Chapter 13

Image reconstruction

Johan Nuyts and Samuel Matej

Draft - February 25, 2015, 18h 4min

13.1 Introduction

This chapter discusses how 2D or 3D images of the tracer distribution can be reconstructed from a series of so-called projection images acquired with a gamma camera or a PET system [1]. This is often called an "inverse problem". The reconstruction is the inverse of the acquisition. The reconstruction is called an "inverse problem", because making software to compute the true tracer distribution from the acquired data turns out to be more difficult than the "forward" direction, i.e. making software to simulate the acquisition.

There are basically two approaches to image reconstruction: analytical reconstruction and iterative reconstruction. The analytical approach is based on mathematical inversion, yielding efficient, non-iterative reconstruction algorithms. In the iterative approach the reconstruction problem is reduced to computing a finite number of image values from a finite number of measurements. That simplification enables the use of iterative instead of mathematical inversion. Iterative inversion tends to require more computer power, but it can cope with more complex (and hopefully more accurate) models of the acquisition process.

13.2 Analytical reconstruction

The (n-dimensional) Radon transform maps an image of dimension n to the set of all integrals over hyperplanes of dimension $n-1$ [2]. Thus, in two dimensions, the Radon transform of image Λ corresponds to all possible line integrals of Λ. In three dimensions, the Radon transform contains all possible plane integrals.

The (n-dimensional) X-ray transform maps an image of dimension n to the set of all possible line integrals. In all PET and in almost all SPECT applications, the measured projections can be well approximated as a subset of the (possibly attenuated) X-ray transform, because the mechanical (SPECT) or electronic (PET) collimation is designed to acquire information along lines (the LOR or line-of-response, see chapter 11). Consequently, reconstruction involves computing the unknown image Λ from (part of) its X-ray transform. Figure 13.1 shows PET projections, which are often represented as a set of projections or a set of sinograms.

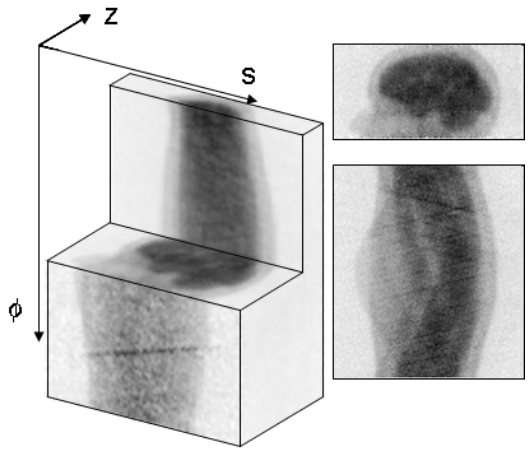

Figure 13.1: *The relation between projections and sinograms in parallel beam projection. The parallel beam (PET) acquisition is shown as a block with dimensions s, ϕ and z. A cross section at fixed ϕ yields a projection, a cross section at fixed z yields a sinogram.*

An important theorem for analytical reconstruction is the central slice (or central section) theorem, which gives a relation between the Fourier transform of an image and the Fourier transforms of its parallel projections. Below, the central slice theorem for 2D is found as eq (13.7), the 3D central section theorem as eq (13.29).

The direct Fourier method is a straightforward application of the central section theorem: it computes the Fourier transform of the projections, uses the central section theorem to obtain the Fourier transform of the image, and applies the inverse Fourier transform to obtain the image. In practice, this method is rarely used; the closely related filtered backprojection algorithm is far more popular.

13.2.1 2D tomography

13.2.1.1 X-ray transform: projection and backprojection

In 2D, the Radon transform and X-ray transform are identical. Mathematically, the 2D X-ray (or Radon) transform of the image Λ can be written as follows:

$$\begin{aligned} Y(s,\phi) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Lambda(x,y)\ \delta_{s\,=\,x\cos\phi+y\sin\phi}\ dx\,dy \\ &= \int_{-\infty}^{\infty} \Lambda(s\cos\phi - t\sin\phi, s\sin\phi + t\cos\phi)dt, \end{aligned} \tag{13.1}$$

where the Dirac delta function δ equals zero except for the points on the LOR(s,ϕ). Note that with the notation used here, $\phi = 0$ corresponds to projection along the y-axis .

The Radon transform describes the acquisition process in 2D PET and in SPECT with parallel hole collimation, if we can ignore attenuation. Assuming that $\Lambda(x,y)$ represents the tracer distribution at transaxial slice z through the patient, then $Y(s,\phi)$ represents the corresponding sinogram, and contains the z-th row of the projections acquired at angles ϕ. Fig 13.1 illustrates the relation between the projection and the sinogram.

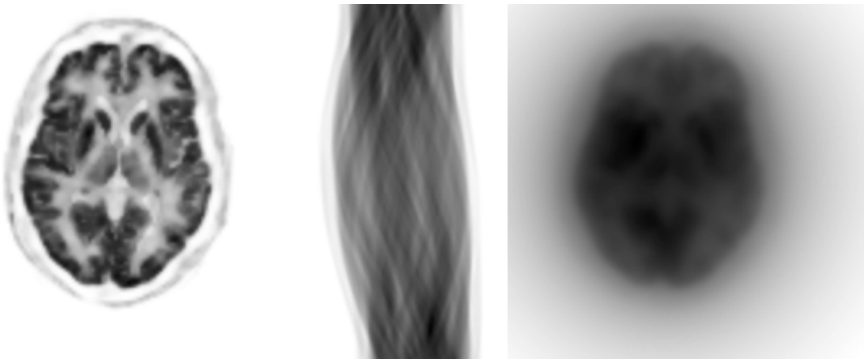

Figure 13.2: *The image (left) is projected to produce a sinogram (centre), which in turn is backprojected, yielding a smoothed version of the original image.*

The X-ray transform has an adjoint operation that appears in both analytical and iterative reconstruction. This operator is usually called the backprojection operator, and can be written as:

$$\begin{aligned} B(x,y) &= \text{Backproj}\,(Y(s,\phi)) \\ &= \int_0^\pi d\phi \int_{-\infty}^{\infty} Y(s,\phi)\delta_{s\,=\,x\cos\phi+y\sin\phi}\,ds \\ &= \int_0^\pi Y(x\cos\phi + y\sin\phi,\phi)\,d\phi. \end{aligned} \tag{13.2}$$

The backprojection is not the inverse of the projection, $B(x,y) \neq \Lambda(x,y)$. Intuitively, the backprojection sends the measured activity back into the image by distributing it uniformly along the projection lines. As illustrated in figure 13.2, projection followed by backprojection produces a blurred version of the original image. This blurring corresponds to the convolution of the original image with the 2D convolution kernel $1/\sqrt{x^2+y^2}$.

13.2.1.2 Central slice theorem

The central slice theorem gives a very useful relation between the 2D Fourier transform of the image and the 1D Fourier transform of its projections (along the detector axis). Consider the projection along the y-axis, $\phi = 0$, and its 1D Fourier transform:

$$Y(s,0) = \int_{-\infty}^{\infty} \Lambda(s,t)dt \tag{13.3}$$

$$\begin{aligned} (\mathcal{F}_1 Y)(\nu_s,0) &= \int_{-\infty}^{\infty} Y(s,0)e^{-i2\pi\nu_s s}ds \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Lambda(s,t)e^{-i2\pi\nu_s s}dt\,ds, \end{aligned} \tag{13.4}$$

and compare this to the 2D Fourier transform of the image $\Lambda(x,y)$:

$$(\mathcal{F}_2\Lambda)(\nu_x,\nu_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Lambda(x,y)e^{-i2\pi(\nu_x x+\nu_y y)}dx\,dy. \tag{13.5}$$

Both expressions are equal if we set $\nu_y = 0$:

$$(\mathcal{F}_1 Y)(\nu_s,0) = (\mathcal{F}_2\Lambda)(\nu_x,0) \tag{13.6}$$

$(\mathcal{F}_1 Y)(\nu_s, 0)$ is the 1D Fourier transform of the projection along the y-axis, $(\mathcal{F}_2\Lambda)(\nu_x, 0)$ is a "central slice" along the ν_x-axis through the 2D Fourier transform of the image. Eq (13.6) is the central slice theorem for the special case of projection along the y-axis. This result would still hold if we had rotated the object or equivalently, the x and y axes. Consequently, it holds for any angle ϕ:

$$(\mathcal{F}_1 Y)(\nu_s, \phi) = (\mathcal{F}_2\Lambda)(\nu_s \cos\phi, \nu_s \sin\phi). \tag{13.7}$$

13.2.1.3 2D filtered backprojection

The central slice theorem (13.7) can be directly applied to reconstruct an unknown image $\Lambda(x, y)$ from its known projections $Y(s, \phi)$. The 1D Fourier transform of the projections provides all possible central slices through $(\mathcal{F}_2\Lambda)(\nu_x, \nu_y)$, if $Y(s, \phi)$ is known for all ϕ in an interval with a length of at least π (Tuy's condition). Consequently, $(\mathcal{F}_2\Lambda)(\nu_x, \nu_y)$ can be constructed from the 1D Fourier transform of $Y(s, \phi)$. Inverse 2D Fourier transform then provides $\Lambda(x, y)$.

However, a basic Fourier method implementation with a simple interpolation in Fourier space does not work well. By contrast, in the case of the filtered backprojection algorithm (FBP) derived below, a basic real-space implementation with a simple convolution and a simple interpolation in the backprojection works well. Inverse Fourier transform of (13.5) yields

$$\Lambda(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (\mathcal{F}_2\Lambda)(\nu_x, \nu_y) e^{i2\pi(\nu_x x + \nu_y y)} d\nu_x \, d\nu_y. \tag{13.8}$$

This can be rewritten with polar coordinates as

$$\Lambda(x, y) = \int_{-\infty}^{\infty} d\nu \int_0^{\pi} (\mathcal{F}_2\Lambda)(\nu\cos\phi, \nu\sin\phi) e^{i2\pi(x\nu\cos\phi + y\nu\sin\phi)} |\nu| d\phi. \tag{13.9}$$

Application of the central slice theorem (13.7) and reversing the order of integration finally results in:

$$\Lambda(x, y) = \int_0^{\pi} d\phi \int_{-\infty}^{\infty} (\mathcal{F}_1 Y)(\nu, \phi) |\nu| e^{i2\pi\nu(x\cos\phi + y\sin\phi)} d\nu, \tag{13.10}$$

which is the FBP algorithm. This algorithm involves the following steps:

1. apply 1D Fourier transform to $Y(s, \phi)$ to obtain $(\mathcal{F}_1 Y)(\nu, \phi)$;
2. filter $(\mathcal{F}_1 Y)(\nu, \phi)$ with the so called ramp filter $|\nu|$
3. apply the 1D inverse Fourier transform to obtain the ramp filtered projections $\hat{Y}(s, \phi) = \int (\mathcal{F}_1\Lambda)(\nu, \phi)|\nu| e^{i2\pi\nu s} d$
4. apply the backprojection operator (13.2) to $\hat{Y}(s, \phi)$ to obtain the desired image $\Lambda(x, y)$.

Note that the ramp filter sets the DC component of the image to zero, while the mean value of the reconstructed image should definitely be positive. As a result, straightforward discretisation of FBP causes significant negative bias. The problem is reduced with "zero padding" before computing the Fourier transform with FFT. Zero padding involves extending the sinogram rows with zeros at both sides. This increases the sampling in the frequency domain and results in a better discrete approximation of the ramp filter. However, a huge amount of zero padding is required to effectively eliminate the bias completely. The next paragraph shows how this need for zero padding can be easily avoided. Note that after inverse Fourier transform the extended region may be discarded, so the size of the filtered sinogram remains unchanged.

Instead of filtering in the Fourier domain, one can also implement the ramp filtering as a 1D convolution in the spatial domain. For that purpose, one needs the inverse Fourier transform of $|\nu|$. This inverse transform actually does not exist, but approximating it as the limit for $\epsilon \to 0$ of the well behaved function $|\nu|e^{-\epsilon|\nu|}$, one obtains [3, 4]:

$$\mathcal{F}^{-1}\left(|\nu|e^{-\epsilon|\nu|}\right) = \frac{\epsilon^2 - (2\pi s)^2}{(\epsilon^2 + (2\pi s)^2)^2} \tag{13.11}$$

$$\simeq -\frac{1}{(2\pi s)^2} \text{ for } |s| >> \epsilon. \tag{13.12}$$

In practice, one always works with band limited functions, implying that the ramp filter has to be truncated at the frequencies $\nu = \pm 1/(2\tau)$, where τ represents the sampling distance. The corresponding convolution kernel h then equals [3]

$$\begin{aligned} h(s) = \mathcal{F}^{-1}\left(|\nu|\, b(\nu)\right) &= \frac{1}{2\tau^2}\frac{\sin(\pi s/\tau)}{\pi s/\tau} - \frac{1}{4\tau^2}\left(\frac{\sin(\pi s/(2\tau))}{\pi s/(2\tau)}\right)^2 \\ \text{with } b(\nu) &= 1 \text{ if } |\nu| \le 1/(2\tau) \\ &= 0 \text{ if } |\nu| > 1/(2\tau). \end{aligned} \tag{13.13}$$

The kernel is normally only needed for samples $s = n\tau$: $h(n\tau) = 1/(4\tau^2)$ if $n = 0$, $h(n\tau) = 0$ if n is even and $h(n\tau) = -1/(n\pi\tau)^2$ if n is odd. One can implement the filter either as a convolution, or use the Fourier transform to obtain a digital version of the ramp filter. Interestingly, this way of computing the ramp filter also reduces the negative bias mentioned above. The reason is that this approach yields a non zero value for the DC-component [3]. When the filtering is done in the frequency domain, some zero padding before FFT is still recommended because of the circular convolution effects, but far less is needed than with straightforward discretization of $|\nu|$.

Although this is not obvious from the equations above, an algorithm equivalent to FBP is obtained by first backprojecting $Y(s, \phi)$, and then applying a 2D ramp filter to the backprojected image $B(x, y)$ [4]:

$$B(x, y) = \int_0^\pi Y(x\cos\phi + y\sin\phi, \phi)d\phi \tag{13.14}$$

$$(\mathcal{F}_2\Lambda)(\nu_x, \nu_y) = \sqrt{v_x^2 + v_y^2}\,(\mathcal{F}_2 B)(\nu_x, \nu_y). \tag{13.15}$$

This algorithm is often referred to as the backproject-then-filter algorithm.

Filtered backprojection assumes that the projections $Y(s, \phi)$ are line integrals. As discussed in chapter 11, PET and SPECT data are not line integrals because of attenuation, detector non-uniformities, the contribution of scattered photons and/or random coincidences etc. It follows that one has to recover (good estimates of) the line integrals by precorrecting the data for these effects. However, a particular problem is posed by the attenuation in SPECT because, different from PET, the attenuation depends on the position along the projection line, precluding straightforward precorrection. A detailed discussion of this problem is beyond the scope of this contribution, but three solutions are briefly mentioned here:

1) If one can assume that the attenuation is constant inside a convex body contour, then filtered backprojection can be modified to correct for the attenuation. Algorithms have been proposed by Bellini, by Tretiak and Metz and by others, an algorithm is presented in [3].

