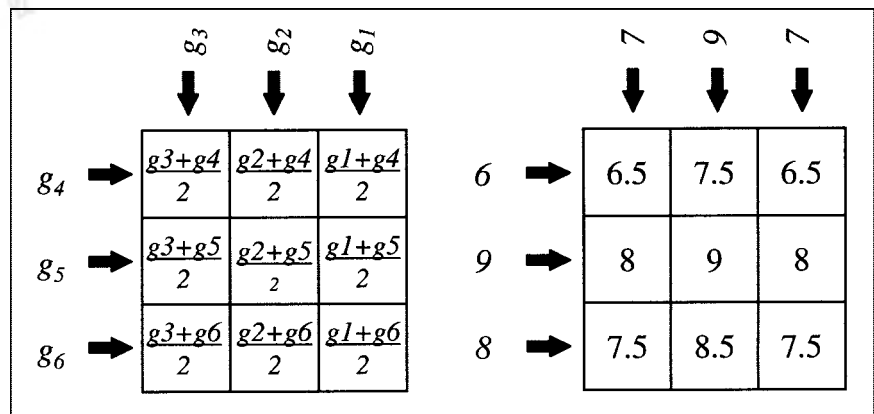
Backprojection Operator

Associated with the projection, the backprojection operation is defined as:

$$b(x, y) = \int_0^\pi g(s, \theta) d\theta. \quad \text{Eq. 4}$$

Backprojection represents the accumulation of the ray-sums of all the rays that pass through any point $M(x, y)$. Remember that for a given point M and for a given projection angle θ , s is given by Equation 1. Applying backprojection to projection data is called the summation algorithm. In ideal conditions (in particular, without attenuation), the projections acquired at angles between π radians (180°) and 2π radians (360°) do not provide new information, because

FIGURE 4. (Left) Principle of backprojection for one 2×3 sinogram. Value in each pixel is sum of values of bins that, given angle of detector, can receive photons from that pixel and is divided by number of rows of sinogram. (Right) Example: $f_1 = (g_3 + g_4)/2 = (7 + 6)/2 = 6.5$. Compare this slice with that of Figure 3, and note that after 1 projection and 1 backprojection, initial slice is not retrieved.



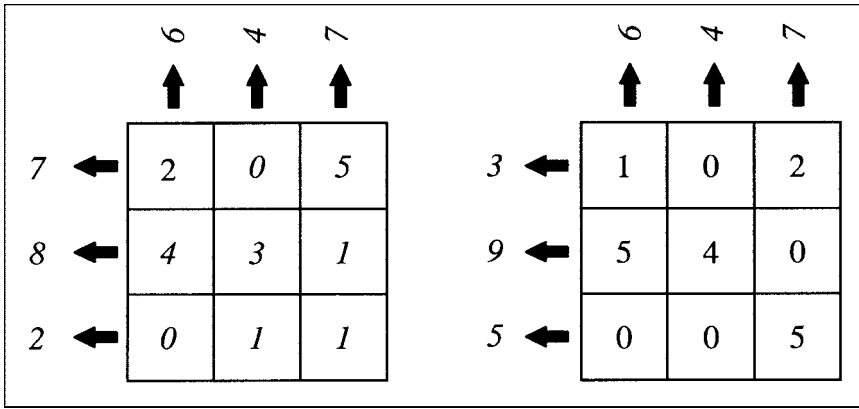


FIGURE 5. Example of 2 distinct images that can yield same projection at angle 0. This illustrates the fact that when number of projections is insufficient, solution (i.e., slice that yields projections) may be not unique.

they are only the symmetric of the projections acquired at angles between 0 and π . This is why the orbit can be here limited to π radians. This redundancy of information is illustrated in Figure 2, where the lower half of the sinogram is identical to the upper half after symmetry along the vertical axis (inverting the left and the right).

Replacing the integral in Equation 4 by a sum, the implementation for a discrete backprojection is given by:

$$\tilde{b}(x, y) = \sum_{k=1}^p g(s_k, \theta_k) \Delta\theta, \quad \text{Eq. 5}$$

where p is the number of projections acquired over π radians, θ_k is the k th angular position of the detector, s_k is the location along the detector, and $\Delta\theta$ is the angular step between 2 successive projections ($\Delta\theta = \pi/p$). In the following, for sake of simplicity, we will use $1/p$ as the angular step and will ignore the factor π . Equation 5 can be interpreted as follows: The goal is to find $\tilde{b}(x, y)$, the result of the backprojection at point $M(x, y)$. For each angle θ_k , using Equation 1, find the location s_k (on the detector) that is the projection location of point $M(x, y)$. Add the value $g(s_k, \theta_k)$ to the current value of point $M(x, y)$ (initial value should be 0). Repeat this process for all angles, and divide the sum by the number of projection angles. A simplified example of backprojection can be seen in Figure 4. Distinct images can

yield the same projection(s), but only for a limited set of angles; if 2 images are distinct, then one can find 1 or more angles at which their projections are different (Fig. 5). An infinite number of projections is theoretically required to perfectly reconstruct an object. When the number of projections is small relative to the matrix size, a particular artifact may be visible: the star (or streak) artifact. An illustration of this artifact is presented Figure 6. It can be reduced by increasing the number of acquired projections, or by interpolation (15).

The contribution of a given pixel to a given bin is variable, because it depends on their relative positions and the angle of acquisition. This has to be taken into account during the backprojection. At least 3 practical solutions exist: With the ray-driven method (16), the value in each bin (i.e., the ray-sum) is added to the image at pixels the ray goes through, but the addition is weighted according to the ray length in each pixel (Fig. 7A). With the pixel-driven approach, the projection site of the center of each pixel is calculated; the ray-sum value at this particular projection site is determined by interpolation using the values in the closest bins (Fig. 7B). A third approach is implemented in 3 steps: A line of the sinogram is replicated into the columns of a matrix (i.e., the line is backprojected). Then, this matrix is rotated about the center of rotation of the imaging system.

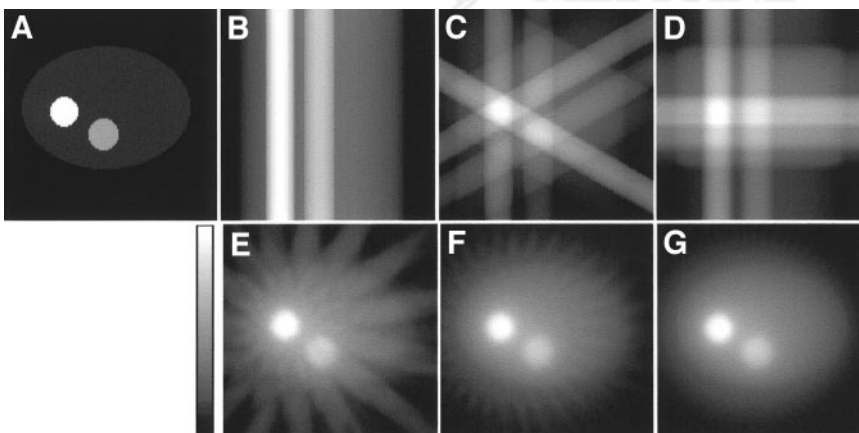


FIGURE 6. Illustration of star (or streak) artifact. (A) Slice used to create projections. (B–G) 1, 3, 4, 16, 32, and 64 projections equally distributed over 360° are used to reconstruct slice using backprojection algorithm. Activity in reconstructed image is not located exclusively in original source location, but part of it is also present along each line of backprojection. As number of projections increases, star artifact decreases.