2) If the attenuation is not constant, then an approximate correction algorithm proposed by Chang

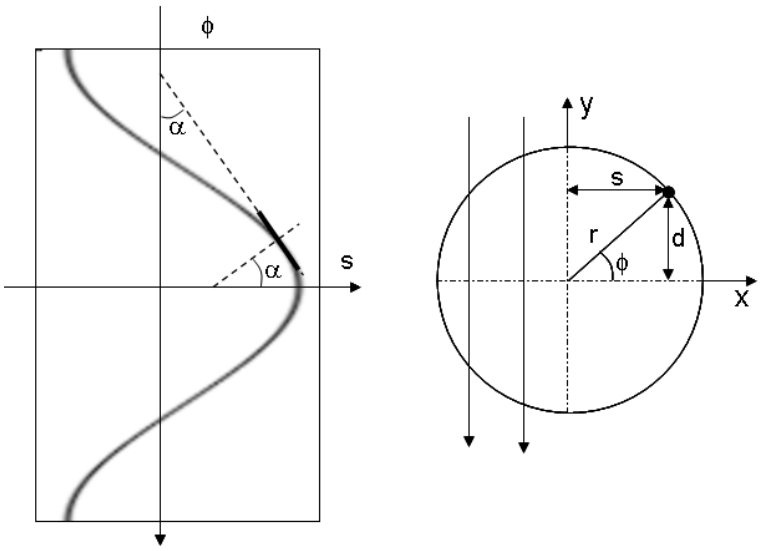

Figure 13.3: *The frequency distance principle. Left: sinogram. Right: vertical projection of a point located at polar coordinates* (r, ϕ).

can be applied [5]. It is a post-correction method, applied to the image obtained without any attenuation correction. To improve the approximation, one can compute the attenuated projection of the first reconstruction, and apply the method again to the difference of the measurement and this computed projection.
3) Finally, a modified filtered backprojection algorithm, compensating for non-uniform attenuation in SPECT, has been found by Novikov in 2000. An equivalent algorithm was derived by Natterer [6]. However, because this algorithm was only found after the successful introduction of iterative reconstruction in clinical practice, it has not received much attention in the nuclear medicine community.

13.2.2 Frequency-distance relation

Several very interesting methods in image reconstruction, including Fourier rebinning, are based on the so-called frequency-distance relation, proposed by PR Edholm, RM Lewitt and B Lindholm and described in detail in [7]. This is an approximate relation between the orthogonal distance to the detector and the direction of the frequency in the sinogram. The relation can be intuitively understood as follows. Consider the PET-acquisition of a point source, as illustrated in fig 13.3. Usually, the acquisition is described by rotating the projection lines while keeping the object stationary. However, here we consider the equivalent description, where projection is always along the y-axis, and tomographic acquisition is obtained by rotating the object around the origin. Suppose that the point is located on the x-axis when $\phi = 0$. When acquiring the parallel projections for angle ϕ, the point has polar coordinates (r, ϕ), with r the distance from the center of the field of view, and ϕ the angle with the x-axis. The distance to the x-axis equals $d = r \sin \phi$. The corresponding sinogram $Y(s, \phi)$ is zero everywhere, except on the curve $s = r \cos \phi$ (fig 13.3). The complete sinogram is obtained by rotating the point over 360^{o}: $\phi = -\pi \ldots \pi$. Consider a small portion of this curve, which can be well approximated as a tangential line segment near a particular point (s, ϕ), as illustrated in fig 13.3. In the 2D Fourier transform of the sinogram, this line segment contributes mostly frequencies in the

direction orthogonal to the line segment. This direction is represented by the angle α, given by

$$\tan\alpha = \frac{\partial}{\partial\phi}(r\cos\phi) = -r\sin\phi = -d. \tag{13.16}$$

Thus, in the 2D Fourier transform $(\mathcal{F}g)(\nu_s, \nu_\phi)$, the value at a particular point (ν_s, ν_ϕ) carries mostly information about points located at a distance $d = -\tan\alpha = -\nu_\phi/\nu_s$ from the line through the center, parallel to the detector. This relation can be exploited to apply distance dependent operations to the sinogram. One example is distance dependent deconvolution, to compensate for the distance dependent blurring in SPECT. Another example is Fourier rebinning, where data from oblique sinograms are rebinned into direct sinograms.

13.2.3 Fully 3D tomography

13.2.3.1 3D Filtered-backprojection

Due to the use of electronic collimation, the PET-scanner can simultaneously acquire information in a four-dimensional space of line integrals. These are the so-called lines-of-response (LOR), where each pair of detectors in coincidence defines a single LOR. In this section, the discrete nature of the detection is ignored, since the analytical approach is more easily described assuming continuous data. Consider the x-ray transform in 3D, which can be written as

$$Y(\hat{\mathbf{u}}, \mathbf{s}) = \int_{-\infty}^{\infty} dt \Lambda(\mathbf{s} + t\hat{\mathbf{u}}), \tag{13.17}$$

where the LOR is defined as the line parallel to $\hat{\mathbf{u}}$ and through the point $\mathbf{s}$. The vector $\hat{\mathbf{u}}$ is a unit vector, and the vector $\mathbf{s}$ is restricted to the plane orthogonal to $\hat{\mathbf{u}}$, hence $(\hat{\mathbf{u}}, \mathbf{s})$ is four dimensional. Most PET systems are either constructed as a cylindrical array of detectors, or as a rotating set of planar detector arrays, and therefore have a cylindrical symmetry. For that reason, the inversion of (13.17) is studied for the case where $\hat{\mathbf{u}}$ is restricted to the band Ω_{θ_0} on the unit sphere, defined by $|u_z| \leq \sin\theta_0$, as illustrated in figure 13.4. Note that we actually need only half the sphere, because $Y(\hat{\mathbf{u}}, \mathbf{s}) = Y(-\hat{\mathbf{u}}, \mathbf{s})$, but working with the complete sphere is more convenient. With $\theta_0 = 0$, the problem reduces to 2D parallel projection (for multiple slices), which was shown to have a unique solution. It follows that with $|\theta_0| > 0$, the problem becomes overdetermined, and there are infinitely many ways to compute the solution. This can be seen as follows. Each point of Ω corresponds to a parallel projection. According to the central slice theorem, this provides a central plane perpendicular to $\hat{\mathbf{u}}$ of the 3D Fourier transform $\mathcal{L}(\boldsymbol{\nu})$ of $\Lambda(\mathbf{x})$. Thus, the set Ω_0 (i.e. all points on the equator of the unit sphere in fig 13.4), provides all planes intersecting the ν_z-axis, which is sufficient to recover the entire image $\Lambda(\mathbf{x})$ via inverse Fourier transform. The set Ω_{θ_0} with $\theta_0 > 0$ provides additional (oblique) planes through $\mathcal{L}(\boldsymbol{\nu})$, which are obviously redundant. A simple solution would be to select a sufficient subset from the data. However, if the data are noisy, a more stable solution is obtained by using all the measurements. This is achieved by computing $\mathcal{L}(\boldsymbol{\nu})$ from a linear combination of all available planes:

$$\mathcal{L}(\boldsymbol{\nu}) = \int_{\Omega_{\theta_0}} d\hat{\mathbf{u}}\, \mathcal{Y}(\hat{\mathbf{u}}, \boldsymbol{\nu}) H(\hat{\mathbf{u}}, \boldsymbol{\nu})\, \delta(\hat{\mathbf{u}}, \boldsymbol{\nu}). \tag{13.18}$$

Here, $\mathcal{Y}(\hat{\mathbf{u}}, \boldsymbol{\nu})$ is the 2D Fourier transform w.r.t. $\mathbf{s}$ of the projection $Y(\hat{\mathbf{u}}, \mathbf{s})$. The Dirac function $\delta(\hat{\mathbf{u}}, \boldsymbol{\nu})$ selects the parallel projections $\hat{\mathbf{u}}$ which are perpendicular to $\boldsymbol{\nu}$ (i.e. the points on the circle

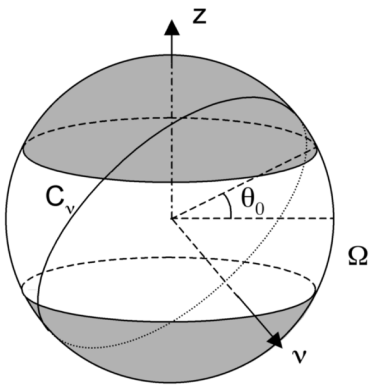

Figure 13.4: *Each point on the unit sphere corresponds to the direction of a parallel projection. An ideal rotating gamma camera with parallel hole collimator only travels through the points on the equator. An idealized 3D PET system would also acquires projections along oblique lines, it collects projections for all points of the set Ω. The set Ω, defined by θ_0, is the non-shaded portion of the unit sphere. To recover a particular frequency $\boldsymbol{\nu}$ (of the Fourier transform of the object), at least one point on the circle C_ν is required.*

C_ν in fig 13.4). Finally, the filter $H(\hat{\mathbf{u}}, \boldsymbol{\nu})$ assigns a particular weight to each of the available datasets $\mathcal{Y}(\hat{\mathbf{u}}, \boldsymbol{\nu})$. The combined weight for each frequency should equal unity, leading to the filter equation

$$\int_{\Omega_{\theta_0}} d\hat{\mathbf{u}} H(\hat{\mathbf{u}}, \boldsymbol{\nu}) \delta(\hat{\mathbf{u}}, \boldsymbol{\nu}) = 1. \tag{13.19}$$

A solution equivalent to that of unweighted least squares is obtained by assigning the same weight to all available data [8]. This results in the Colsher filter which can be written as:

$$\begin{array}{rcll} H_C(\hat{\mathbf{u}}, \boldsymbol{\nu}) & = & |\boldsymbol{\nu}|/(2\pi) & \text{if } \sin\psi \leq \sin\theta_0 \\ & = & |\boldsymbol{\nu}|/(4 \arcsin(\sin\theta_0 / \sin\psi)) & \text{if } \sin\psi > \sin\theta_0, \end{array} \tag{13.20}$$

where ψ is the angle between $\boldsymbol{\nu}$ and the z-axis: $\nu_z/|\boldsymbol{\nu}| = \cos\psi$. One could apply the direct Fourier reconstruction method here, by straightforward inverse Fourier transform of 13.18. However, a filtered backprojection approach is usually preferred, which can be written as

$$\Lambda(\mathbf{x}) = \int_{\Omega_{\theta_0}} d\hat{\mathbf{u}}\, Y^F(\hat{\mathbf{u}}, \mathbf{x} - (\mathbf{x} \cdot \hat{\mathbf{u}})\hat{\mathbf{u}}). \tag{13.21}$$

Here, Y^F is obtained by filtering Y with the Colsher filter (or another filter satisfying (13.19)): $Y^F(\hat{\mathbf{u}}, s) = \mathcal{F}^{-1}(H_c(\hat{\mathbf{u}}, \boldsymbol{\nu})\ \mathcal{Y}(\hat{\mathbf{u}}, \boldsymbol{\nu}))$. The coordinate $\mathbf{s} = \mathbf{x} - (\mathbf{x} \cdot \hat{\mathbf{u}})\hat{\mathbf{u}}$ is the projection of the point $\mathbf{x}$ on the plane perpendicular to $\hat{\mathbf{u}}$; it selects the LOR through $\mathbf{x}$ in the parallel projection $\hat{\mathbf{u}}$.

13.2.3.2 The reprojection algorithm

The previous analysis assumed that the acceptance angle θ_0 was a constant, independent of $\mathbf{x}$. As illustrated in fig 13.5 this is not the case in practice. The acceptance angle is maximum for the center of the field of view, it becomes smaller with increasing distance to the center, and vanishes near the axial edges of the field of view. In other words: the projections are complete for $\hat{\mathbf{u}}$ orthogonal to the

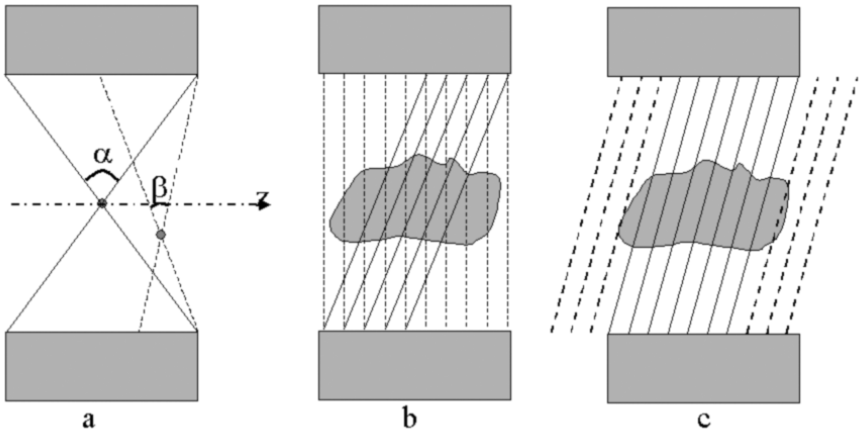

Figure 13.5: *An axial cross section through a cylindrical PET system, illustrating that the acceptance angle is position dependent (a). Oblique projections are truncated (b). In the reprojection algorithm, the missing oblique projections (dashed lines) are computed from a temporary multi-slice 2D reconstruction (c).*

z-axis (these are the 2D multislice parallel beam projections), and they are truncated for the oblique parallel projections. The truncation becomes more severe for more oblique projections (fig 13.5).

As the acceptance angle is position dependent, the required filtering is position dependent as well, and cannot be implemented as a shift-invariant convolution (or Fourier filter). Several strategies for dealing with this truncation have been developed. One approach is to subdivide the image in a set of regions, and then optimize a shift-invariant filter in each of the regions. The filter is determined by the smallest acceptance angle of the region, so some of the data will not be used. A good compromise between minimum data loss and practical implementation must be sought [9].

Another approach is to start with a first reconstruction, using the smallest acceptable angle over all positions $\mathbf{x}$ in the field of view. This usually means that only the parallel projections orthogonal to the z-axis are used. From this first reconstruction, the missing oblique projection values are computed (fig 13.5) and used to complete the measured oblique projections. This eliminates the truncation, and the 3D filtered backprojection method of the previous section can be applied. This method [10] was the standard 3D PET reconstruction method for several years, until it was replaced by the faster Fourier rebinning approach (see below).

13.2.3.3 Rebinning techniques

The complexity (estimated as the number of LORs) increases linearly with the axial extent for 2D PET, but quadratically for 3D PET. To keep the processing time acceptable, researchers have sought ways to reduce the size of the data as much as possible, while minimizing the loss of information induced by this reduction.

Most PET systems have a cylindrical detector surface: the detectors are located on rings with radius R, and the rings are combined in a cylinder along the z-axis. The data are usually organized in sinograms which can be written as:

$$Y_P(s,\phi,z,\Delta_z) = \int_{-\infty}^{\infty} dt \Lambda(s\cos\phi + t\hat{u}_x,\ s\sin\phi + t\hat{u}_y,\ z + t\hat{u}_z), \tag{13.22}$$

where $\hat{\mathbf{u}}$ is a unit vector in the direction of the LOR:

$$\hat{\mathbf{u}} = \mathbf{u}/||\mathbf{u}|| \text{ with } \mathbf{u} = (-\sin\phi, \cos\phi, \Delta_z/(2\sqrt{R^2 - s^2})).$$

The parameter s is the distance between the LOR and the z-axis. The LOR corresponds to a coincidence between detector points with axial positions $z - \Delta_z/2$ and $z + \Delta_z/2$. Finally, ϕ is the angle between the y-axis and the projection of the LOR on the xy-plane. The coordinates (s, ϕ, z) are identical to those often used in 2D tomography. Note that in practice $s \ll R$, and as a result the direction of the LOR, the vector $\hat{\mathbf{u}}$, is virtually independent of s. In other words, a set of LORs with fixed Δ_z can then be treated as a parallel projection with good approximation. LORs with $\Delta_z = 0$ are often called "direct" LORs, while LORs with $\Delta_z \neq 0$ are called "oblique".

The basic idea of rebinning algorithms is to compute estimates of the direct sinograms from the oblique sinograms. If the rebinning algorithm is good, most of the information from the oblique sinograms will go into these estimates. As a result, the data have been reduced from a complex 3D geometry into a much simpler 2D geometry without discarding measured signal. The final reconstruction can then be done with 2D algorithms, which tend to be much faster than fully 3D algorithms. A popular approach is to use Fourier rebinning, followed by maximum likelihood reconstruction.

Single slice and multi-slice rebinning

The simplest way to rebin the data, is to treat oblique LORs as direct LORs [11]. This corresponds to the approximation:

$$Y_P(s, \phi, z, \Delta_z) \simeq Y_P(s, \phi, z, 0). \tag{13.23}$$

The approximation is only exact if the object consists of points located on the z-axis, and it introduces mispositioning errors that increase with increasing distance to the z-axis and increasing Δ_z. Consequently, single slice rebinning is applicable when the object is small and positioned centrally in the scanner, or when Δ_z is small. The axial extent of most current PET-systems is too large to rebin an entire 3D data set with (13.23). However, single slice rebinning is used on all PET systems to reduce the sampling of the Δ_z-dimension in the 3D data, by combining sinograms with similar Δ_z. This typically reduces the data size with a factor of about 10, when compared to the finest possible sampling.

Application of (13.23) obviously causes blurring in the z-direction, in a degree proportional to the distance from the z-axis. However, it may also cause severe inconsistencies in the sinograms, producing blurring artifacts in the xy-planes of the reconstructed images as well. Lewitt et al [12] proposed to distribute the oblique LOR values $Y_P(s, \phi, z, \Delta_z)$ over all LORs with $z \in [z - \Delta_z R_f/(2R), z + \Delta_z R_f/(2R)]$, i.e. over all slices intersected by the LOR, and within a field of view with radius R_f. This so-called multi-slice rebinning reduces the inconsistencies in the sinograms, eliminating most of the xy blurring artifacts in the reconstruction. Unfortunately, the improvement comes at the cost of strong axial blurring. This blurring depends strongly on z, but it is found to be approximately independent of x and y. A z-dependent 1D axial filter is applied to reduce this axial blurring [12]. Multi-slice rebinning is superior to single slice rebinning, but the noise characteristics are not optimal

Fourier rebinning.

Fourier rebinning [13] is based on the frequency distance principle, which was explained above. The Fourier rebinning method is most simply formulated when the projection is written as follows:

$$Y(s, \phi, z, \delta) = \int_{-\infty}^{\infty} dt\ \Lambda(s\cos\phi - t\sin\phi,\ s\sin\phi + t\cos\phi,\ z + t\delta), \tag{13.24}$$

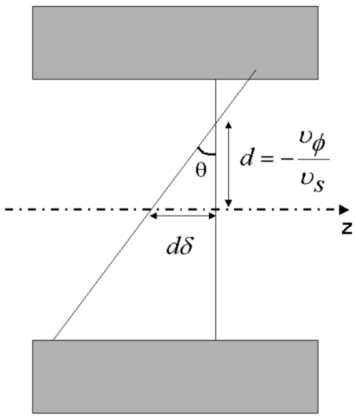

Figure 13.6: *Fourier rebinning: via the frequency distance principle, the distance from the rotation axis is obtained. This distance is used to identify the appropriate direct sinogram.*

with $\delta = \tan\theta$, where θ is the angle between the LOR and the xy-plane. The integration variable t is the distance between the position on the LOR and the z-axis. It follows that

$$Y(s,\phi,z,\delta) = \frac{Y_P(s,\phi,z,\Delta_z = 2\delta\sqrt{R^2-s^2})}{\sqrt{1+\delta^2}} \tag{13.25}$$

$$\simeq \frac{Y_P(s,\phi,z,\Delta_z = 2\delta R)}{\sqrt{1+\delta^2}}, \tag{13.26}$$

where the approximation is valid whenever $s \ll R$. In that case, no interpolation is needed; it suffices to scale the PET data Y_P with the weight factor $\sqrt{1+\delta^2}$. Fourier rebinning uses the frequency distance principle to find the distance d corresponding to a particular portion of the oblique sinogram. As illustrated in fig 13.6, that distance is used to locate the direct sinogram to which this portion should be assigned. Denoting with $\mathcal{Y}$ the 2D Fourier transform of Y w.r.t. s and ϕ, this can be written as

$$\mathcal{Y}(\nu_s,\nu_\phi,z,\delta) \simeq \mathcal{Y}(\nu_s,\nu_\phi,z-\delta\frac{\nu_\phi}{\nu_s},0). \tag{13.27}$$

This equation tells how to distribute frequency components from a particular oblique sinogram into different direct sinograms. Frequencies located on the line $\nu_\phi = -d\nu_s$ in the oblique sinogram z can be assigned to that same line in the direct sinogram $z+\delta d$.

The final rebinning algorithm (often called "FORE") is obtained by averaging all available estimates of the direct sinogram:

$$\begin{array}{lll} \mathcal{Y}(\nu_s,\nu_\phi,z,0) \simeq \frac{1}{\delta_{max}}\int_0^{\delta_{max}} d\delta \mathcal{Y}(\nu_s,\nu_\phi,z+\delta\frac{\nu_\phi}{\nu_s},\delta) & \text{if } |\nu_s| \gg 0 \\ \simeq \mathcal{Y}(0,0,z,0) & \text{if } \nu_s \simeq 0, \nu_\phi \simeq 0 \\ \simeq 0 & \text{if } |\nu_\phi/\nu_s| > R_f \end{array} \tag{13.28}$$

Note that the rebinning expression is only valid for large ν_s. In the low frequency range, only the direct sinogram is used. The last line of (13.28) holds because the image $\Lambda(x,y,z)$ is assumed to be zero outside the field of view $\sqrt{x^2+y^2} > R_f$.

A more rigorous mathematical derivation of the frequency distance relation is given in [14]. Alternative derivations based on exact rebinning expressions are given in [13].

After Fourier rebinning, the resulting 2D data set can be reconstructed with any 2D reconstruction algorithm. A popular method is the combination of Fourier rebinning with a 2D statistical reconstruction algorithm.

Exact rebinning methods

Fourier rebinning is an approximate method, but was found to be sufficiently accurate for apertures up to $\theta_0 = 25^o$, and it is therefore largely sufficient for most current PET systems. However, there is a tendency toward still larger acceptance angles, and a more exact Fourier rebinning algorithm may be needed in the future. An example of an "exact" rebinning algorithm is FOREX [13]. It is exact in the sense that the rebinning expression is exact for the continuous 3D x-ray transform.

According to the central section theorem, the 2D Fourier transform of a projection $Y(s, \phi, z, \delta)$ equals a cross section through the 3D Fourier transform of the image $\Lambda(x, y, z)$:

$$\mathcal{Y}_{13}(\nu_s, \phi, \nu_z, \delta) = \mathcal{L}(\nu_s \cos\phi + \nu_z \delta \sin\phi,\ \nu_s \sin\phi - \nu_z \delta \cos\phi,\ \nu_z). \tag{13.29}$$

The subscript of $\mathcal{Y}_{13}$ denotes Fourier transform w.r.t. s and z. Defining

$$\begin{aligned} \sigma &= \arctan(\delta\nu_z/\nu_s) \\ \nu'_s &= \nu_s\sqrt{1+\delta^2\nu_z^2/\nu_s^2} \end{aligned}$$

equation (13.29) can be rewritten as

$$\mathcal{Y}_{13}(\nu_s, \phi, \nu_z, \delta) = \mathcal{L}(\nu'_s \cos(\phi-\sigma),\ \nu'_s \sin(\phi-\sigma),\ \nu_z). \tag{13.30}$$

Taking the 1D Fourier transform w.r.t. ϕ yields

$$\mathcal{Y}_{123}(\nu_s, \nu_\phi, \nu_z, \delta) = e^{-i\nu_\phi\sigma} \int_0^{2\pi} e^{-i\nu_\phi\phi} \mathcal{L}(\nu'_s \cos\phi,\ \nu'_s \sin\phi,\ \nu_z) d\phi. \tag{13.31}$$

By comparing the expressions for $\mathcal{Y}_{123}(\nu_s, \nu_\phi, \nu_z, \delta)$ and $\mathcal{Y}_{123}(\nu_s, \nu_\phi, \nu_z, 0)$ one finally obtains:

$$\mathcal{Y}_{123}(\nu_s, \nu_\phi, \nu_z, \delta) = e^{-i\nu_\phi\sigma} \mathcal{Y}_{123}(\nu'_s, \nu_\phi, \nu_z, 0). \tag{13.32}$$

A problem of FOREX is that it needs the 1D Fourier transform along z, which cannot be computed for truncated projections. Similar as with 3D filtered backprojection, the problem can be avoided by completing the truncated projections with synthetic data. Fortunately, expression (13.32) can be used in both ways, and allows to estimate (missing) oblique sinograms from the available direct sinograms. The resulting algorithm is slower than FORE, but still considerably faster than 3D filtered backprojection with reprojection.

13.2.4 Time-of-flight PET

In time-of-flight PET, the difference in arrival time of the two detected photons is used to estimate the position of their emission along the LOR. The uncertainty on the time estimation results in a similar uncertainty on the position estimation, which can be well modeled as a Gaussian distribution. As a result, the TOF-projections correspond to Gaussian convolutions along lines, rather than to line integrals, as illustrated in fig 13.7. The corresponding TOF-backprojection corresponds to convolving the measured data with the same 1D Gaussians, followed by summation over all angles.

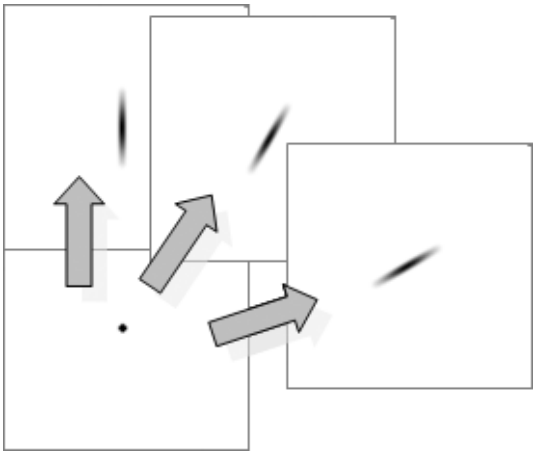

Figure 13.7: *TOF-projection can be well modeled as a 1D Gaussian convolution in the direction of the LOR.*

Recall from (13.2) that the regular projection followed by the regular backprojection corresponds to a convolution with a blurring filter

$$B_{nonTOF}(x,y) = \frac{1}{\sqrt{x^2+y^2}}. \tag{13.33}$$

The Fourier transform of $1/\sqrt{x^2+y^2}$ equals $1/\sqrt{\nu_x^2+\nu_y^2}$. Consequently, this blurring can be undone by the ramp filter $\sqrt{\nu_x^2+\nu_y^2}$, which can be applied either before or after backprojection (see section 13.2.1.3).

If σ_{TOF} is the standard deviation of the TOF-blurring kernel, then TOF-projection followed by TOF-backprojection corresponds to convolution with the blurring kernel

$$B_{TOF}(x,y) = \frac{\text{Gauss}(x,\, y,\, \sqrt{2}\,\sigma_{TOF})}{\sqrt{x^2+y^2}} \tag{13.34}$$

$$= \frac{1}{\sqrt{x^2+y^2}} \; \frac{1}{2\sqrt{\pi}\sigma_{TOF}} \exp\left(-\frac{x^2+y^2}{4\sigma_{TOF}^2}\right). \tag{13.35}$$

Note that the Gaussian in the equation above has a standard deviation of $\sqrt{2}\sigma_{TOF}$. This is because the Gaussian blurring is present in the projection *and* in the backprojection. The filter required in TOF-PET filtered backprojection is derived by inverting the Fourier transform of B_{TOF}, and equals

$$\text{TOF-recon-filter}(\nu) = \frac{1}{\exp(-2\pi^2\sigma_{TOF}^2\nu^2)\; I_0(2\pi^2\sigma_{TOF}^2\nu^2)}, \tag{13.36}$$

where I_0 is the zeroth order modified Bessel function of the first kind.

This FBP expression is obtained by using the "natural" TOF backprojection, defined as the adjoint of the TOF projection. This backprojection appears also in least squares approaches, and it has been shown that with this backprojection definition, FBP is optimal in an (unweighted) least squares sense [15]. However, TOF-PET data are redundant, and different backprojection definitions could be used; they would yield different expressions for $B_{TOF}(x,y)$ in (13.34) and therefore different TOF-reconstruction filters.

Just as for non-TOF PET, exact and approximate rebinning algorithms for TOF-PET have been derived to reduce the data size. Because the TOF-information limits the backprojection to a small region, the errors from approximate rebinning are typically much smaller than in the non-TOF case.

13.3 Iterative reconstruction

13.3.1 Introduction

13.3.1.1 Discretisation

In analytical reconstruction, it is initially assumed that the unknown object can be represented as a function $\Lambda(\vec{x})$ with $\vec{x} \in \mathbb{R}^3$, and that the acquired data can be represented as a function $Y(s, \theta)$ with $s \in \mathbb{R}^2$ and θ a unit vector in $\mathbb{R}^2$ or $\mathbb{R}^3$. Then the reconstruction algorithm is derived by mathematical inversion (assuming some convenient properties for Λ and Y), and finally the resulting algorithm is discretized to make it ready for software implementation. In iterative reconstruction, one usually starts by discretizing the problem. This reduces the reconstruction problem to finding a finite set of unknown values from a finite set of equations, a problem which can be solved with numerical inversion. The advantage of numerical inversion is that only a model for the acquisition process is needed, not for its inverse. That makes it easier (although it still may be non-trivial) to take into account some of the undesired but unavoidable effects that complicate the acquisition, such as photon attenuation, position dependent resolution, gaps between the detectors, patient motion etc.

After discretisation, the unknown image values and the known measured values can be represented as column vectors $\boldsymbol{\lambda}$ and $\mathbf{y}$. The PET or SPECT acquisition process is characterized by the system matrix $\mathbf{A}$ and an additive contribution $\bar{\mathbf{b}}$, and $\mathbf{n}$ is the measurement noise:

$$\mathbf{y} = \mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}} + \mathbf{n} \qquad \text{or} \qquad \mathbf{y}_i = \sum_{j=1}^{J} \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i + \mathbf{n}_i, \quad i = 1 \ldots I. \tag{13.37}$$

The symbol $\mathbf{y}_i$ denotes the number of photons measured at LOR i, where the index i runs over all the sinogram elements (merging the 3 or 4 sinogram dimensions into a single index). The index j runs over all image voxels, and $\mathbf{A}_{ij}$ is the probability that a unit of radioactivity in j gives rise to the detection of a photon (SPECT) or photon pair (PET) in LOR i. The estimate of the additive contribution is denoted as $\bar{\mathbf{b}}$. This estimate is assumed to be noise-free and includes e.g. scatter and randoms in PET or cross-talk between different energy windows in multi-tracer SPECT studies. Finally, $\mathbf{n}_i$ represents the noise contribution in LOR i.

Image reconstruction now consists of finding $\boldsymbol{\lambda}$, given $\mathbf{A}$, $\mathbf{y}$ and $\bar{\mathbf{b}}$, and a statistical model for $\mathbf{n}$.