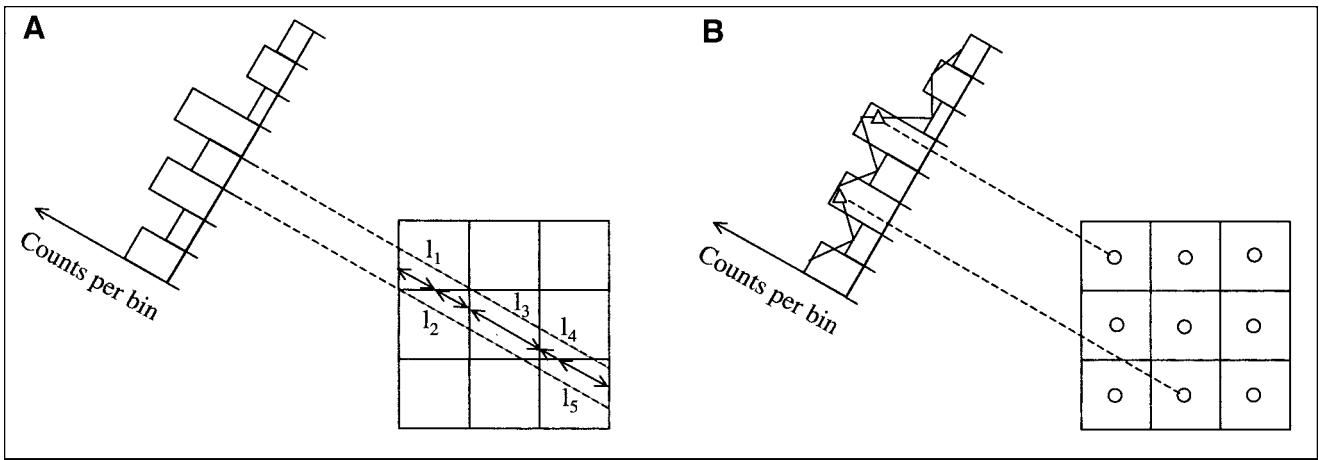


FIGURE 7. Modelization of geometry of backprojection. (A) With ray-driven backprojection, value attributed to each pixel along path is proportional to line length ($l_1, l_2 \dots l_5$). (B) With pixel-driven backprojection, center of each pixel is projected (dashed lines) and value attributed to each pixel is given by linear interpolation of values of closest bins (Δ).

Finally, this matrix is added to the matrix of the image under reconstruction (17).

The main point of this part is that the backprojection operation is not the inverse of the projection operation. This means that applying backprojection to the ray-sums $g(s, \theta)$ does not yield $f(x, y)$ but a blurred $f(x, y)$ (Figs. 8 and 9). This problem is due to the fact that during the backprojection process, each bin value is attributed to all pixels that project onto that bin and not only to the pixel(s) the activity is coming from (because the source location is unknown). In the following, we shall see how to reduce this blur.

Backprojection Filtering and FBP

It is important to understand the concept of spatial frequency to apprehend the role of filtering in image recon-

struction. The frequencies of an image are similar to the frequencies of a sound. A high frequency in a sound is heard when its amplitude varies quickly over a given period. A high frequency in an image is seen when its amplitude in any direction varies quickly over a given distance (and that is why the frequencies in an image are called spatial frequencies). A checkerboard with 4 large squares or with 64 small squares is an example of an image with low or high spatial frequencies, respectively. Any image will usually include several frequencies, when there are large and small objects in it. Using a mathematic tool called the Fourier transform, an image can be split up in several components, each corresponding to a given frequency. For sake of simplicity, we shall consider here only 2 components, the low-

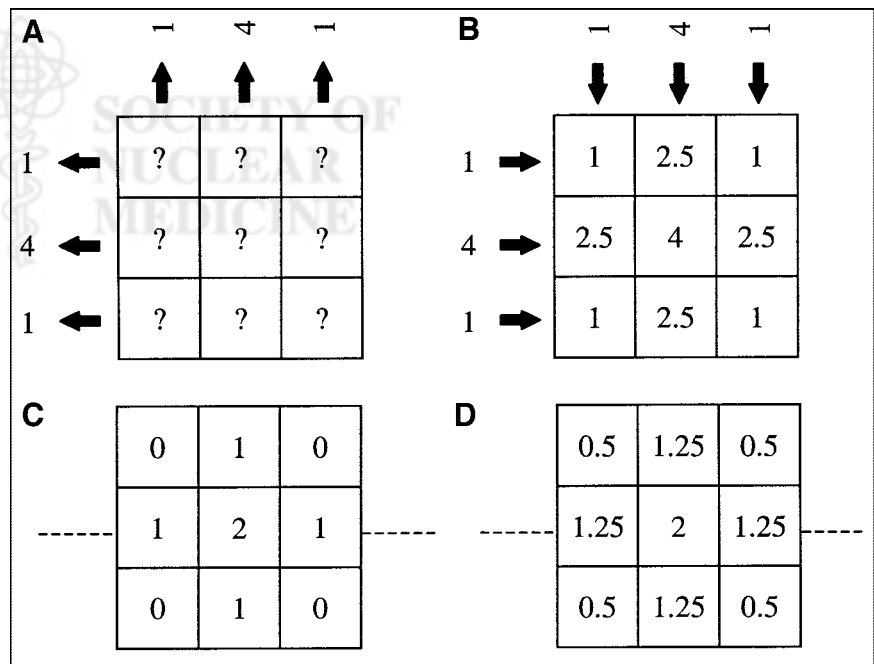


FIGURE 8. Blur introduced by backprojection. (A) Projection data are given. (B) Backprojection allows one to find values for 9 pixels. (C) Original image, whose projections are given in A, is shown. To compare original image and reconstructed image, image in B has been arbitrarily normalized to same maximum as original image: (D) Result is presented. Note how absolute difference between any 2 pixels is lower in D than in C.

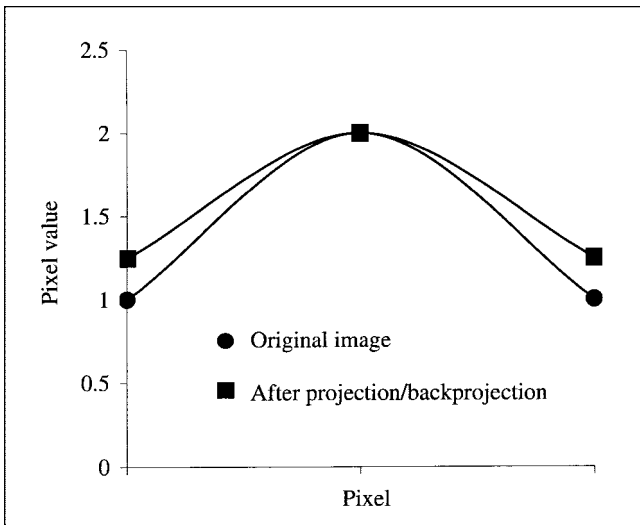


FIGURE 9. Activity profiles drawn along dashed lines in Figure 8. More gentle curve of profile after backprojection is illustration of blur.

and high-frequency components. If we add the 2 components, we will get the original image back. Let us now imagine that the image is rebuilt by adding the 2 components, but suppose that, before the addition, we change the relative weights of the 2 components: For example, we may divide the low-frequency component by a factor of 5 to decrease the blur that is typically present in the low-frequency components. Thus, the image has been filtered to reduce the amplitude of the low-frequency component (the same way as

you reduce the bass of your hi-fi using its graphic equalizer), and the result is an image in which the edges are better defined (Fig. 10). Coming back to our problem of image reconstruction, a first solution to decrease the blur is to backproject the projection dataset (using the summation algorithm) and then to filter the result: This is the backprojection filtering (BPF) algorithm. In practice, the image is split up in a large number of frequency components, and a filter defines the weight assigned to each of the components.

With the more commonly used FBP algorithm, the order of filtering and backprojection is reversed: Each line of the sinogram is filtered first, and the filtered projections are backprojected (18). This is mathematically expressed as:

$$f(x, y) = \int_0^\pi \hat{g}(s, \theta) d\theta, \quad \text{Eq. 6}$$

where $\hat{g}(s, \theta)$ is $g(s, \theta)$ filtered with the ramp filter, which gives a weight proportional to its frequency to each of the components. Notice that with this approach, the amplitude of the low-frequency components is reduced in the sinogram, row by row, before the backprojection. The action of filtering in FBP can be understood as follows: After filtering, the projections may contain negative values; the blur is reduced because negative and positive values cancel each other in the vicinity of the edges of the image, so that the edges are sharpened (19) (Fig. 11). Therefore, in the reconstructed image the high-frequency components corresponding to the sharp edges in the image are enhanced, but