For further reading about this subject, the excellent recent review paper on iterative reconstruction by Qi and Leahy [16] is an ideal starting point.

13.3.1.2 Objective functions

The presence of the noise precludes exact reconstruction. For that reason, the reconstruction is often treated as an optimisation task: it is assumed that a useful clinical image can be obtained by maximizing some well chosen objective function. When the statistics of the noise are known, one can apply a

Bayesian approach, searching for the image $\hat{\boldsymbol{\lambda}}$ that maximizes the conditional probability on the data:

$$\begin{aligned}
\hat{\boldsymbol{\lambda}} &= \arg\max_{\lambda} \; p(\boldsymbol{\lambda}|\mathbf{y}) \\
&= \arg\max_{\lambda} \; \frac{p(\mathbf{y}|\boldsymbol{\lambda})\; p(\boldsymbol{\lambda})}{p(\mathbf{y})} \\
&= \arg\max_{\lambda} \; p(\mathbf{y}|\boldsymbol{\lambda})\; p(\boldsymbol{\lambda}) \qquad (13.38) \\
&= \arg\max_{\lambda} \; (\ln p(\mathbf{y}|\boldsymbol{\lambda}) + \ln p(\boldsymbol{\lambda})), \qquad (13.39)
\end{aligned}$$

the second equation is Bayes' rule, the third equation holds because $\mathbf{y}$ does not depend on $\boldsymbol{\lambda}$, and the fourth equation is valid because computing the logarithm does not change the position of the maximum. The probability $p(\mathbf{y}|\boldsymbol{\lambda})$ gives the likelihood of measuring a particular sinogram $\mathbf{y}$, when the tracer distribution equals $\boldsymbol{\lambda}$. This distribution is often simply called the *likelihood*. The probability $p(\boldsymbol{\lambda})$ represents the a priori knowledge about the tracer distribution, available already before doing the PET or SPECT acquisition. This probability is often called the *prior* distribution. The knowledge available after the measurements equals $p(\mathbf{y}|\boldsymbol{\lambda})\; p(\boldsymbol{\lambda})$ and is called the *posterior* distribution. To keep things simple, one often assumes that no prior information is available, i.e. $p(\boldsymbol{\lambda}|\mathbf{y}) \sim p(\mathbf{y}|\boldsymbol{\lambda})$. Finding the solution then reduces to maximizing the likelihood $p(\mathbf{y}|\boldsymbol{\lambda})$ (or its logarithm). In this section, maximum likelihood algorithms are discussed. Maximum-a-posteriori algorithms will be discussed later in section 13.3.5.4, as a strategy to suppress noise propagation.

A popular approach to solve equations of the form (13.37) is least squares (LS) estimation. This is equivalent to a maximum-likelihood approach, if one assumes that the noise is Gaussian with zero mean and a fixed, position independent standard deviation σ. The probability to measure the noisy value $\mathbf{y}_i$ when the expected value was $\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i$ then equals:

$$p_{LS}(\mathbf{y}_i|\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\mathbf{y}_i - (\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i))^2}{2\sigma^2}\right). \qquad (13.40)$$

Because the noise in the sinogram is not correlated, the likelihood (i.e. the probability of measuring the entire noisy sinogram $\mathbf{y}$) equals

$$p_{LS}(\mathbf{y}|\boldsymbol{\lambda}) \;=\; p_{LS}(\mathbf{y}|\mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}}) \;=\; \prod_i p_{LS}(\mathbf{y}_i|\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i). \qquad (13.41)$$

It is more convenient to maximize the logarithm of p_{LS}; dropping constants one finally obtains the objective function L_{LS}:

$$L_{LS} \;=\; -\sum_i (\mathbf{y}_i - (\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i))^2 \;=\; -(\mathbf{y} - (\mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}}))'(\mathbf{y} - (\mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}})), \qquad (13.42)$$

where the prime denotes matrix transpose. Setting the first derivatives with respect to $\boldsymbol{\lambda}_j$ to zero for all j, one obtains

$$\begin{aligned}
&\mathbf{A}'(\mathbf{y} - \mathbf{A}\boldsymbol{\lambda} - \bar{\mathbf{b}}) = 0 \\
&\boldsymbol{\lambda} = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'(\mathbf{y} - \bar{\mathbf{b}}), \qquad (13.43)
\end{aligned}$$

provided that $\mathbf{A}'\mathbf{A}$ is non-singular. The operator $\mathbf{A}$ is the discrete projection; its transpose $\mathbf{A}'$ is the discrete backprojection. Its analytical counterpart was given in equation (13.2) and illustrated in fig.

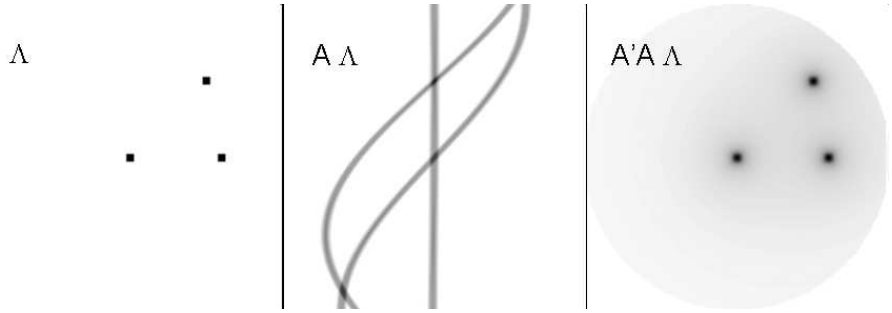

Figure 13.8: *The image of point sources is projected and backprojected again along ideal parallel beams. This yields a shift invariant blurring.*

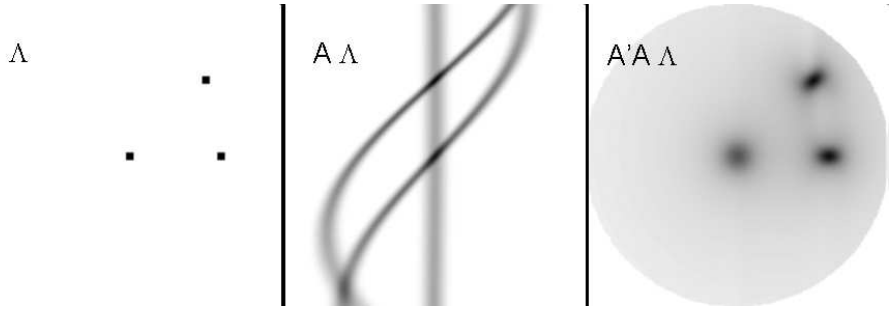

Figure 13.9: *The image of point sources is projected and backprojected again with collimator blurring. This yields a shift variant blurring.*

13.2. The same figure shows that the operator $\mathbf{A}'\mathbf{A}$ behaves as a blurring filter. Figure 13.8 is similar, but illustrates $\mathbf{A}$ and $\mathbf{A}'\mathbf{A}$ on an image of three point sources, using ideal parallel beam projection. The figure shows the resulting point spread functions of $\mathbf{A}'\mathbf{A}$ for each of the point sources. They are identical: for ideal parallel beam projection, $\mathbf{A}'\mathbf{A}$ is shift invariant, equivalent to a convolution. It follows that $(\mathbf{A}'\mathbf{A})^{-1}$ is the corresponding shift invariant deconvolution, which is easily computed via the Fourier transform. In this situation, least squares reconstruction (eq 13.43) is the discrete equivalent of the backproject-then-filter algorithm (eq 13.15), applied to the data after precorrection for $\bar{\mathbf{b}}$.

Figure 13.9 illustrates $\mathbf{A}$ and $\mathbf{A}'\mathbf{A}$ for a projector that models the position dependent blurring of a typical parallel beam SPECT collimator. The blurring induced by $\mathbf{A}'\mathbf{A}$ is now shift variant - it cannot be modelled as a convolution, and its inverse cannot be computed with the Fourier transform. For real life problems, direct inversion of $\mathbf{A}'\mathbf{A}$ is not feasible. Instead, one applies iterative optimisation to find the maximum of (13.42).

It is known that the number of detected photons is subject to Poisson noise, not to uniform Gaussian noise. The Poisson distribution can be well approximated with a Gaussian distribution, where the variance of the Gaussian equals its mean. With this approximation, σ must be replaced by σ_i in (13.40) because now we have a different Gaussian distribution for every sinogram pixel i. Proceeding as before one obtains the weighted least squares (WLS) objective function:

$$\begin{aligned} L_{WLS} &= -\sum_i \frac{(\mathbf{y}_i - (\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i))^2}{\sigma_i^2} \\ &= -(\mathbf{y} - (\mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}}))' \, \mathbf{C_y}^{-1} \, (\mathbf{y} - (\mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}})), \end{aligned} \tag{13.44}$$

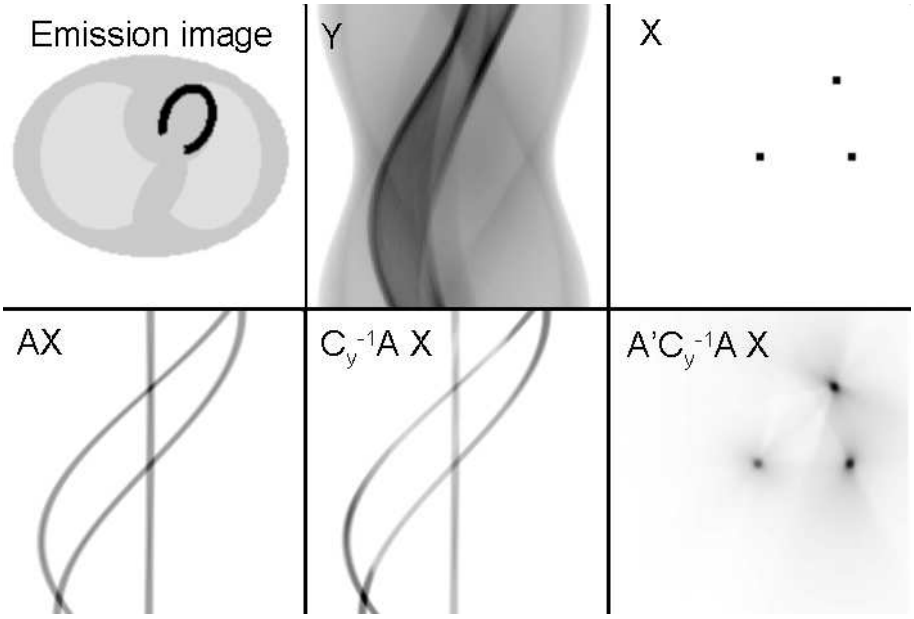

Figure 13.10: *The operator $\mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\mathbf{A}$ is derived for a particular activity distribution (top left), and then applied to a few point sources X. Although ideal parallel beam projection was used, shift variant blurring is obtained.*

where $\mathbf{C_y}$ is the covariance matrix of the data. For emission tomography, it is a diagonal matrix (all covariances are zero) with elements $\mathbf{C_y}[i,i] = \sigma_i^2$. The corresponding WLS-reconstruction can be written as:

$$\boldsymbol{\lambda} = (\mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\mathbf{A})^{-1}\ \mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\ (\mathbf{y} - \bar{\mathbf{b}}). \tag{13.45}$$

The operator $\mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\mathbf{A}$ is always shift variant, even for ideal parallel beam tomography. This is illustrated in fig 13.10. The noise-free sinogram $\bar{\mathbf{y}}$ is computed for a particular activity distribution. Setting $\mathbf{C_y} = \text{diag}(\bar{\mathbf{y}})$, we can analyze the operator $\mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\mathbf{A}$ by applying it to the image of a few point sources, called $\mathbf{x}$ in the figure. The image $\mathbf{x}$ is projected, the sinogram $\mathbf{Ax}$ is divided by $\bar{\mathbf{y}}$ on a pixel basis, and the result is backprojected. Clearly, position dependent blurring is obtained. Consequently, iterative optimisation must be used for weighted least squares reconstruction.

In practice, because one only has a noisy sinogram $\mathbf{y}$, the noise free sinogram $\bar{\mathbf{y}}$ must be estimated to find $\mathbf{C_y}$. There are basically two approaches. In the first approach, $\bar{\mathbf{y}}$ is estimated from $\mathbf{y}$ e.g. by smoothing $\mathbf{y}$ to suppress the noise. In the second approach, $\bar{\mathbf{y}}$ is estimated as $\mathbf{A}\boldsymbol{\lambda}^{(k)} + \bar{\mathbf{b}}$ during the iterative optimisation, where $\boldsymbol{\lambda}^{(k)}$ is the estimate of the reconstruction available at iteration k. A drawback of the first approach is that the noise on the data affects the weights, with a tendency to give higher weight when the noise contribution happens to be negative. A complication of the second approach is that it makes σ_i a function of $\boldsymbol{\lambda}$. In that case, the normalizing amplitude $1/(\sqrt{2\pi}\sigma_i)$ of the Gaussians cannot be dropped, implying that an additional term $-\sum_i \ln \sigma_i$ should be added to 13.44.

It is possible to use the Poisson distribution itself, instead of approximating it with Gaussians. The probability of the noise realisation $\mathbf{y}_i$ then becomes

$$p_{ML}(\mathbf{y}_i|\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i) \ = \ \frac{e^{-(\sum_j \mathbf{A}_{ij}\lambda_j + \bar{\mathbf{b}}_i)}(\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i)^{\mathbf{y}_i}}{\mathbf{y}_i!}. \tag{13.46}$$

Proceeding as before, one finds for the log likelihood function:

$$\begin{aligned} \ln\left(\prod_i p_{LS}(\mathbf{y}_i|\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i)\right) &= \sum_i \mathbf{y}_i \ln(\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i) - (\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i) - \ln \mathbf{y}_i! \\ L_{ML} &= \sum_i \mathbf{y}_i \ln(\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i) - (\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i). \quad (13.47) \end{aligned}$$

Note that the term $\ln \mathbf{y}_i!$ can be dropped, because it is not a function of $\boldsymbol{\lambda}$. Because L_{ML} is a non-linear function of $\boldsymbol{\lambda}$, the solution cannot be written as a product of matrices. However, it is sometimes helpful to know that the features of the Poisson-objective function are often very similar to those of the weighted least squares function (13.44).

13.3.2 Optimisation algorithms

Many iterative reconstruction algorithms have been proposed to optimize the objective functions L_{WLS} and L_{ML}. Here only two approaches are briefly described: preconditioned conjugate gradient methods and optimisation transfer, with expecation maximisation as a special case of the latter.

13.3.2.1 Preconditioned gradient methods

The objective function will be optimised when its first derivatives are zero:

$$\hat{\mathbf{y}}_i = \sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j + \bar{\mathbf{b}}_i \quad (13.48)$$

$$\frac{\partial L_{WLS}(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} = \sum_i \mathbf{A}_{ij}\frac{\mathbf{y}_i - \hat{\mathbf{y}}_i}{\sigma_i^2} \quad (13.49)$$

$$\frac{\partial L_{ML}(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} = \sum_i \mathbf{A}_{ij}\frac{\mathbf{y}_i - \hat{\mathbf{y}}_i}{\hat{\mathbf{y}}_i}. \quad (13.50)$$

The optimisation can be carried out by a steepest ascent method, which can be formulated as follows:

$$\begin{aligned} \mathbf{d}^k &= \nabla L(\boldsymbol{\lambda}^{k-1}) \\ \boldsymbol{\lambda}^k &= \boldsymbol{\lambda}^{k-1} + \alpha_k \mathbf{d}^k \\ \alpha_k &= \arg\max_\alpha L(\boldsymbol{\lambda}^{k-1} + \alpha \mathbf{d}^k) \end{aligned} \quad (13.51)$$

where the superscripts k and $k-1$ denote the iteration numbers and ∇L is the vector of first derivatives of L with respect to $\boldsymbol{\lambda}_j$.