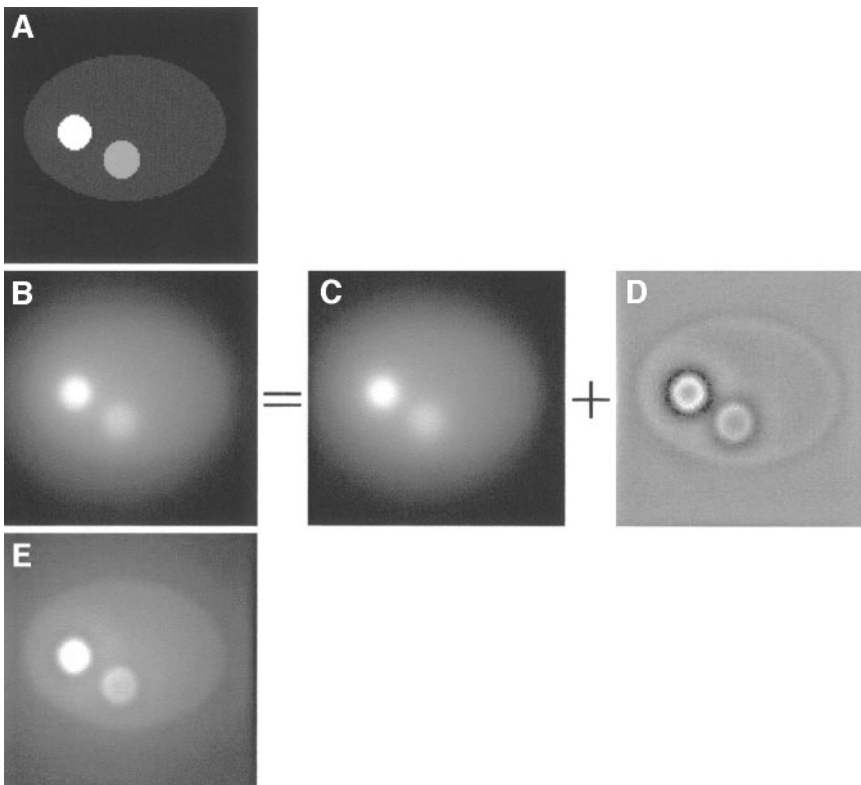


FIGURE 10. Simplified illustration of filtering process. (A) Model (128×128 pixels). (B) Image obtained after backprojection of 128 projections. (C) Low-frequency component of image presented in B. Only overall aspect of image is visible. (D) High-frequency component of image presented in B. Edges are emphasized. Dark rings correspond to negative pixel values. Sum of images in C and D yields image in B. (E) Images in C and D are added, but after C is given low weight to reduce amplitude of low-frequency component.

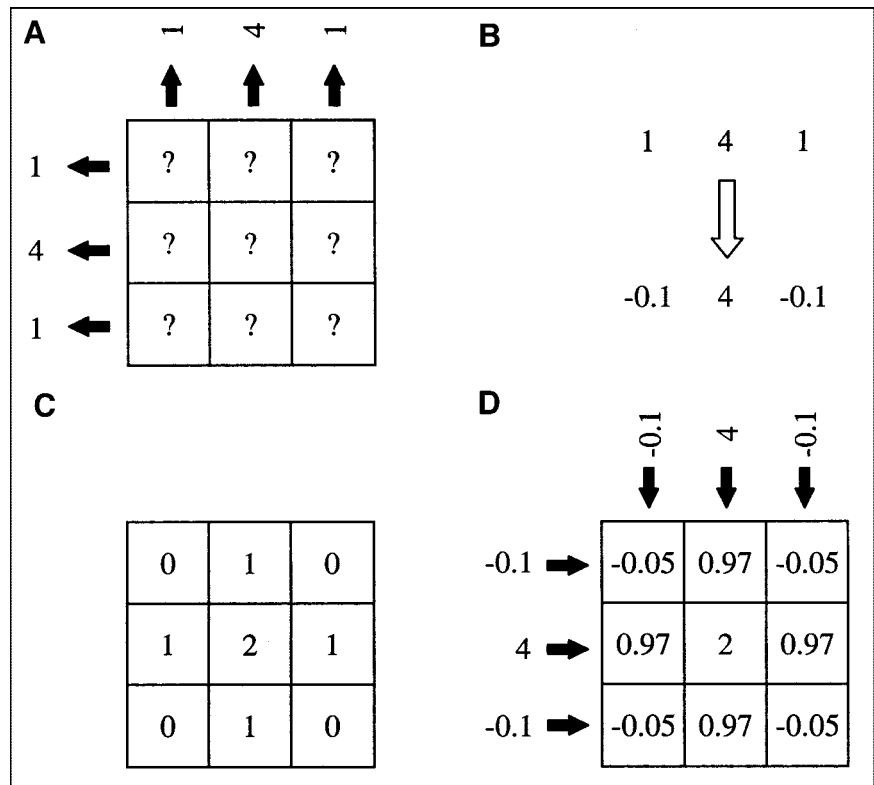


FIGURE 11. (A) Two projections are same as in Figure 8. (B) Filtering of projections using ramp filter yields negative values. (C) Original image. (D) Image obtained after backprojection of filtered projections. Note how negative and positive values substantially cancel each other, yielding result closer to original image that can be seen in Figure 8D.

unfortunately much of the image noise is present at these frequencies. This problem can be partially solved by replacing the ramp filter with a filter that also reduces the weight of the highest-frequency components (Fig. 12). As always,

there are trade-offs involved; the reduction in noise comes at the cost of a loss of spatial resolution. More details on image filtering in nuclear medicine can be found elsewhere (18–20).

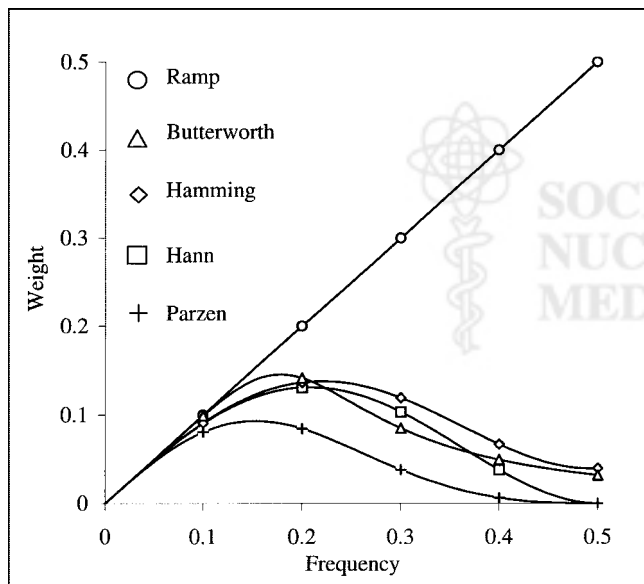


FIGURE 12. Some filters currently used in FBP and their shape. Value on y-axis indicates to what extent contribution of each frequency to image is modified by filters. These filters, except ramp filter, simultaneously reduce high-frequency components (containing much noise) and low-frequency component (containing blur introduced by summation algorithm).

ITERATIVE RECONSTRUCTION METHODS

General Concept of Iterative Methods

We are interested in finding a vector f that is a solution of $g = Af$. The principle of the iterative algorithms is to find a solution by successive estimates. The projections corresponding to the current estimate are compared with the measured projections. The result of the comparison is used to modify the current estimate, thereby creating a new estimate. The algorithms differ in the way the measured and estimated projections are compared and the kind of correction applied to the current estimate. The process is initiated by arbitrarily creating a first estimate—for example, a uniform image initialized to 0 or 1 (depending on whether the correction is carried out under the form of an addition or a multiplication).

For illustration purpose, we shall start with the additive form of the algebraic reconstruction technique (ART) as an example of iterative algorithm (21). The iterative process is given by:

$$f_j^{(k+1)} = f_j^{(k)} + \frac{g_i - \sum_{j=1}^N f_j^{(k)}}{N}, \quad \text{Eq. 7}$$

where $f_j^{(k)}$ and $f_j^{(k+1)}$ are the current and the new estimates, respectively; N is the number of pixels along ray i ; $\sum_{j=1}^N f_{ji}^{(k)}$ is the sum of counts in the N pixels along ray i , for the k th iteration; and g_i is the measured number of counts for ray i .

Observing Equation 7, we see that (a) the new estimate is found by adding a correction term to the current estimate and (b) the comparison method consists in the subtraction of the estimated projections from the measured projections.

Notice that when the projections of the current estimate are close to the measured projections, the correction factor is close to zero. This algorithm is applied to a 2×2 image (Fig. 13). The solution is found by successively applying the Equation 6 to the measured projections. The algorithms described below work basically the same way.

In the following, the CG algorithm (18,22) and the MLEM algorithm (23–27) are discussed. Both are optimization methods—that is, they find the best estimate for the solution fitting a given criterion; in the CG algorithm, the

criterion is the minimization of the difference between g and Af , whereas in the MLEM algorithm, the criterion is the maximization of the likelihood of the reconstructed image.