Steepest gradient ascent is known to be suboptimal, requiring many iterations for reasonable convergence. To find a better update, we require that after the update, the first derivatives of L are zero as intended. Approximating this with a first order Taylor expansion yields

$$\begin{aligned} \nabla L(\boldsymbol{\lambda}^{k-1} + \mathbf{p}^k) &= 0 \\ \nabla L(\boldsymbol{\lambda}^{k-1}) + \mathbf{H}\mathbf{p}^k &\simeq 0 \\ \mathbf{p}^k &\simeq -\mathbf{H}^{-1}\nabla L(\boldsymbol{\lambda}^{k-1}) = -\mathbf{H}^{-1}\mathbf{d}^k \end{aligned} \quad (13.52)$$

where the Hessian $\mathbf{H}$ is the matrix of second derivatives of L. This is obviously a very large matrix, but its elements are relatively easy to compute:

$$\text{for WLS:} \quad \mathbf{H}_{jk} = -\sum_i \frac{\mathbf{A}_{ij}\mathbf{A}_{ik}}{\sigma_i^2} = -(\mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\mathbf{A})[j,k] \tag{13.53}$$

$$\text{for ML} \quad \mathbf{H}_{jk} = -\sum_i \frac{\mathbf{A}_{ij}\mathbf{A}_{ik}\mathbf{y}_i}{\hat{\mathbf{y}}_i^2} \simeq \sum_i \frac{\mathbf{A}_{ij}\mathbf{A}_{ik}}{\hat{\mathbf{y}}_i} \tag{13.54}$$

$$\simeq -(\mathbf{A}'{\mathbf{C}_\mathbf{y}}^{-1}\mathbf{A})[j,k] \quad \text{if } \hat{\mathbf{y}} \simeq \bar{\mathbf{y}}. \tag{13.55}$$

For a Gaussian likelihood, (13.52) is in fact exact, and a single iteration would suffice. As shown before, however, it is usually impossible to compute $\mathbf{H}^{-1}$. Instead, one can use approximations to the Hessian (or other heuristics) to obtain a good $\mathbf{M}$ to derive a so-called preconditioned gradient ascent algorithm:

$$\begin{aligned} \mathbf{d}^k &= \nabla L(\boldsymbol{\lambda}^{k-1}) \\ \boldsymbol{\lambda}^k &= \boldsymbol{\lambda}^{k-1} + \alpha_k \mathbf{M}\mathbf{d}^k. \end{aligned} \tag{13.56}$$

To ensure that the convergence is preserved, the matrix $\mathbf{M}$ must be symmetric positive definite (note that $-\mathbf{H}^{-1}$ is symmetric positive definite, since $\mathbf{H}$ is symmetric negative definite, if $\mathbf{A}$ has maximum rank).

A simple way to obtain a reasonable $\mathbf{M}$ is to use only the diagonal elements of $\mathbf{H}$: $\mathbf{M}_{ii} = -1/\mathbf{H}_{ii}$ and $\mathbf{M}_{ij} = 0$ if $i \neq j$. A more sophisticated approach is discussed in [17]: a circulant, i.e. shift invariant, approximation of the Hessian is proposed. Such an approximation is easily computed by fixing j at a particular location in the image in (13.53) or (13.54), which yields an image that can be considered as the point spread function of a convolution operator. This shift invariant operator is then inverted via the Fourier transform, yielding a non-diagonal matrix $\mathbf{M}$. For cases where the true Hessian depends heavily on position, the approach could be repeated for a few well chosen positions j, applying linear interpolation for all other positions.

13.3.2.2 Conjugate gradient methods

Fig. 13.11 shows the convergence of the steepest gradient ascent algorithm for a nearly quadratic function of two variables. In every iteration, the algorithm starts moving in the direction of the maximum gradient (i.e. perpendicular to the iso-contour), and keeps moving along the same line until a maximum is reached (i.e. until the line is tangent to the iso-contour). Often, this leads to a zigzag line, requiring many iterations for good convergence.

The conjugate gradient algorithm is designed to avoid these oscillations [18]. The first iteration is identical to that of steepest gradient ascent. However, in the following iterations, the algorithm attempts to move in a direction for which the gradient along the previous direction(s) remains the same (i.e equal to zero). The idea is to eliminate the need for a new optimisation along these previous directions. Let $\mathbf{d}_{\text{old}}$ be the previous direction, and $\mathbf{H}$ the Hessian matrix (i.e. the second derivatives). We now require that the new direction $\mathbf{d}_{\text{new}}$ is such that the gradient along $\mathbf{d}_{\text{old}}$ does not change. When moving in direction $\mathbf{d}_{\text{new}}$, the gradient will change (using a quadratic approximation) as $\mathbf{H}\,\mathbf{d}_{\text{new}}$. Requiring that the resulting change along $\mathbf{d}_{\text{old}}$ is zero yields the condition

$$\mathbf{d}'_{\text{old}}\ \mathbf{H}\ \mathbf{d}_{\text{new}} = 0. \tag{13.57}$$

This behaviour is illustrated by the dashed line in fig 13.11: in the second iteration, the algorithm moves in a direction such that the trajectory cuts the iso-contours at the same angle as in the starting

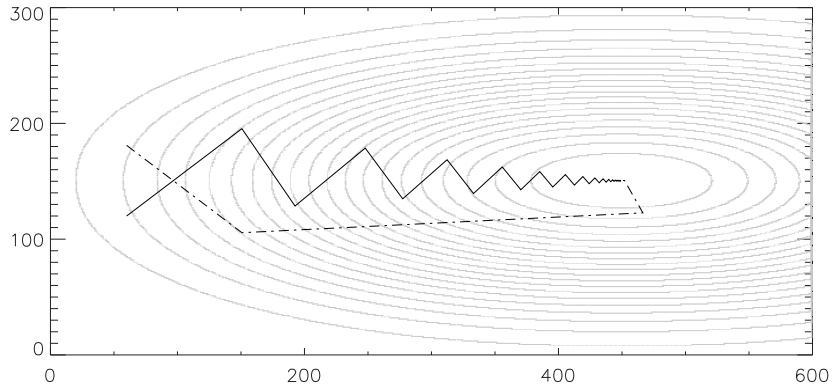

Figure 13.11: *The dotted lines are isocontours of the objective function. The solid line shows the convergence of the steepest gradient ascent algorithm, the dashed line the convergence of conjugent gradient ascent. Note that the starting points are equivalent because of the symmetry. The objective function equals* $-(a|x-x_0|^p + b|y-y_0|^p)$*, with* $p = 2.15$.

point. For a quadratic function in n dimensions, convergence is obtained after no more than n iterations. Because the function in fig 13.11 is not quadratic, more than 2 iterations are required for full convergence.

It turns out that the new direction can be easily computed from the previous ones, without computation of the Hessian $\mathbf{H}$. The Polak-Ribiere algorithm is given by [18]:

$$\begin{aligned} \mathbf{g}_{\text{new}} &= \nabla L(\boldsymbol{\lambda}_{\text{old}}) \\ \gamma &= \frac{(\mathbf{g}_{\text{new}} - \mathbf{g}_{\text{old}})' \, \mathbf{g}_{\text{new}}}{\mathbf{g}'_{\text{old}} \, \mathbf{g}_{\text{old}}} \\ \mathbf{d}_{\text{new}} &= \mathbf{g}_{\text{new}} + \gamma \, \mathbf{d}_{\text{old}} \\ \alpha &= \arg\max_{\alpha} L(\boldsymbol{\lambda}_{\text{old}} + \alpha \, \mathbf{d}_{\text{new}}) \\ \boldsymbol{\lambda}_{\text{new}} &= \boldsymbol{\lambda}_{\text{old}} + \alpha \, \mathbf{d}_{\text{new}}. \end{aligned} \tag{13.58}$$

This algorithm requires storage of the previous gradient $\mathbf{g}_{\text{old}}$ and the previous search direction $\mathbf{d}_{\text{old}}$. In each iteration it computes the new gradient and search direction, and applies a line search along the new direction.

13.3.2.3 Preconditioned conjugate gradient methods

Both techniques mentioned above can be combined to obtain a fast reconstruction algorithm, as described in [17]. The preconditioned conjugate gradient ascent algorithm (with preconditioning matrix

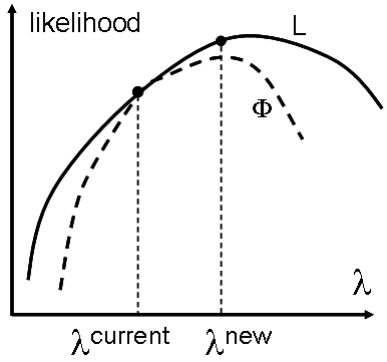

Figure 13.12: *Optimisation transfer: a surrogate function is designed, which is equal to the likelihood in the current reconstruction, and less or equal everywhere else.*

M) can be written as follows:

$$\begin{aligned}
\mathbf{g}_{\text{new}} &= \nabla L(\boldsymbol{\lambda}_{\text{old}}) \\
\mathbf{p}_{\text{new}} &= \mathbf{M}\mathbf{g}_{\text{new}} \\
\gamma &= \frac{(\mathbf{g}_{\text{new}} - \mathbf{g}_{\text{old}})' \, \mathbf{p}_{\text{new}}}{\mathbf{g}'_{\text{old}} \, \mathbf{p}_{\text{old}}} \\
\mathbf{d}_{\text{new}} &= \mathbf{p}_{\text{new}} + \gamma \, \mathbf{d}_{\text{old}} \\
\alpha &= \arg\max_{\alpha} L(\boldsymbol{\lambda}_{\text{old}} + \alpha \, \mathbf{d}_{\text{new}}) \\
\boldsymbol{\lambda}_{\text{new}} &= \boldsymbol{\lambda}_{\text{old}} + \alpha \, \mathbf{d}_{\text{new}}.
\end{aligned} \tag{13.59}$$

13.3.2.4 Optimisation transfer

The log-likelihood function (13.47) can be maximized by setting its gradients (13.50) to zero for all $j = 1 \ldots J$. A problem is that each of these derivatives is a function of many voxels of $\boldsymbol{\lambda}$, which makes the set of equations very hard to solve. The idea of "optimisation transfer" is to replace the problematic log-likelihood function with another function $\Phi(\boldsymbol{\lambda})$ that leads to a simpler set of equations, usually one where the derivative with respect to $\boldsymbol{\lambda}_j$ is only a function of $\boldsymbol{\lambda}_j$ and not of the other voxels of $\boldsymbol{\lambda}$. That makes the problem separable into J one-dimensional optimisations, which are easily solved. Ideally, Φ and L should have the same optimum, but that is asking for too much. The key is to design $\Phi(\boldsymbol{\lambda})$ in such a way that maximisation of $\Phi(\boldsymbol{\lambda})$ is guaranteed to increase $L(\boldsymbol{\lambda})$. This leads to an iterative algorithm, since new functions Φ will have to be designed and maximised repeatedly to maximise L. At iteration k, the surrogate function $\Phi(\boldsymbol{\lambda})$ needs to satisfy the following conditions (illustrated in fig 13.12):

$$\Phi(\boldsymbol{\lambda}^{(k)}) = L(\boldsymbol{\lambda}^{(k)}) \tag{13.60}$$

$$\Phi(X) \leq L(X) \tag{13.61}$$

It follows that the new reconstruction image $\boldsymbol{\lambda}^{(k+1)}$ which maximises $\Phi(\boldsymbol{\lambda})$ has a higher likelihood than $\boldsymbol{\lambda}^{(k)}$:

$$L(\boldsymbol{\lambda}^{(k)}) = \Phi(\boldsymbol{\lambda}^{(k)}) \leq \Phi(\boldsymbol{\lambda}^{(k+1)}) \leq L(\boldsymbol{\lambda}^{(k+1)}). \tag{13.62}$$

Several algorithms for maximum likelihood and maximum a posteriori reconstruction in emission and transmission tomography have been developed with this approach. De Pierro [19] has shown how

the well known MLEM algorithm can be derived using the optimisation transfer principle. He also showed how this alternative derivation provides a natural way to extend it to a maximum a posteriori algorithm.

13.3.3 Maximum Likelihood Expectation Maximisation

13.3.3.1 Reconstruction from sinogram data

There are many ways to derive the maximum likelihood expectation maximisation (MLEM) algorithm, including the original statistical derivation by Shepp and Vardi [20] (based on the work by Dempster et al) and the optimisation transfer approach by De Pierro [19]. Below only the expectation maximisation recipe is given.

Recall that we wish to find the image $\boldsymbol{\lambda}$ that maximizes the likelihood function L_{ML} of (13.47). The expectation maximisation does this in a remarkable way. Instead of concentrating on L_{ML}, an alternative (different) likelihood function is derived by introducing a set of so-called "complete data" $\mathbf{x}_{ij}$, defined as the number of photons that were emitted at voxel j and detected in LOR i during the measurement. These unobserved data are "complete" in the sense that they describe in more detail than the observed data $\mathbf{y}_i$ what happened during the measurement. These variables $\mathbf{x}_{ij}$ are Poisson distributed. Just as for the actual data $\mathbf{y}_i$, one can write the log-likelihood function for observing the data $\mathbf{x}_{ij}$ while $\bar{\mathbf{x}}_{ij} = \mathbf{A}_{ij}\boldsymbol{\lambda}_j$ were expected:

$$L_x(\boldsymbol{\lambda}) = \sum_i \sum_j \mathbf{x}_{ij} \ln(\mathbf{A}_{ij}\boldsymbol{\lambda}_j) - \mathbf{A}_{ij}\boldsymbol{\lambda}_j. \tag{13.63}$$

However, this likelihood cannot be computed, because the data $\mathbf{x}_{ij}$ are not available. The emission measurement only produces sums of the complete data, since

$$\mathbf{y}_i = \sum_j \mathbf{A}_{ij}\mathbf{x}_{ij} + \mathbf{b}_i, \tag{13.64}$$

where $\mathbf{b}_i$ represents the actual (also unobserved) additive contribution $\mathbf{b}_i$ in LOR i.

The EM recipe prescribes to compute the expectation of L_x, based on the available data and on the current reconstruction $\boldsymbol{\lambda}^{(k)}$. Based on the reconstruction alone, one would write $E(\mathbf{x}_{ij}|\boldsymbol{\lambda}^{(k)}) = \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)}$. However, we also know that $\mathbf{x}_{ij}$ should satisfy (13.64). One can show that this leads to the following estimate:

$$E(\mathbf{x}_{ij}|\boldsymbol{\lambda}^{(k)}, \mathbf{y}) = \frac{\mathbf{y}_i}{\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)} + \bar{\mathbf{b}}_i} \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)}, \tag{13.65}$$

where $\bar{\mathbf{b}}_i$ is the noise-free estimate of $\mathbf{b}_i$, which we assume to be available. Inserting this in (13.63) produces the expectation of $L_x(\boldsymbol{\lambda})$ and completes the expectation (E) step. For the maximisation (M) step, one simply sets the first derivatives to zero:

$$\frac{\partial L_x(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} = \sum_i \left(\frac{\mathbf{y}_i}{\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)} + \bar{\mathbf{b}}_i} \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)} \frac{1}{\boldsymbol{\lambda}_j} - \mathbf{A}_{ij} \right) = 0. \tag{13.66}$$

This is easily solved for $\boldsymbol{\lambda}_j$, yielding the new reconstruction $\boldsymbol{\lambda}_j^{(k+1)}$

$$\boldsymbol{\lambda}_j^{(k+1)} = \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_i \mathbf{A}_{ij}} \sum_i \mathbf{A}_{ij} \frac{\mathbf{y}_i}{\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)} + \bar{\mathbf{b}}_i}. \tag{13.67}$$

This is the well known MLEM algorithm for emission tomography.