Gradient and CG Algorithms

Let us imagine that we have a huge number of images. For each of these images, we compute their projections, and we compare these projections to the measured projections by computing their difference. For some images this difference is large, for other images it is small. So if we plot this difference as a function of the values in the images, we can visualize the plot as a mountainous terrain. The visualization is easier if we first assume that an image has only 2 pixels, because in this case we can create a 3D plot, where the 2 horizontal axes are used to indicate the value of each pixel, and the vertical axis is used to plot the difference. However, the reasoning still holds for a larger number of pixels in the images, although the visualization is much

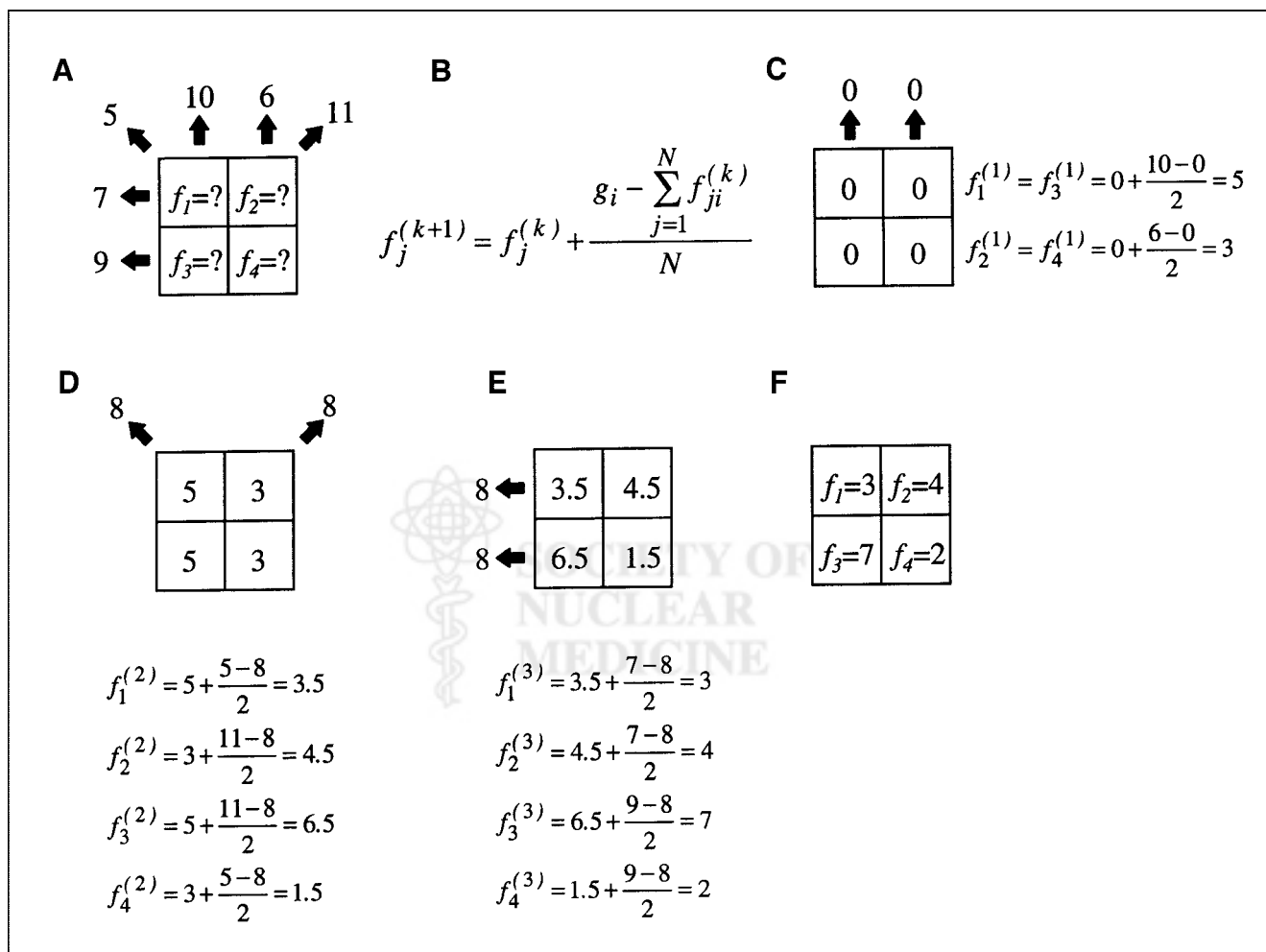


FIGURE 13. How ART algorithm works. (A) Problem is to find values of 4 pixels given values in 6 bins. (B) ART algorithm: Difference between estimated and measured projections is computed and divided by number of pixels in given direction. Result is added to current estimate. (C) First step: Project initial estimate (zeros) in vertical direction, apply ART algorithm, and update pixel values. Repeat this process for oblique (D) and horizontal (E) rays. (F) Solution is obtained after 1 full iteration. However, with larger images, more iterations are typically required.

more difficult because the number of dimensions of the plot is equal to the number of pixels in the images. The difference is minimal at the point of the lowest altitude. Our goal is to find the location of that point, because the coordinates of that point are the pixel values of the solution image. In the following, we present a method to find this point.

Actually, computing the difference for thousands of images would be a tedious and lengthy process. Again, we use the analogy with the mountainous terrain to devise a more efficient method: Basically, from a starting point on the mountain, we look for a point of lower altitude, we jump to this point, and this process is repeated until the point with the lowest altitude is reached. The problem is to optimize the direction and the length of the steps, so that the lowest point is reached in a minimal number of steps. Let us imagine the contours—that is, the curved lines joining the points with the same altitude. The gradient lines are curved lines perpendicular to the contours and along which the slope is constant. The more intuitive way to find, as quickly as possible, the point with the lowest altitude is to follow the steepest descent, whose direction is the opposite to the gradient. So, from an arbitrary location, we step in that direction, and the step length can be optimally chosen so that we stop before we are about to ascend (when the direction is parallel to a contour). Then, the gradient found for the new location indicates the new direction to follow, the new step length is found, and the process is repeated until the lowest location is found (22) (Fig. 14). This algorithm is called the gradient algorithm, or steepest-descent algorithm.

Mathematically, this algorithm iteratively searches for f using the equation:

$$f^{(k+1)} = f^{(k)} + \alpha^{(k)} p^{(k)}. \quad \text{Eq. 8}$$

This means that the new estimate $f^{(k+1)}$ is equal to the previous estimate $f^{(k)}$ plus a vector $p^{(k)}$ indicating the new

direction (chosen to be opposite to the local gradient, and therefore directed toward the steepest descent), weighted by a coefficient $\alpha^{(k)}$ representing the step length. This coefficient can be chosen to optimize the convergence of the process.

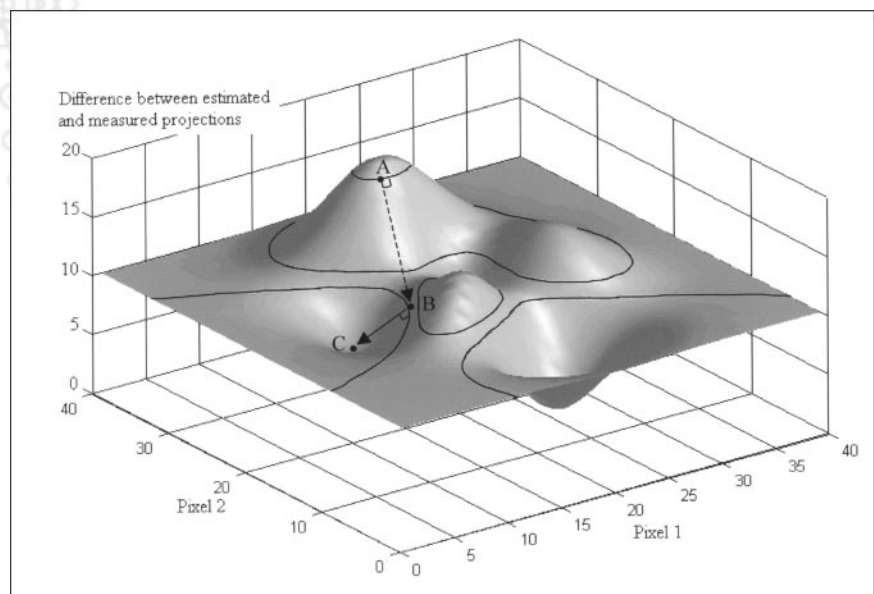
However, the direction given by the gradient method is only locally the best, and the gradient algorithm is not very efficient (i.e., a large number of iterations is usually required). The more efficient conjugate gradient algorithm is basically identical, but it uses a different vector $p^{(k)}$. In this case, the direction of descent is not opposite to the gradient at the current location, but opposite to a combination of this gradient with the gradient found at the previous location. In both gradient and CG algorithms, the correction is additive (each new estimate is found by adding a quantity to the current estimate). Notice that these algorithms may generate negative values in the reconstructed image.

In theory, the conjugate gradient algorithm applied to $g = Af$ converges in a number of iterations equal to the number m of equations of the system (i.e., the number of pixels here). Actually, because of rounding errors in the determination of the conjugate directions, the solution is not reached in m iterations. However, it can be shown that the convergence rate of reconstruction (reflected by the number of iterations required to reach a given estimate of the solution) can be increased by using $C^{-1}g = C^{-1}Af$ instead of $g = Af$, where C^{-1} is a matrix judiciously chosen. In this case, A is said to be preconditioned (28,29).

MLEM Algorithm

Again, one wants to solve $g = Af$, where g is the vector of values in the sinogram, A is a given matrix, and f is the unknown vector of pixel values in the image to be reconstructed. Measurements are subject to variations due to the Poisson probabilistic phenomena of the radioactive disintegrations. As a consequence, a dataset g corresponds to a

FIGURE 14. Gradient algorithm. This plot displays difference between estimated and measured projections (vertical axis) as function of values in 2 pixels of an image. Black lines are contour lines. Goal of algorithm is to find lowest point. From initial estimate for image (point A), step along steepest descent (dashed arrow) to reach point B. Then, at B, step along steepest descent (solid arrow) to reach minimum (point C). Values for 2 pixels at location C give solution. Note that, depending on location for starting point A, minimum can be different.



particular measurement; when the probabilistic phenomena are not taken into account, f is the particular solution corresponding to that particular measurement g . The goal of the MLEM algorithm is to find a “general” solution as the best estimate for f : the mean number of radioactive disintegrations \bar{f} in the image that can produce the sinogram g with the highest likelihood (23–27). This can be done using the Poisson law that allows one to predict the probability of a realized number of detected counts, given the mean number of disintegrations. Thus, each iteration of the algorithm is divided in 2 steps: in the expectation step (E step), the formula expressing the likelihood of any reconstructed image given the measured data is formed, and in the maximization step (M step), the image that has the greatest likelihood to give the measured data is found (26).

This leads to the MLEM algorithm as described by Lange and Carson (24) (Appendix 4):

$$\bar{f}_j^{(k+1)} = \frac{\bar{f}_j^{(k)}}{\sum_{i=1}^n a_{ij}} \sum_{i=1}^n \frac{g_i}{\sum_{j=1}^m a_{ij} \bar{f}_j^{(k)}} a_{ij} \quad \text{Eq. 9}$$

The first image $\bar{f}^{(0)}$ can be a uniform disk enclosed in the field of view (30) or an image obtained by FBP (in this case the negative values, if any, must be set to zero or some small positive value). Notice that because the first image is positive (negative values do not make sense) and because each new value is found by multiplying the current value by a positive factor, any $\bar{f}^{(k)}$ cannot have negative values and that any values set initially to zero will remain zero.

The EM algorithm can be seen as a set of successive projections/backprojections (31): The factor $g_i / (\sum_{j=1}^m a_{ij} \bar{f}_j^{(k)})$ is the ratio of the measured number of counts to the current estimate of the mean number of counts in bin i . $\sum_{i=1}^n (g_i / \sum_{j=1}^m a_{ij} \bar{f}_j^{(k)}) a_{ij}$ is the backprojection of this ratio for pixel j . Equation 9, which is to be applied pixel by pixel, can be extended to the whole image and interpreted as:

$$\text{Image}^{(k+1)} = \text{Image}^{(k)} \times \text{Normalized Backprojection of} \\ \left(\frac{\text{Measured projections}}{\text{Projections of image}^{(k)}} \right). \quad \text{Eq. 10}$$

In other words, at each iteration k , a current estimate of the image is available. Using a system model (which may include attenuation and blur), it is possible to simulate what projections of this current estimate should look like. The measured projections are then compared with simulated projections of the current estimate, and the ratio between these simulated and measured projections is used to modify the current estimate to produce an updated (and hopefully more accurate) estimate, which becomes iteration $k + 1$. This process is then repeated many times. The MLEM algorithm converges slowly and may require 50–200 iterations.

The ordered-subsets expectation maximization (OSEM) has been proposed by Hudson and Larkin (32) to accelerate

the reconstruction process using the MLEM algorithm. With this method, the set of projections is divided into subsets (or blocks). For example, if there were 64 projections (acquired at 64 angles about the patient), they might be divided as into 16 subsets as follows, each containing 4 images:

Subset 1: Projections 1, 17, 33, 49

Subset 2: Projections 2, 18, 34, 50

...

Subset 16: Projections 16, 32, 48, 64.

It is useful for each subset to contain projections equally distributed about the patient, to help promote convergence of the algorithm. The MLEM is then applied to each subset in turn, as a subiteration. The first full iteration is complete when all subsets have been processed. Use of 16 subsets in OSEM, as in this example, would accelerate convergence by nearly a factor of 16 compared with standard MLEM and thus considerably shorten the computing time needed for reconstruction.

Maximum A Posteriori (MAP) Algorithms

Actually, the reconstructed images obtained using the MLEM algorithm tend to become noisy as the number of iterations increases, because noisy reconstructed images may yield projections that are very close to the measured noisy projections. Thus, the criterion “estimated projections have to be as close as possible to measured projections” is not the best criterion to obtain subjectively high-quality images. A better criterion may be “(a) estimated projections have to be as close as possible to measured projections AND (b) the reconstructed images have to be not too noisy.”

The introduction of a prior knowledge as a constraint that may favor convergence of the EM process is called regularization. The prior, based on an assumption of what the true image is, is usually chosen to penalize the noisy images. The goal is now to find \bar{f} so that the requirements (a) and (b) above are simultaneously maximized (Appendix 5). The maximization leads to an iterative scheme called the one-step-late (OSL) algorithm, described by Green (31):

$$\bar{f}_j^{(k+1)} = \frac{\bar{f}_j^{(k)}}{\sum_{i=1}^n a_{ij} + \beta \frac{\partial}{\partial \bar{f}_j} U(\bar{f}_j^{(k)})} \sum_{i=1}^n \frac{g_i}{\sum_{j=1}^m a_{ij} \bar{f}_j^{(k)}} a_{ij}, \quad \text{Eq. 11}$$

where $(\partial/\partial \bar{f}_j)U(\bar{f}_j^{(k)})$ (the prior term) is the derivative of a function U , called an energy function, and chosen to enforce smoothing and β is a value chosen to modulate the importance of the prior. This modified EM algorithm is called the MAPEM algorithm using the OSL approach.

To understand how the MAPEM algorithm favors smooth images, we shall use as an example the quadratic prior (33). The energy function U in this case is defined as:

$$\frac{\partial}{\partial \bar{f}_j^{(k)}} U(\bar{f}_j^{(k)}) = \sum_{b \in N_j} w_{jb} (\bar{f}_j^{(k)} - \bar{f}_b^{(k)}), \quad \text{Eq. 12}$$

where N_j is a set of pixels in the neighborhood of pixel j .

Observe that the only difference between the MLEM and MAPEM algorithms resides in the additional term $\beta(\partial/\partial\bar{f}_j^{(k)}) \times U(\bar{f}_j^{(k)})$ in the denominator (compare Eqs. 9 and 11); if all pixels in the neighborhood of pixel j have the same value (i.e., the image in this neighborhood is smooth), then $\beta(\partial/\partial\bar{f}_j^{(k)})U(\bar{f}_j^{(k)}) = 0$ and the new value is the same as when the MLEM algorithm is used. If the value for pixel j is in average higher than its neighbors, then the derivative of the energy function is positive. Then, the denominator is larger than $\sum_{i=1}^n a_{ji}$, and the new value $\bar{f}_j^{(k+1)}$ is forced to be lower than it would be if the MLEM would be used. On the contrary, if the value for pixel j is in average lower than its neighbors, the denominator is lower than $\sum_{i=1}^n a_{ji}$, and the new value $\bar{f}_j^{(k+1)}$ is forced to be higher than it would be if the MLEM were used.

A first remark is that the denominator can be negative, and, in this case, the new pixel value may become negative, which does not make sense when this value is a number of photons. Negative values can be avoided by keeping the weight of the prior β low enough or using an algorithm that avoids the OSL trick—for example, the interior point algorithm of Byrne (34) or the algorithm using surrogate functions of De Pierro (35).

Second, the quadratic prior forces the image to be smooth on the edges too, and this leads to a blurring of the image. To overcome this problem, many other priors have been proposed. When information about the edges is not available (for example, from anatomic images), the basic assumption is that if the difference between pixel values in the potential function is very high, then one considers that it is likely that the pixel is on an edge, and in this case the smoothing term $\beta(\partial/\partial\bar{f}_j^{(k)})U(\bar{f}_j^{(k)})$ has to be as close to zero as possible. One way to keep the smoothing term low when the difference $\bar{f}_j - \bar{f}_b$ becomes too high is to use nonlinear functions as potential function, so that this term does not grow proportionally to the pixel differences (31,36–38). Another possibility is to compare the current pixel value with the median value of the neighborhood (39). If anatomic information is available (e.g., from CT or MRI), then a simple way to avoid edge smoothing is to consider for a neighborhood only the pixels that belong to the same anatomic area as the current pixel. However, because edges within scintigraphic and MRI or CT images may be different, more sophisticated approaches have been described (40–42).