One can show that this recipe has the wonderful feature that each new EM-iteration increases the value of the likelihood L_{ML}. Note that the complete data $\mathbf{x}_{ij}$ do not appear in (13.67); they are needed in the derivation but they don't need to be computed explicitly. This is very fortunate, because their number is huge.

An initial image $\boldsymbol{\lambda}^{(1)}$ is required to start the iterations. Because experience (and theoretical analysis) has shown that higher spatial frequencies have slower convergence, and because smooth images are preferred, the initial image is usually chosen to be uniform, by setting $\boldsymbol{\lambda}_j^{(1)} = C, j = 1 \dots J$, where C is a strictly positive constant.

The MLEM algorithm is multiplicative, implying that it cannot change the value of a reconstruction voxel, when the current value is zero. For that reason, the voxels in the initial image should only be set to zero if it is known a-priori that they are indeed zero. The derivation of the MLEM algorithm uses the assumption that all $\mathbf{y}_i$, all $\mathbf{x}_{ij}$ and all $\boldsymbol{\lambda}_j$ are nonnegative. Assuming that $\mathbf{y}_i \geq 0, i = 1 \dots I$, and considering that the probabilities $\mathbf{A}_{ij}$ are also nonnegative, it is clear that when the initial image $\boldsymbol{\lambda}^{(1)}$ is nonnegative, all subsequent images $\boldsymbol{\lambda}^{(k)}$ will be nonnegative as well. However, when for some reason a reconstruction value becomes negative (e.g. because one or a few sinogram values $\mathbf{y}_i$ are negative), then convergence is no longer guaranteed. In practice, divergence is almost guaranteed in that case. Consequently, if the sinogram is preprocessed with a procedure that may produce negatives (e.g. randoms subtraction in PET), MLEM reconstruction will only work if all negative values are set to a nonnegative value.

13.3.3.2 Reconstruction from list-mode data

The measured data $\mathbf{y}_i$ considered in the derivations above (so-called *binned* data) represent the number of counts acquired within an individual crystal pair i (LOR i), that is, $\mathbf{y}_i$ represents the sum of those acquired events (indexed by m) that were assigned (histogrammed) to the i-th LOR: $\mathbf{y}_i = \sum_{m \in i} 1$. However, in modern PET systems the number of possible LORs within the FOV typically exceeds (often by many times) the number of events acquired in a clinical PET study. Consequently, the binned data are very sparse and it is more efficient to store and process each acquired event (with all its relevant information) separately, in the so called *list-mode* format.

Modification of the maximum-likelihood algorithms is straightforward (whether ML-EM, or accelerated algorithms based on ordered subsets discussed in a later subsection), as shown in works by Barrett and by Reader. Note that the same is not true about other algorithms, for example algorithms with additive updates. The ML-EM algorithm for the list-mode data can be obtained by replacing the $\mathbf{y}_i$ in the ML-EM equation (13.67) by the above mentioned sum over events, skipping the LORs with zero counts (which do not contribute to the ML-EM sum), and combining the sum over LORs i with the sum over events m:

$$\boldsymbol{\lambda}_j^{(k+1)} = \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_{i \in LORs} \mathbf{A}_{ij}} \sum_{m \in event\_list} \mathbf{A}_{i_m j} \frac{1}{\sum_j \mathbf{A}_{i_m j} \boldsymbol{\lambda}_j^{(k)} + \bar{\mathbf{b}}_{i_m}}, \tag{13.68}$$

where i_m represents the LOR index in which the m-th event has been recorded. The main difference is that the ML-EM sum is now evaluated (including calculations of the relevant forward and back-projections) only over the list of the available events (in any order). However, it is important to mention here that the normalizing term in front of the sum (sensitivity matrix $\sum_i \mathbf{A}_{ij}$) still has to be calculated over all possible LORs, and not only those with non-zero counts. This represents a challenge for the attenuated data (attenuation considered as part of the system matrix $\mathbf{A}$), since the

sensitivity matrix has to be calculated specifically for each object, and therefore it cannot be pre-computed. For modern systems with a large number of LORs, its calculation often takes more time than the list-mode reconstruction itself. For this reason, alternative approaches (involving certain approximations) have been considered for the calculation of the sensitivity matrix, such as sub-sampling approaches [21] or Fourier-based approaches [22].

13.3.3.3 Reconstruction of TOF-PET data

In the time-of-flight case, the probability of a pair of photons arriving from a particular point along the LOR (as reported based on the difference of their detection times) is given by a Gaussian kernel having a width determined by the timing uncertainty of the detection system. In contrast, in the non-TOF case the probability of detecting the event is approximately uniform along the LOR. Modification of iterative reconstruction algorithms (whether for binned or list-mode data) to account for the TOF is straightforward. One just needs to replace integrations along the LORs (the main component of the system matrix $\mathbf{A}$) with the TOF-kernel weighted integrations along the LORs. The forward-projection (or back-projection) in a certain direction can now be viewed, and performed, as a convolution of the image with a proper TOF kernel in the LOR direction (see Figure 13.7). The rest of the algorithm, i.e., formulae derived in the previous subsections, stays exactly the same (only the form of the system matrix $\mathbf{A}$ is changed). Additional information provided by the TOF measurements, leading to more localized data, results in faster, and more uniform, convergence, as well as in improved signal-to-noise ratios in reconstructed images, as widely reported in the literature.

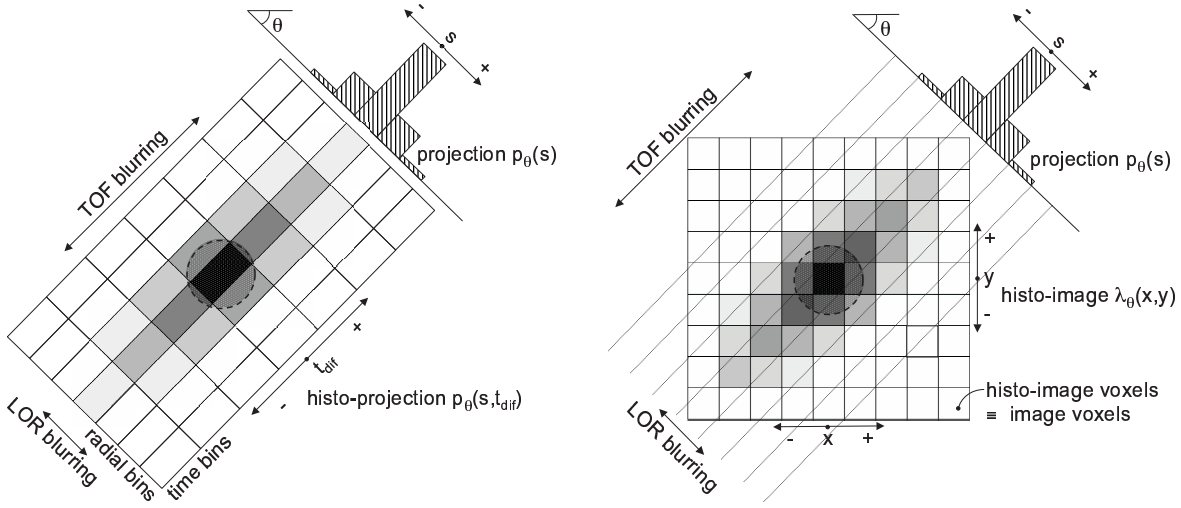

Figure 13.13: *Comparison of the data formats for binned TOF data (left: histo-projection for* $45°$ *view) and for DIRECT approach (right: histo-image for* $45°$ *view).* Histo-projections *can be viewed as an extension of individual non-TOF projections into TOF directions (time bins), and their sampling intervals relate to the projection geometry and timing resolution.* Histo-images *are defined by the geometry and desired sampling of the reconstructed image. Acquired events and correction factors are directly placed into the image resolution elements of individual histo-images (one histo-image per view) having a one-to-one correspondence with the reconstructed image voxels.*

The TOF mode of operation has some practical consequences (and novel possibilities) on the ways the acquired data are stored and processed. The *list-mode* format is very similar to the non-TOF

case. The event structure is just slightly expanded by a few bits (5-8 bits/event) to include the TOF information, and the events are processed event-by-event as in the non-TOF case.

On the other hand, the *binned* data undergoes considerable expansion when accommodating the TOF information. Namely, the projections (x-ray transform) structures are expanded by one dimension, that is, each projection bin is expanded in the LOR direction into the set of time bins forming the so called *histo-projections* (see Figure 13.13-left). In practice, the effect of this expansion on the data size is not as bad as it appears, because the localized nature of TOF data allows decreased angular sampling (typically about 5-10-times) in both azimuthal and co-polar directions (views) while still satisfying angular sampling requirements. The resulting data size thus remains fairly comparable to the non-TOF case. During the reconstruction process the histo-projection data are processed time-bin by time-bin (instead of projection line by line in the non-TOF case). Note, that there exist also hybrid approaches between the two above, in which the data are binned in the LOR space, but events are stored in list-mode for each LOR bin.

TOF allows also a conceptually different approach of data partitioning, leading to more efficient reconstruction implementations, by using DIRECT (Direct Image Reconstruction for TOF) approach utilizing the so-called *histo-images* (see Figure 13.13-right) [22]. In the DIRECT approach the data are directly histogrammed (deposited), for each view, into the image resolution elements (voxels) of desired size. Similarly, all correction arrays and data are estimated or calculated in the same histo-image format. The fact that all data and image structures are now in image arrays (of the same geometry and size) makes possible very efficient computer implementations of the data processing and reconstruction operations.

13.3.3.4 Reconstruction of dynamic data

Under dynamic data we understand data acquired from an object dynamically changing with time in activity distribution, or in morphology (shape), or in both. An example of the first case would be a study looking at temporal changes in activity uptake in individual organs or tissues, so called Time Activity Curves (TAC). An example of the second case would be a gated cardiac study providing information about changes of the heart morphology during the heart beat cycle (such as changes of the heart wall thickness or movements of the heart structures).

The dynamic data can be viewed as an expansion of static (3D) data by the temporal information into 4D (or 5D) data. The dynamic data are usually subdivided (spread) into a set of temporal (time) frames. In the first application, each time frame represents data acquired within a certain sequential time sub-interval of the total acquisition time. The sub-intervals can be uniform, or non-uniform with their durations adjusted, for example, to the speed of the change of the activity curves. In the second application, each time frame represents the total counts acquired within a certain stage (gate) of the periodic organ movement (e.g., gated based on the EKG signal). In the following we address issues of the reconstruction of dynamic data in general. Problems related to the motion and its corrections will be discussed in the subsection on motion correction.

Once the data are subdivided (during acquisition) or sorted (acquired list-mode data) into the set of time frames, seemingly the most natural way is to reconstruct each time frame separately. Note that this is the only available option for the analytical reconstruction approaches, while the iterative reconstruction techniques can reconstruct the dynamic data also directly in 4D (or 5D). A problem with the frame-by-frame reconstruction is that data in the individual time frames are quite noisy, since each time frame has only a fraction of the total acquired counts, leading to noisy reconstructions. Consequently, the resulting reconstructions have often to be filtered in the spatial and/or temporal directions to get images of any practical value. Temporal filtering takes into account time correlations

between the signal components in the neighboring time frames, while the noise is considered to be independent. Filtering, however, leads to resolution versus noise trade-offs.

On the other hand, reconstructing the whole 4D (or 5D) data set together, while using this correlation information in the (4D) reconstruction process via proper temporal (resolution) kernels or basis functions, can considerably improve those trade-offs as reported in the literature (similarly to the case of the spatial resolution modeling). The temporal kernels (basis functions) can be uniform in shape and distribution, or can have a non-uniform shape (for example taking into account expected or actual shape of the TAC curves) and can be distributed on a non-uniform grid (for example, reflecting count levels at individual frames or image locations). The kernel shapes and distributions can be defined, or determined, beforehand and be fixed during the reconstruction. That is, during the reconstruction process we are reconstructing just the amplitudes of the basis functions. The algorithms derived in the previous subsections stay basically the same, where the temporal kernels can be considered as part of the system matrix $\mathbf{A}$ (comparable to the including of the TOF-kernel in TOF-PET). Another approach, more accurate but mathematically and computationally much more involved, is to iteratively build-up the shape (and distribution) of the temporal kernels during the reconstruction in conjunction with the reconstruction of the emission activity (that is, the amplitude of the basis functions).

While iterative methods lead to a clear quality improvement when reconstructing dynamic data, thanks to the more accurate models of the signal and data noise components, for the quantitative dynamic studies their shortcoming is their non-linear behavior, especially if they are not fully converged. For example, the local bias levels can vary across the time frames as the counts, local activity levels, and object morphology changes, which can lead to less accurate TACs. On the other hand, analytic techniques which are linear and consequently do not depend on the count levels and local activity, might provide a more consistent (accurate) behavior across the time frames in the mean (less bias of the mean), but much less consistent (less precise) behavior in the variance due to the largely increased noise. It is still an open issue which of the two approaches provides more clinically useful results, and the discussions and research on this topic are still open and ongoing.

13.3.4 Acceleration

13.3.4.1 Ordered subsets expectation maximisation (OSEM)

The MLEM algorithm requires a projection and a backprojection in every iteration, which are operations involving a huge amount of computations. Typically MLEM needs several tens to hundreds of iterations for good convergence. Consequently, MLEM reconstruction is slow and many researchers have studied methods to accelerate convergence.

The method most widely used is ordered subsets expectation maximisation (OSEM) [23]. The MLEM algorithm (13.67) is rewritten here for convenience:

$$\hat{\mathbf{y}}_i^{(k)} = \sum_j \mathbf{A}_{ij} \boldsymbol{\lambda}_j^{(k)} + \bar{\mathbf{b}}_i \tag{13.69}$$

$$\boldsymbol{\lambda}_j^{(k+1)} = \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_i \mathbf{A}_{ij}} \sum_i \mathbf{A}_{ij} \frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i^{(k)}}, \tag{13.70}$$

where k is the iteration number, and $\boldsymbol{\lambda}^{(1)}$ is typically set to a uniform, strictly positive image. In OSEM, the set of all projections $\{1 \ldots \mathrm{I}\}$ is divided in a series of subsets $S_t, t = 1 \ldots T$. Usually, these subsets are exhaustive and non-overlapping, i.e. every projection element i belongs to exactly one subset S_t. In SPECT and PET the data $\mathbf{y}$ are usually organized as a set of (parallel or fanbeam)

projections, indexed by projection angle ϕ. Therefore, the easiest way to produce subsets of $\mathbf{y}$ is by assigning all the data for each projection angle to exactly one of the subsets. However, if the data $\mathbf{y}$ are stored in list mode (see section 13.3.3.2), the easiest way is to simply cut the list in blocks, assigning each block to a different subset.