CONCLUSION

Three algorithms are presented here: the FBP, CG, and MLEM algorithms. The first applications of FBP (43,44) and iterative algorithms (45) were both described in the 1960s, but for a long time, and despite the advantages of iterative algorithms, the FBP algorithm was preferred because it was computationally faster (46). However, in clinical practice, the FBP algorithm has several flaws: A smoothing filter is usually

required to reduce the noise, resulting in a loss of resolution. Also, an optimal filter has to be chosen (47,48) to provide the best trade-off between image noise and image resolution (although iterative algorithms may also require optimization—for example, by choosing the value of the β coefficient). The MAP algorithm can give better image quality compared with the FBP algorithm as shown in a simulation study (49). Moreover, the streak artifact observed when a region of the body is highly radioactive relative to the neighborhood (e.g., the bladder) has been shown to be intense with FBP but strongly reduced when OSEM is used in bone SPECT studies (50). Given the current computational power of modern computers, iterative reconstruction algorithms have become feasible for routine clinical use. Practical implementations of FBP, CG algorithm, and other algorithms can be found in the Donner Algorithms for Reconstruction Tomography. This package includes a detailed user's manual and program sources in Fortran and can be freely downloaded from the Web site http://cfi.lbl.gov/cfi_software.html. Also freely available is the Kansas University Image Processing (KUIM) package (<http://www.itc.ukans.edu/~jgauch/research/kuim/html/>), which provides C source for many image processing tools—in particular, projection and backprojection. Commercial softwares IDL (Research Systems, Inc., Boulder, CO) and Matlab (The MathWorks, Inc., Natick, MA) include projection and back-projection functions.

APPENDIX 1

The goal of the calculations below is to find in the field of view the set of points $M(x, y)$ that perpendicularly projects on D at P (Fig. 15). Using the usual trigonometric formulas, we have:

$$\begin{cases} x_l = s \cos \theta \\ y_l = s \sin \theta \end{cases} \quad \text{Eq. A1.1}$$

and

$$\begin{cases} x_l - x = u \sin \theta \\ y - y_l = u \cos \theta. \end{cases} \quad \text{Eq. A1.2}$$

Then, combining these 2 equations to eliminate x_l and y_l , we obtain:

$$\begin{cases} x = s \cos \theta - u \sin \theta \\ y = s \sin \theta + u \cos \theta \end{cases} \quad \text{Eq. A1.3}$$

or equivalently

$$\begin{cases} s = x \cos \theta + y \sin \theta \\ u = -x \sin \theta + y \cos \theta. \end{cases} \quad \text{Eq. A1.4}$$

The line D' defined by the equation $s = x \cos \theta + y \sin \theta$ perpendicularly projects on D at P. Thus, this line D' is the set of points $M(x, y)$ in the field of view whose photons can be detected at a distance s from the middle of the detector when the detector is at an angle θ .

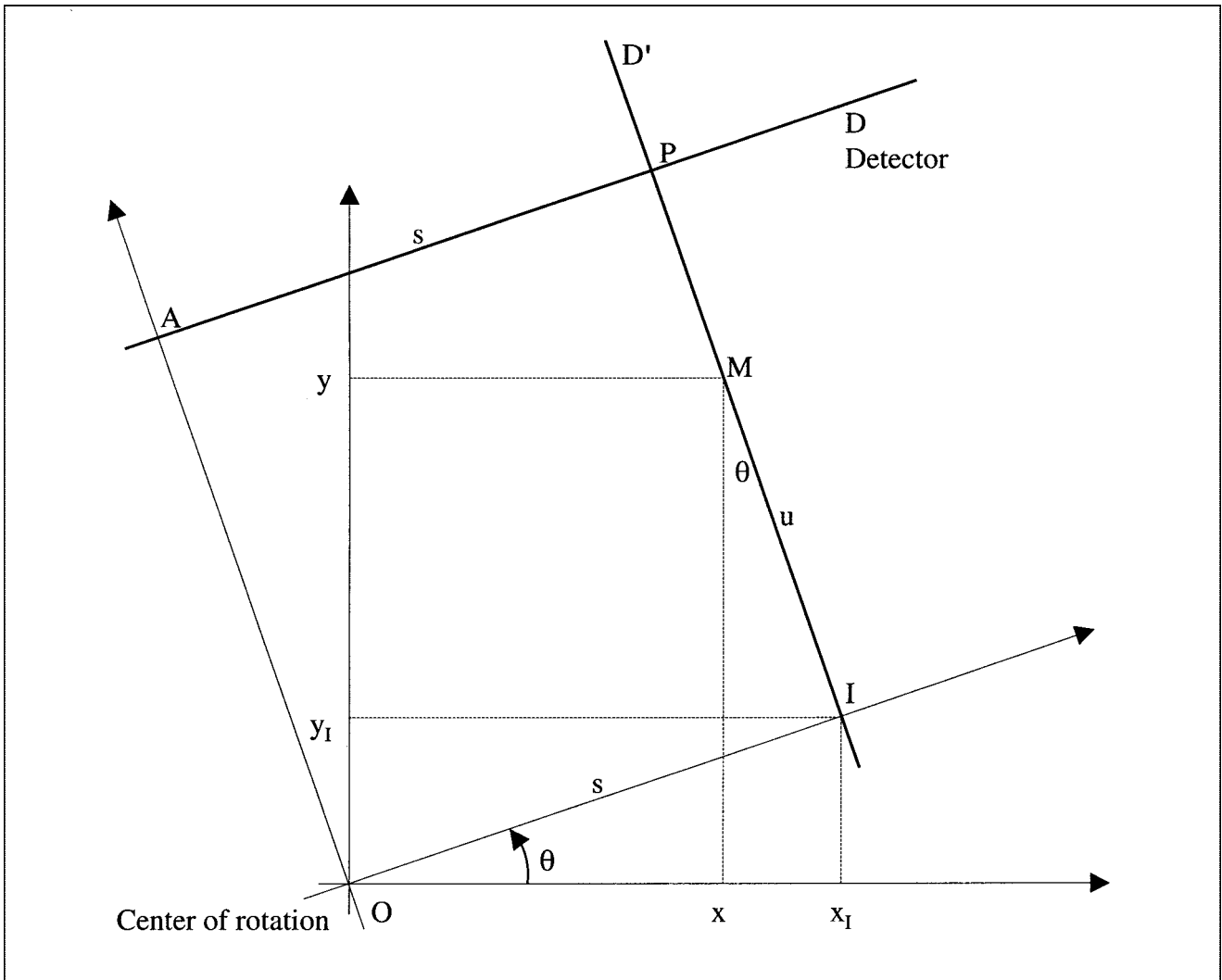


FIGURE 15. Geometric considerations. Point O is center of rotation of detector, and A is middle of detector line symbolized by line D . Angle θ marks angular position of detector. Line D' is set of points M in field of view that projects perpendicularly on D in P . Distance from I to M is u . Distance from A to P is s . Note that (s, θ) are not polar coordinates of M or P .

APPENDIX 2

Let us consider 2 matrices $B(n, m)$ (i.e., with n rows and m columns) and $C(m, l)$. The matrix product of B and C is the matrix D such that the value of any element d_{ij} (at the intersection of the i th row and j th column) of D is $d_{ij} = b_{i1}c_{1j} + b_{i2}c_{2j} + \dots + b_{im}c_{mj} = \sum_{k=1}^m b_{ik}c_{kj}$.

Example: if $B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$ and $C = \begin{bmatrix} 7 & 8 \\ 9 & 10 \end{bmatrix}$, then

$$D = BC = \begin{bmatrix} (1 \times 7) + (2 \times 9) & (1 \times 8) + (2 \times 10) \\ (3 \times 7) + (4 \times 9) & (3 \times 8) + (4 \times 10) \\ (5 \times 7) + (6 \times 9) & (5 \times 8) + (6 \times 10) \end{bmatrix}$$

$$= \begin{bmatrix} 25 & 28 \\ 57 & 64 \\ 89 & 100 \end{bmatrix}.$$

APPENDIX 3

In the example presented in Figure 3, we have the following relationships between values in the slice and in the sinogram:

$$\begin{cases} g_1 = & f_3 & + f_6 & + f_9 \\ g_2 = & f_2 & + f_5 & + f_8 \\ g_3 = f_1 & + f_4 & + f_7 \\ g_4 = f_1 + f_2 + f_3 & & & \\ g_5 = & f_4 + f_5 + f_6 & & \\ g_6 = & & f_7 + f_8 + f_9. \end{cases}$$

Eq. A3.1

This set of equations can be equivalently expressed as a matrix product:

$$\underbrace{\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \end{bmatrix}}_g = \underbrace{\begin{bmatrix} 001001001 \\ 010010010 \\ 100100100 \\ 111000000 \\ 000111000 \\ 000000111 \end{bmatrix}}_A \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix}}_f. \quad \text{Eq. A3.2}$$

The number of rows and columns of A equals the number of pixels and bins, respectively. The values in matrix A are appropriately chosen so that, using the definition of matrix product (Appendix 2), it can be verified that the product of A by f yields $g_1, g_2 \dots g_6$ as presented in Equation A3.1.

APPENDIX 4

The following is a simplified description of the MLEM algorithm as presented by Shepp and Vardi (23) and by Lange and Carson (24). In these works, the numbers of both the emitted and the detected disintegrations are assumed to be Poisson random variables.

Let us consider \bar{f}_j , the mean number of disintegrations in pixel j , and the element a_{ij} of the matrix A , the probability that bin i detects a photon emitted from pixel j . The mean number of photons emitted from pixel j and detected by bin i is $a_{ij}\bar{f}_j$. The mean number of photons \bar{g}_i detected by bin i is the sum of the mean numbers of photons emitted from each pixel:

$$\bar{g}_i = \sum_{j=1}^m a_{ij}\bar{f}_j. \quad \text{Eq. A4.1}$$

It can be shown that the number of photons emitted from the m pixels and detected by bin i is a Poisson variable. Thus, the probability of detecting g_i photons is given by:

$$P(g_i) = \frac{e^{-\bar{g}_i} \bar{g}_i^{g_i}}{g_i!}, \quad \text{Eq. A4.2}$$

where ! denotes the factorial operator defined for any positive integer n as $n! = n \times (n - 1) \times \dots \times 1$, and $0! = 1$. For example, if the mean number of photons detected in bin i is $\bar{g}_i = 3$, then the probability to detect zero, 1, or 5 photons is, respectively, $P(0) = (e^{-3}3^0)/0! \approx 0.050$, $P(1) = (e^{-3}3^1)/1! \approx 0.149$, and $P(5) = (e^{-3}3^5)/5! \approx 0.101$. The maximal probability is reached when the number of counts is equal to the mean number $P(g_i = \bar{g}_i)$; here $P(3) = (e^{-3}3^3)/3! \approx 0.224$.

The i Poisson variables are independent, and the conditional probability $P(g|\bar{f})$ of observing the vector g when the emission map is f is the product of the individual

probabilities $P(g_i)$. The likelihood function $L(\bar{f})$ is given by:

$$\begin{aligned} L(\bar{f}) &= P(g|\bar{f}) \\ &= P(g_1)P(g_2) \dots P(g_n) \\ &= \prod_{i=1}^n P(g_i) \\ &= \prod_{i=1}^n \frac{e^{-\bar{g}_i} \bar{g}_i^{g_i}}{g_i!}. \end{aligned} \quad \text{Eq. A4.3}$$

The highest value for the likelihood $L(\bar{f})$ is found by computing its derivative. To maximize the expectation, one usually considers $l(f) = \ln(L(f))$, where \ln denotes the natural logarithm. Remembering that for any non-null, positive real values $x_1, x_2 \dots x_n$ we have:

- (1) $\ln(x_1 x_2 \dots x_n) = \ln(x_1) + \ln(x_2) + \dots + \ln(x_n)$,
- (2) $\ln(e^{x_1}) = x_1$,
- (3) $\ln(1/x_1) = -\ln(x_1)$,

Equation A4.3 becomes:

$$l(\bar{f}) = \sum_{i=1}^n (-\bar{g}_i + g_i \ln(\bar{g}_i) - \ln(g_i!)), \quad \text{Eq. A4.4}$$

and using Equation A4.1 to introduce \bar{f}_j , we obtain:

$$l(\bar{f}) = \sum_{i=1}^n \left(-\sum_{j=1}^m a_{ij}\bar{f}_j + g_i \ln \left(\sum_{j=1}^m a_{ij}\bar{f}_j \right) - \ln(g_i!) \right). \quad \text{Eq. A4.5}$$

This equation, called the likelihood function, is of fundamental importance in the MLEM algorithm, because it allows one to calculate the probability to observe a projection dataset for any mean image \bar{f} . Obviously, we want the image that has the highest probability to yield g , so the essential point here is that the vector \bar{f} for which $l(\bar{f})$ is maximal is considered as the best estimate for the solution.

It can be shown that $l(\bar{f})$ has 1 and only 1 maximum (26). This maximum is found when the derivative of $l(\bar{f})$ is zero:

$$\frac{\partial l(\bar{f})}{\partial \bar{f}_j} = -\sum_{i=1}^n a_{ij} + \sum_{i=1}^n \frac{g_i}{\sum_{j'=1}^m a_{ij'}\bar{f}_{j'}} a_{ij} = 0. \quad \text{Eq. A4.6}$$

One can also write:

$$\bar{f}_j \frac{\partial l(\bar{f})}{\partial \bar{f}_j} = -\bar{f}_j \sum_{i=1}^n a_{ij} + \bar{f}_j \sum_{i=1}^n \frac{g_i}{\sum_{j'=1}^m a_{ij'}\bar{f}_{j'}} a_{ij} = 0, \quad \text{Eq. A4.7}$$

that is

$$\bar{f}_j = \frac{\bar{f}_j}{\sum_{i=1}^n a_{ij}} \sum_{i=1}^n \frac{g_i}{\sum_{j'=1}^m a_{ij}\bar{f}_{j'}} a_{ij}. \quad \text{Eq. A4.8}$$

This leads to the iterative scheme of the MLEM algorithm as described by Lange and Carson (24):

$$\bar{f}_j^{(k+1)} = \frac{\bar{f}_j^{(k)}}{\sum_{i=1}^n a_{ij}} \sum_{i=1}^n \frac{g_i}{\sum_{j'=1}^m a_{ij}\bar{f}_{j'}^{(k)}} a_{ij}. \quad \text{Eq. A4.9}$$

APPENDIX 5

The prior knowledge of what the reconstructed image should be is introduced in the EM algorithm using Bayes' theorem. It states that:

$$P(\bar{f}|g) = \frac{P(g|\bar{f})P(\bar{f})}{P(g)}, \quad \text{Eq. A5.1}$$

where $P(g|\bar{f})$ is the likelihood function as defined in Equation A4.3. $P(\bar{f})$, the prior function, defines the a priori knowledge of the image; $P(g)$ is the a priori probability distribution of the measurements; $P(\bar{f}|g)$ is the a posteriori probability distribution of the image.

Taking the logarithm of both sides of Equation A5.1 yields:

$$\ln P(\bar{f}|g) = \ln P(g|\bar{f}) + \ln P(\bar{f}) - \ln P(g). \quad \text{Eq. A5.2}$$

A common Bayesian prior to enforce local smoothness is the Gibbs prior, with $P(\bar{f}) = Ce^{-\beta U(\bar{f})}$, where U is an energy function of \bar{f} , β is the weight of the prior, and C is a normalizing constant. Using Equation A4.1, we obtain:

$$\ln P(\bar{f}|g) = \sum_{i=1}^n \left(-\sum_{j=1}^m a_{ij}\bar{f}_j + g_i \ln \left(\sum_{j=1}^m a_{ij}\bar{f}_j \right) - \ln(g_i!) \right) - \beta U(\bar{f}) + K, \quad \text{Eq. A5.3}$$

where $K = \ln C - \ln P(g)$ is a constant independent of \bar{f} .

Again, the derivative of $P(\bar{f}|g)$ is used to find the image \bar{f} maximizing $P(\bar{f}|g)$:

$$\frac{\partial \ln P(\bar{f}|g)}{\partial \bar{f}_j} = -\sum_{i=1}^n a_{ij} + \sum_{i=1}^n \frac{g_i}{\sum_{j'=1}^m a_{ij}\bar{f}_{j'}} a_{ij} - \beta \frac{\partial}{\partial \bar{f}_j} U(\bar{f}_j). \quad \text{Eq. A5.4}$$

The OSL algorithm uses the current image when calculating the energy. With this simplification, $U(\bar{f}_j)$ is replaced by $U(\bar{f}_j^{(k)})$ in Equation A5.4.