The OSEM algorithm can then be written as

$$\begin{aligned}
&\text{initialize } \boldsymbol{\lambda}_j^{\text{old}}, j = 1 \dots J \\
&\text{for } k = 1 \dots K \\
&\quad \text{for } t = 1 \dots T \\
&\qquad \hat{\mathbf{y}}_i = \sum_j \mathbf{A}_{ij} \boldsymbol{\lambda}_j^{\text{old}} + \bar{\mathbf{b}}_i, \quad i \in S_t \\
&\qquad \text{for } j = 1 \dots J \\
&\qquad\quad \boldsymbol{\lambda}_j^{\text{new}} = \frac{\boldsymbol{\lambda}_j^{\text{old}}}{\sum_{i \in S_t} \mathbf{A}_{ij}} \sum_{i \in S_t} \mathbf{A}_{ij} \frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i}.
\end{aligned} \tag{13.71}$$

If all projections are combined in a single subset, the OSEM algorithm is identical to the MLEM algorithm. Otherwise, a single OSEM iteration k consists of T subiterations, where each subiteration is similar to an MLEM iteration, except that the projection and backprojection are only done for the projections of the subset S_t. If every sinogram pixel i is in exactly one subset, the computation burden of a single OSEM iteration is similar to that of an MLEM iteration. However, MLEM would update the image only once, while OSEM updates it T times. Experience shows that this improves convergence by a factor of about T, which is very significant.

Convergence is only guaranteed for consistent data, and provided that there is subset balance, which requires

$$\sum_{i \in S_t} \mathbf{A}_{ij} = \sum_{i \in S_u} \mathbf{A}_{ij}, \tag{13.72}$$

where S_t and S_u are different subsets. In practice, these conditions are never satisfied, and OSEM can be shown to converge to a limit cycle rather than to a unique solution, with the result that the OSEM reconstruction is noisier than the corresponding MLEM reconstruction. However, in many applications, the difference between the two is not clinically relevant. The procedure is illustrated with a simple simulation in fig 13.14. Because there was no noise and no attenuation, convergence of OSEM is guaranteed in this example. In more realistic cases, it may be recommended to have 4 or more projections in a single subset, to prevent excessive noise amplification at higher iteration numbers.

13.3.4.2 Refinements of the OSEM-algorithm

As mentioned above, OSEM converges to a limit cycle: after many iterations, it starts cycling through a series of solutions rather than converging to the maximum-likelihood solution. When compared to the initial image (usually a uniform image), these series of solutions are “relatively close” to the maximum likelihood solution. Consequently, the convergence of OSEM is initially much faster but otherwise similar to that of MLEM; the better performance of MLEM only becomes noticeable at high iteration numbers. Thus, a simple solution to avoid the limit cycle is to gradually decrease the number of subsets: this approach preserves the initial fast convergence of OSEM, avoiding the limit cycle by returning to MLEM at high iteration numbers. A drawback of this approach is that convergence

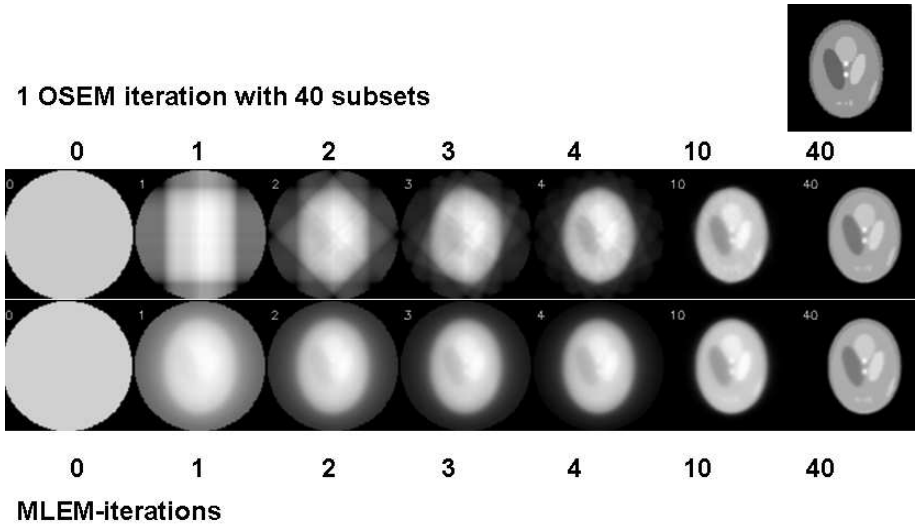

Figure 13.14: *A simulation comparing a single OSEM iteration with 40 subsets, to 40 MLEM iterations. The computation time of the MLEM-reconstruction is about 40 times longer than that of OSEM. In this example there were only 2 (parallel beam) projection angles per subset, which is clearly visible in the first OSEM iteration.*

becomes slower each time the number of subsets is reduced. In addition, there is no theory available that prescribes how many subiterations should be used for each OSEM-iteration.

Many algorithms have been proposed that use some form of relaxation to obtain convergence under less restrictive conditions than those of OSEM. As an example, relaxation can be introduced by rewriting the OSEM expression (13.71) in an additive way. Then, a relaxation factor α is inserted to scale the update term to obtain RAMLA (row-action maximum likelihood algorithm [24]):

$$\boldsymbol{\lambda}_j^{\text{new}} = \boldsymbol{\lambda}_j^{\text{old}} + \alpha \boldsymbol{\lambda}_j^{\text{old}} \sum_{i \in S_t} \mathbf{A}_{ij} \left(\frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i} - 1 \right) \quad \text{with} \quad \alpha < \frac{1}{\max_t \left(\sum_{i \in S_t} \mathbf{A}_{ij} \right)} \tag{13.73}$$

The relaxation factor α decreases with increasing iteration numbers to ensure convergence. Note that setting $\alpha = 1/\sum_{i \in S_t} A_{ij}$ for all (sub-) iterations yields OSEM. Several alternative convergent block iterative algorithms have been proposed. They are typically much faster than MLEM but slightly slower than the (non-convergent) OSEM algorithm.

13.3.5 Regularisation

MLEM maximizes the likelihood, by making the computed projections (from the current reconstruction) as similar as possible to the measured projections, where the similarity is measured based on the Poisson distribution. An upper limit of the likelihood would be obtained when the measured and calculated projections are identical. However, this is never possible, because Poisson noise introduces inconsistencies. Nevertheless, a large part of the noise is consistent, which means that it can obtained as the projection of a (noisy) activity distribution. This part of the noise propagates into the reconstructed image, and is responsible for the so-called “deterioriation” of the MLEM image at high iterations.

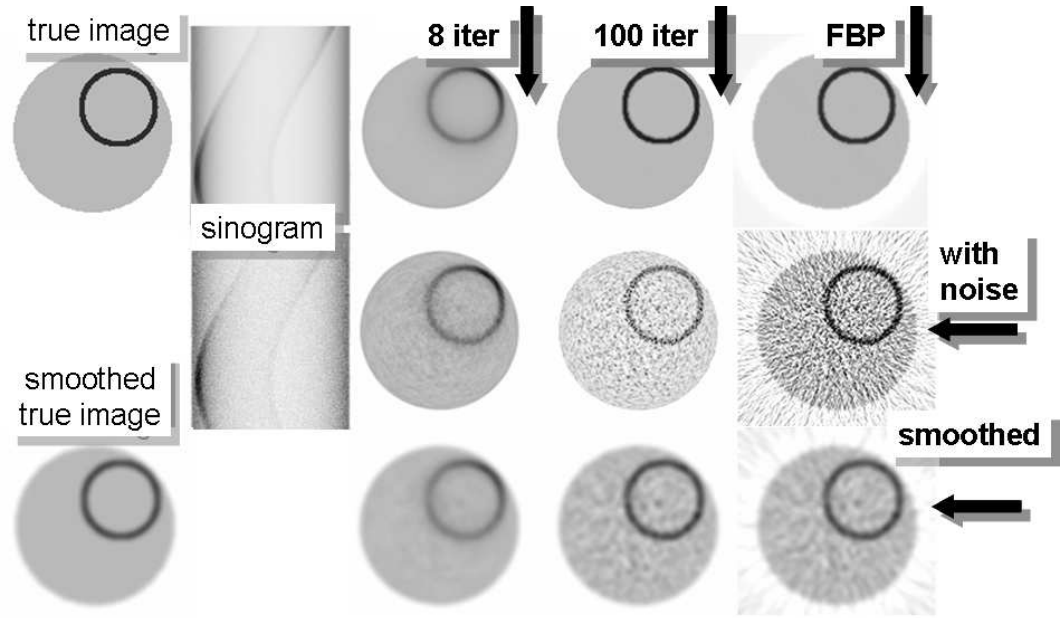

Figure 13.15: *Simulation study illustrating position dependent convergence in PET with attenuation. After 8 iterations, convergence in highly attenuated regions is poor. After 100 iterations, good convergence is obtained, but with strong noise propagation. Post-smoothing yields a fair compromise between noise and nearly position independent resolution.*

13.3.5.1 Stopping iterations early

An "accidental" feature of the MLEM algorithm is its frequency dependent convergence: low spatial frequencies converge faster than higher frequencies. This is due to the low pass effect of the back-projection operation. This effect is easily verified for the reconstruction of the activity in a point source, if the MLEM reconstruction is started from a uniform image. The first iteration then yields the backprojection of the point source measurement. As discussed in section 13.2.1.1, this yields an image with intensity $\boldsymbol{\lambda}(x, y) \sim 1/\sqrt{x^2 + y^2}$, if the point source was located at $(0, 0)$. Each iteration multiplies with a similar backprojection, implying that after t iterations, the image intensity at (x, y) is proportional to $1/(x^2 + y^2)^{t/2}$, so the peak at $(0, 0)$ becomes a bit sharper with every iteration. For more complicated objects, the evolution is more subtle.

It follows that reducing the number of iterations has an effect which is similar to reducing the cut-off frequency of a low pass filter. However, the effect on the resolution is position dependent, as illustrated in fig 13.15. Attenuated PET projections of a highly radioactive uniform ring inside a less active disk were simulated with and without Poisson noise. After 8 MLEM iterations, the reconstructed ring has non-uniform activity. In the center of the phantom, convergence is slower, resulting in poorer resolution and poorer recovery of the activity in the ring. After 100 iteration, convergence is much better everywhere in the phantom, but for noisy data, there is a very disturbing noise propagation.

If the image was acquired for detection (e.g. to see if there is a radioactive ring inside the disk or not), then the image produced at 8 iterations is excellent. However, if the aim is quantification (e.g. analyze the activity distribution along the ring), then quantification errors can be expected at low iteration numbers.

13.3.5.2 Post-smoothed ML

The noise in the higher MLEM iterations is high frequency noise, and there are strong negative correlations between neighboring pixels. As a result, a modest amount of smoothing strongly suppresses the noise at the cost of a mild loss of resolution. This is illustrated in the third row of fig 13.15.

If the MLEM implementation takes into account the (possibly position dependent) spatial resolution effects, then the resolution should improve with every MLEM iteration. After many iterations, the spatial resolution should be rather good, similar or even better than the sinogram resolution, but the noise will have propagated dramatically. Let us assume that the obtained spatial resolution corresponds to a position dependent point spread function which can be approximated as a Gaussian with a full width at half maximum of $F_{ml}(x, y)$. Assume further that this image is post-smoothed with a (position independent) Gaussian convolution kernel with a FWHM of F_p. The local point spread function in the smoothed image will then have a FWHM of $\sqrt{(F_{ml}(x, y))^2 + F_p^2}$. If enough iterations are applied and if the post-smoothing kernel is sufficiently wide, we have $F_p >> F_{ml}(x, y)$ and therefore $\sqrt{(F_{ml}(x, y))^2 + F_p^2} \simeq F_p$. Under these conditions, the post-smoothed MLEM image has a nearly position independent and predictable spatial resolution. Therefore, if PET or SPECT images are acquired for quantification, it is recommended to use many iterations and post-smoothing, rather than a reduced number of iterations, for noise suppression.

13.3.5.3 Smoothing basis functions

An alternative approach to counter noise propagation is to use an image representation that does not accomodate noisy images. Instead of representing the image with a grid of non-overlapping pixels, a grid of smooth, overlapping basis functions can be used. The two mostly used approaches are the use of spherical basis functions or “blobs” [25] and the use of Gaussian basis functions or sieves [26].

In the first approach, the projector and backprojector operators are typically adapted to work directly with line integrals of the basis functions. In the sieves approach, the projection of a Gaussian blob is usually modeled as the combination of a Gaussian convolution and projection along lines. The former approach produces a better approximation of the mathematics, the latter approach yields a faster implementation.

The blobs or sieves are probably most effective when their width is very similar to the spatial resolution of the tomographic system. In this setting, the basis function allows accurate representation of the data measured by the tomographic system, and prevents reconstruction of much of the (high frequency) noise. It has been shown that using the blob during reconstruction is more effective than using the same blob only as a post-smoothing filter. The reason is that the post-filter always reduces the spatial resolution, while a sufficiently small blob does not smooth data if it is used during reconstruction.

If the blob or sieve is wider than the spatial resolution of the tomographic system, then its use during reconstruction produces Gibbs over- and undershoots, also known as “ringing”. This effect always arises when steep edges have to be represented with a limited frequency range, and is related to the ringing effects observed with very sharp low pass filters. For some imaging tasks, these ringing artifacts are a disadvantage.

13.3.5.4 Maximum-a-posteriori or penalized likelihood

Smoothing the MLEM image is not a very elegant approach: first the likelihood is maximized, and then it is decreased again by smoothing the image. It seems more elegant to modify the objective

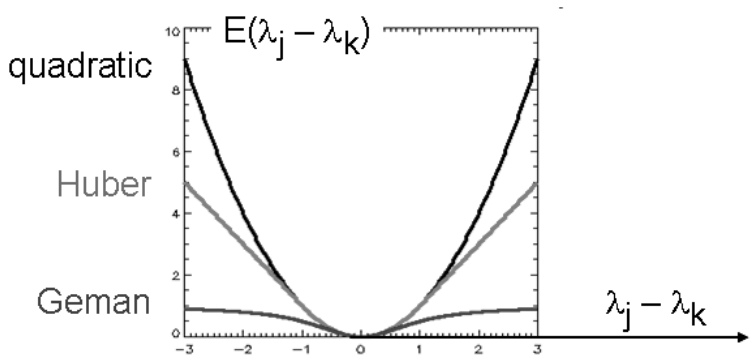

Figure 13.16: *The energy function of the quadratic prior, the Huber prior and the Geman prior.*

function, such that the image that maximizes it does not need further processing. This can be done with a Bayesian approach, which is equivalent to combining the likelihood with a penalty function.

It is assumed that a good reconstruction image $\boldsymbol{\lambda}$ will be obtained, if that image maximizes the (logarithm of the) probability $p(\boldsymbol{\lambda}|\mathbf{y})$, given by equation (13.39) and repeated here for convenience:

$$\hat{\boldsymbol{\lambda}} = \arg\max_{\lambda} \left(\ln p(\mathbf{y}|\boldsymbol{\lambda}) + \ln p(\boldsymbol{\lambda})\right) \tag{13.74}$$

The second term represents the a-priori knowledge about the tracer distribution, and it can be used to express our belief that the true tracer distribution is fairly smooth. This is usually done with a Markov prior. In a Markov prior, the a priori probability for a particular voxel, given the value of all other voxels, is only a function of the direct neighbors of that voxel:

$$p(\boldsymbol{\lambda}_j|\boldsymbol{\lambda}_k, \forall k \neq j) = p(\boldsymbol{\lambda}_j|\boldsymbol{\lambda}_k, k \in \mathbf{N}_j), \tag{13.75}$$

where $\mathbf{N}_j$ denotes the set of neighbor voxels of j. Such priors are usually written in the following form:

$$P(\boldsymbol{\lambda}) = \ln p(\boldsymbol{\lambda}) = \sum_j \ln p(\boldsymbol{\lambda}_j|\boldsymbol{\lambda}_k, k \in \mathbf{N}_j) = -\beta \sum_j \sum_{k \in N_j} E(\boldsymbol{\lambda_j}, \boldsymbol{\lambda_k}), \tag{13.76}$$

where the "energy" function E is designed to obtain the desired noise suppressing behaviour. The parameter β is the weight assigned to the prior, a higher weight results in smoother images, at the cost of a decreased likelihood, i.e. poorer agreement with the acquired data. In most priors, the expression is further simplified by making E a function of a single variable, the absolute value of the difference $|\boldsymbol{\lambda_j} - \boldsymbol{\lambda_k}|$.