ACKNOWLEDGMENTS

The author is grateful to Jacques Sau for the invaluable discussions about this manuscript. The author also thanks Michael A. King and Stephen J. Glick for their assistance in improving this manuscript.

REFERENCES

- Brooks RA, Di Chiro G. Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging. *Phys Med Biol.* 1976;21:689–732.
- Kay DB, Keyes JW Jr, Simon W. Radionuclide tomographic image reconstruction using Fourier transform techniques. *J Nucl Med.* 1974;15:981–986.
- Kak AC, Slaney M. Algorithms for reconstruction with non-diffracting sources. In: Cotellessa R, ed. *Principles of Computerized Tomographic Imaging.* New York, NY: IEEP Press; 1988:49–112.
- Edholm PR, Herman GT. Linograms in image reconstruction from projections. *IEEE Trans Med Imaging.* 1987;MI-6:301–307.
- Edholm PR, Herman GT, Roberts DA. Image reconstruction from linograms: implementation and evaluation. *IEEE Trans Med Imaging.* 1988;7:239–246.
- Bortfeld T, Oelfke U. Fast and exact 2D image reconstruction by means of Chebyshev decomposition and backprojection. *Phys Med Biol.* 1999;44:1105–1120.
- Andia BI, Sauer KD, Bouman CA. Nonlinear backprojection for tomographic reconstruction. *IEEE Trans Nucl Sci.* 2002;49:61–68.
- Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. *J Opt Soc Am.* 1984;1:612–619.
- Perter TM, Lewitt RM. Computed tomography with fan beam geometry. *J Comput Assist Tomogr.* 1977;1:429–436.
- Zeng GL. Image reconstruction: a tutorial. *Comput Med Imaging Graph.* 2001; 25:97–103.
- Vandenberghe S, D'Asseler Y, Van de Walle R, et al. Iterative reconstruction algorithms in nuclear medicine. *Comput Med Imaging Graph.* 2001;25:105–111.
- Rowland SW. Computer implementation of image reconstruction formulas. In: Herman GT, ed. *Topics in Applied Physics: Image Reconstruction from Projections.* Vol. 32. Heidelberg, Germany: Springer-Verlag Heidelberg; 1979:29–79.
- Jain AK. Image reconstruction from projections. In: Thomas Kailath, ed. *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall; 1989:431–475.
- Radon J. On the determination of functions from their integrals along certain manifolds [in German]. *Math Phys Klass.* 1917;69:262–277.
- Bryant PP, Sau J, Mallet JJ. Streak artifact reduction in filtered backprojection using a level line-based interpolation method. *J Nucl Med.* 2000;41:1913–1919.
- Siddon RL. Fast calculation of the exact radiological path for a three-dimensional CT array. *Med Phys.* 1985;12:252–255.
- Wallis JW, Miller TR. An optimal rotator for iterative reconstruction. *IEEE Trans Med Imaging.* 1997;16:118–123.
- Budinger TF, Gullberg GT, Huesman RH. Emission computed tomography. In: Herman GT, ed. *Topics in Applied Physics: Image Reconstruction from Projections.* Vol. 32. Heidelberg, Germany: Springer-Verlag Heidelberg; 1979:147–246.
- Van Laere K, Koole M, Lemahieu I, Dierckx R. Image filtering in single-photon emission computed tomography: principles and applications. *Comput Med Imaging Graph.* 2001;25:127–133.
- Germano G. Technical aspects of myocardial SPECT imaging. *J Nucl Med.* 2001;42:1499–1507.
- Macovski A. Tomography. In: Kailath T, ed. *Medical Imaging Systems.* Englewood Cliffs, NJ: Prentice-Hall; 1997:106–144.
- Press WH, Teukolski SA, Vetterling WT, Flannery BP. Minimization or maximization of functions. In: Press W, ed. *Numerical Recipes in C.* Cambridge, MA: Cambridge University Press; 1996:394–455.
- Shepp LA, Vardi Y. Maximum likelihood reconstruction for emission tomography. *IEEE Trans Med Imaging.* 1982;MI-1:113–122.
- Lange K, Carson R. EM reconstruction algorithms for emission and transmission tomography. *J Comput Assist Tomogr.* 1984;8:306–316.
- Dempster A, Laird N, Rubin D. Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc [B].* 1977;39:1–38.
- Kaufman L. Implementing and accelerating the EM algorithm for positron emission tomography. *IEEE Trans Med Imaging.* 1987;MI-6:37–51.
- Kaufman L. Maximum likelihood, least squares and penalized least squares for PET. *IEEE Trans Med Imaging.* 1993;12:200–214.
- Clinthorne NH, Pan T-S, Chiao P-C, Rogers WL, Stamos JA. Preconditioning

- methods for improved convergence rates in iterative reconstructions. *IEEE Trans Med Imaging*. 1993;12:78–83.
29. Chinn G, Huang SC. A general class of preconditioners for statistical iterative reconstruction of emission computed tomography. *IEEE Trans Med Imaging*. 1997;16:1–10.
 30. Llacer J, Veklerov E, Coakley K, Hoffman E, Nunez J. Statistical analysis of maximum likelihood estimator images of human brain FDG PET studies. *IEEE Trans Med Imaging*. 1993;12:215–231.
 31. Green PJ. Bayesian reconstructions from emission tomography data using a modified EM algorithm. *IEEE Trans Med Imaging*. 1990;9:84–93.
 32. Hudson HM, Larkin RS. Accelerated image reconstruction using ordered subsets of projection data. *IEEE Trans Med Imaging*. 1994;13:601–609.
 33. Geman S, Geman D. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans Pattern Anal Machine Intell*. 1984;6:721–741.
 34. Byrne CL. Block-iterative interior point optimization methods for image reconstruction from limited data. *Inverse Problems*. 2000;16:1405–1419.
 35. De Pierro AR. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE Trans Med Imaging*. 1995;14:132–137.
 36. Geman S, MacClure D. Bayesian image analysis: an application to single photon emission tomography. In: *Proceedings of the Statistical Computing Section, American Statistical Association*. Alexandria, VA: American Statistical Association; 1985;12–18.
 37. Hebert T, Leahy R. A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors. *IEEE Trans Med Imaging*. 1989;8:194–202.
 38. Lalush DS, Tsui BMW. Simulation evaluation of Gibbs prior distributions for use in maximum a posteriori SPECT reconstructions. *IEEE Trans Med Imaging*. 1992;11:267–275.
 39. Alenius S, Ruotsalainen U. Bayesian image reconstruction for emission tomography based on median root prior. *Eur J Nucl Med*. 1997;24:258–265.
 40. Ardekani BA, Braun M, Hutton BF, Kanno I, Iida H. Minimum cross-entropy reconstruction of PET images using prior anatomical information. *Phys Med Biol*. 1996;41:2497–2517.
 41. Gindi G, Rangarajan A, Zubal G. Bayesian reconstruction of functional images using anatomical information as priors. *IEEE Trans Med Imaging*. 1993;12:670–680.
 42. Bowsler JE, Johnson VE, Turkington TG, Jaszczak RJ, Floyd CE Jr, Coleman RE. Bayesian reconstruction and use of anatomical a priori information for emission tomography. *IEEE Trans Med Imaging*. 1996;15:673–686.
 43. Cormack AM. Representation of a function by its line integrals with some radiological applications. *J Appl Phys*. 1963;34:2722–2727.
 44. Cormack AM. Representation of a function by its line integrals with some radiological applications. II. *J Appl Phys*. 1964;35:2908–2913.
 45. Kuhl DE, Edwards RQ. Image separation radioisotope scanning. *Radiology*. 1963;80:653–662.
 46. Hutton BF, Hudson HM, Beekman FJ. A clinical perspective of accelerated statistical reconstruction. *Eur J Nucl Med*. 1997;24:797–808.
 47. King MA, Glick SJ, Penney BC, Schwinger RB, Doherty PW. Interactive visual optimization of SPECT prereconstruction filtering. *J Nucl Med*. 1987;28:1192–1198.
 48. Gilland DR, Tsui BMW, MacCartney WH, Perry JR, Berg J. Determination of the optimum filter function for SPECT imaging. *J Nucl Med*. 1988;29:643–650.
 49. Wang CX, Snyder WE, Bilbro G, Santago P. Performance evaluation of filtered backprojection reconstruction and iterative reconstruction methods for PET images. *Comput Biol Med*. 1998;28:13–24.
 50. Blocklet D, Seret A, Popa N, Schoutens A. Maximum-likelihood reconstruction with ordered subsets in bone SPECT. *J Nucl Med*. 1999;40:1978–1984.