Some popular energy functions $E(|\boldsymbol{\lambda_j} - \boldsymbol{\lambda_k}|)$ are shown in fig. 13.16. A simple and effective one is the quadratic prior $E(x) = x^2$; a MAP-reconstruction with this prior is shown in fig 13.17. Better preservation of strong edges is obtained with the Huber prior: it is quadratic for $|\boldsymbol{\lambda_j} - \boldsymbol{\lambda_k}| \leq \delta$, and linear for $|\boldsymbol{\lambda_j} - \boldsymbol{\lambda_k}| > \delta$, with a continuous first derivative at δ. Consequently, it applies less smoothing than the quadratic prior for differences larger than δ, as illustrated in fig 13.17. Even stronger edge tolerance is obtained with the Geman prior, which converges asymptotically to a constant for large differences, implying that it does not smooth at all over very large pixel differences.

One can show that the prior (13.76) is concave function of $\boldsymbol{\lambda}$ if $E|\boldsymbol{\lambda_j} - \boldsymbol{\lambda_k}|$ is a convex function. Consequently, the quadratic and Huber energy functions yield a concave prior: it has a single maximum. In contrast, the Geman prior is not concave (see fig 13.16) and has local maxima. Such concave

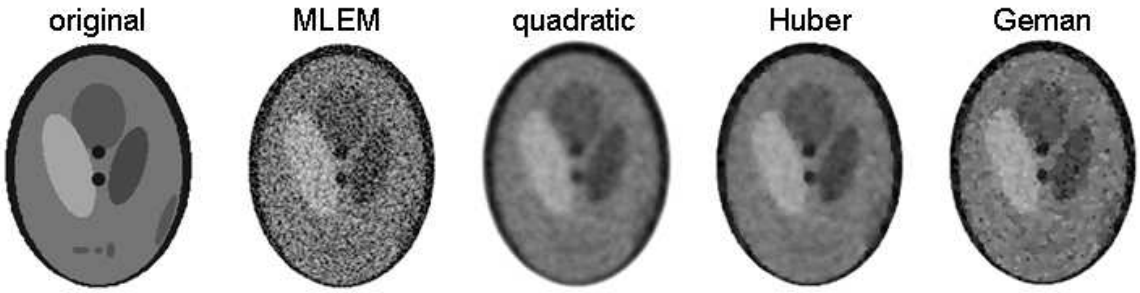

Figure 13.17: *MLEM and MAP reconstructions of the Shepp-Logan phantom. Three different smoothing priors were used: quadratic, Huber and Geman. The latter smooth small differences quadratically, but are more tolerant for large edges.*

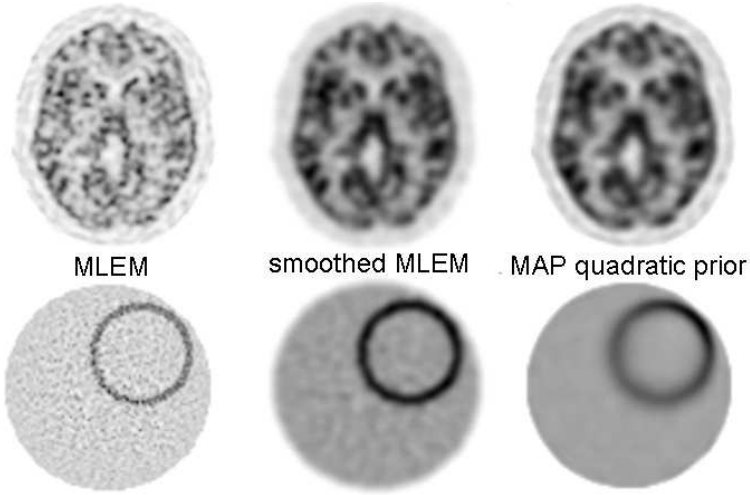

Figure 13.18: *MLEM, smoothed MLEM and MAP (quadratic prior) reconstructions of simulated PET data of a brain and a ring phantom. The ring phantom reveals position dependent smoothing for MAP.*

priors require careful initialisation, because the final reconstruction depends on the initial image and on the behaviour of the optimisation algorithm.

Fig 13.18 shows that MAP-reconstructions produce position dependent spatial resolution, similar to MLEM with reduced number of iterations. The reason is that the prior is applied with a uniform weight, whereas the likelihood provides more information about some voxels than about others. As a result, the prior produces more smoothing in regions where the likelihood is "weaker", e.g. regions that have contributed only few photons to the measurement due to high attenuation.

The prior can be made position dependent as well, to ensure that the balance between the likelihood and the prior is about the same in the entire image. In that case, MAP with quadratic prior produces images which are very similar to MLEM images with post-smoothing: if the prior and smoothing are tuned to produce the same spatial resolution, then both algorithms also produce nearly identical noise characteristics.

Many papers have been devoted to the development of algorithms for MAP-reconstruction. A popular algorithm is the so-called one step late (OSL) algorithm. Inserting the derivative of the prior

P in equation (13.66) yields:

$$\frac{\partial(L_x(\boldsymbol{\lambda}) + P(\boldsymbol{\lambda}))}{\partial \boldsymbol{\lambda}_j} = \sum_i \left(\frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i^{(k)}} \mathbf{A}_{ij} \boldsymbol{\lambda}_j^{(k)} \frac{1}{\lambda_j} - \mathbf{A}_{ij} \right) + \frac{\partial P(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} = 0 \tag{13.77}$$

where $\hat{\mathbf{y}}_i^{(k)}$ is the projection of the current reconstruction for detector i. A problem with this equation is that $\partial P(\boldsymbol{\lambda})/\partial \boldsymbol{\lambda}_j$ is itself a function of the unknown image $\boldsymbol{\lambda}$. To avoid this problem, the derivative of the prior is simply evaluated in the current reconstruction $\boldsymbol{\lambda}^{(k)}$. The equation can then be solved to produce the MAP update expression:

$$\boldsymbol{\lambda}_j^{(k+1)} = \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_i \mathbf{A}_{ij} - \left. \frac{\partial P(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} \right|_{\boldsymbol{\lambda}^{(k)}}} \sum_i \mathbf{A}_{ij} \frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i^{(k)}} \tag{13.78}$$

Because of the approximation, convergence is not guaranteed. The algorithm usually works fine, except with very high values for the prior.

The MLEM algorithm can be considered as a gradient ascent algorithm (see also eq (13.50)):

$$\boldsymbol{\lambda}_j^{(k+1)} = \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_i \mathbf{A}_{ij}} \sum_i \mathbf{A}_{ij} \frac{\mathbf{y}_i}{\hat{\mathbf{y}}_i^{(k)}} \tag{13.79}$$

$$= \boldsymbol{\lambda}_j^{(k)} + \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_i \mathbf{A}_{ij}} \left. \frac{\partial L_{ML}(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} \right|_{\boldsymbol{\lambda}^{(k)}}. \tag{13.80}$$

Extensions to a MAP gradient ascent algorithm then typically have a form like

$$\boldsymbol{\lambda}_j^{(k+1)} = \boldsymbol{\lambda}_j^{(k)} + S(\boldsymbol{\lambda}^{(k)}) \left. \frac{\partial L_{ML}(\boldsymbol{\lambda}) + \partial P(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}_j} \right|_{\boldsymbol{\lambda}^{(k)}}, \tag{13.81}$$

where the key is to determine a good preconditioner S. Several methods with (almost) guaranteed convergence have been based on the previously described optimisation transfer method, by designing useful surrogate functions for both the likelihood and the prior.

13.3.6 Corrections

In typical emission data the true events (having a Poisson character) are distorted and contaminated by a number of physical effects. To make the best use of the acquired data and of our knowledge of the acquisition system, these effects should be included in the reconstruction model. The distortion effects include resolution effects (such as detector resolution, collimator effects, and in PET also non-colinearity and positron range) and motion effects. The contamination effects can be divided, by their character and the way they are treated, into multiplicative and additive terms. The multiplicative factors include: attenuation of the annihilation photons by the object, the probability of the detector elements detecting an event once they are hit by the photon (detector normalization factors), coefficients accounting for the decay time, and the geometrical restriction of directions/LORs for which true events are detected (axial acceptance angle, detector gaps). The additive terms include scattered and random (in the PET case) coincidences. Details on calculation of the correction factors and terms are discussed in other chapters. This chapter is limited to the discussion of their utilization within the reconstruction process.

The most straightforward approach is to pre-correct the data before reconstruction for the contamination effects (multiplying by multiplicative correction coefficients and subtracting the scatter and random estimates), so as to approximate the X-ray transform (or attenuated X-ray transform in the SPECT case) of the reconstructed object. For analytical reconstruction approaches (derived for the ideal X-ray transform data) the data have always to be pre-corrected.

For the statistical reconstruction methods, derived based on the statistical properties of the data, an attempt is made to preserve the Poisson character of the data as much as possible by including the correction effects inside the reconstruction model. Theoretically, the most appropriate way is to include the multiplicative and scatter effects directly into the system matrix. The system matrix would have to include not only an accurate model of the direct data (true events) but also of the physical processes of the generation of the contamination scatter data. In a sense the contamination would then become valid data, bringing extra information to our model and thus adding valid (properly modeled) counts to the image. However, inclusion of the scatter model into the system matrix tremendously increases the number of non-zero elements of the system matrix, i.e. the matrix is not sparse anymore, and consequently the system is more ill-posed (the contamination data are typically quite noisy) and computationally exceedingly expensive, and thus not feasible for routine clinical use.

The more practical, and commonly used, approach is to include correction effects as multiplicative factors and additive terms within the forward projection model of the iterative reconstruction approaches:

$$\mathbf{y} = \mathbf{A}\boldsymbol{\lambda} + \mathbf{b}, \tag{13.82}$$

where the effects directly influencing the direct (true) data are included inside the system matrix $\mathbf{A}$ and will be discussed in the following, while the additive terms $\mathbf{b}$ (including scatter and randoms) will be discussed separately in the subsection on the additive terms.

13.3.6.1 Factors affecting direct events - multiplicative effects

In the PET case the sequence of the physical effects (described in previous chapters) that occur as the true coincident events are generated and detected can be described by the following factorization of the system matrix $\mathbf{A}$ as discussed in detail in the review paper [27]:

$$\mathbf{A} = \mathbf{A}_{det.sens}\mathbf{A}_{det.blur}\mathbf{A}_{att}\mathbf{A}_{geom}\mathbf{A}_{tof}\mathbf{A}_{positron} \tag{13.83}$$

where $\mathbf{A}_{positron}$ models the positron range, $\mathbf{A}_{tof}$ models the timing accuracy for the TOF PET systems (TOF resolution effects, as discussed in the sub-section on the TOF iterative reconstruction), $\mathbf{A}_{geom}$ is the geometric projection matrix, the core of the system matrix, which is a geometrical mapping between the source (voxel j) and data (projection bin i, defined by the LOR, or its time bin in the TOF case); the geometrical mapping is based on the probability (in the absence of attenuation) that photon pairs emitted from individual image location (voxel) reach the front faces of given crystal pair (LOR), $\mathbf{A}_{att}$ is a diagonal matrix containing attenuation factors on individual LORs, $\mathbf{A}_{det.blur}$ models the accuracy of reporting the true LOR positions (detector resolution effects, discussed in the subsection on the spatial resolution effects), and $\mathbf{A}_{det.sens}$ is a diagonal matrix modeling the probability that an event will be reported once the photon pair reaches the detector surface - a unique multiplicative factor for each detector crystal pair (LOR) modeled by normalization coefficients, but can include also the detector axial extent and detector gaps.

In practice, the attenuation operation $\mathbf{A}_{att}$ is usually moved to the left (to be performed after the blurring operation). This is strictly correct only if the attenuation factors change slowly, i.e., they do not change within the range of detector resolution kernels. But, even if this is not the case, a good

approximation can be obtained by using blurred (with the detector resolution kernels) attenuation coefficients. In this case the multiplicative factors $\mathbf{A}_{det.sens}$ and $\mathbf{A}_{att}$ can be removed from the system matrix $\mathbf{A}$ and applied only after the forward projection operation as a simple multiplication operation (for each projection bin). The rest of the system matrix (except $\mathbf{A}_{positron}$, which is object dependent) can now be pre-computed, whether in a combined or a factorized form, since it is now independent on the reconstructed object. On the other hand, the attenuation factors $\mathbf{A}_{att}$ (and $\mathbf{A}_{positron}$, if considered) have to be calculated for each given object.

In the SPECT case, the physical effects affecting the true events can be categorized and factorized into the following sequence:

$$\mathbf{A} = \mathbf{A}_{det.sens}\mathbf{A}_{det.blur}\mathbf{A}_{geom,att} \tag{13.84}$$

where $\mathbf{A}_{det.sens}$ includes multiplicative factors (such as detector efficiency and decay time), $\mathbf{A}_{det.blur}$ represents the resolution effects within the gamma camera (the intrinsic resolution of the system), and $\mathbf{A}_{geom,att}$ is the geometric projection matrix including also the collimator effects (such as the depth dependent resolution) and the depth and view dependent attenuation factors.

For the gamma cameras the energy and linearity corrections are usually performed on the fly, and the remaining (detector efficiency) normalization factors are usually very close to one and can be, for all practical purposes, ignored or pre-corrected. Similarly, the theory says that one should do the decay correction during the reconstruction, because it is different for each projection angle. But for most tracers, the decay during the scan is very modest, and in practice it is usually either ignored or done as a pre-correction. The attenuation component is object dependent and it needs to be recalculated for each reconstructed object. Furthermore, its calculation is much more computationally expensive than in the PET case, since it involves separate calculations of the attenuation factors for each voxel and for each view. This is one of the reasons why the attenuation factors have been often ignored in SPECT. More details on the inclusion of the resolution effects into the system matrix will be discussed in the subsection on the resolution effects.

13.3.6.2 Additive contributions

The main additive contaminations are scatter (SPECT and PET) and random events (PET). The simplest possibility of dealing with them is to subtract their estimates ($\bar{\mathbf{s}}$ and $\bar{\mathbf{r}}$) from the acquired data. While this is a valid (and necessary) pre-correction step for the analytic reconstructions, it is not recommended for statistical approaches since it changes the statistical properties of the data, causing them to lose their Poisson character. Because the maximum likelihood algorithm is designed for Poisson distributed data, its performance is suboptimal if the data noise is different from Poisson. Furthermore, subtraction of the estimated additive terms from the noisy acquired data can introduce negative values into the pre-corrected data, especially for low count studies. The negative values have to be truncated before the maximum likelihood reconstruction, since it is not able to correctly handle the negative data. This truncation however leads to a bias in the reconstruction.

On the other end of the spectrum of possibilities would be considering the scatter and randoms directly in the (full) system model, that is, including a complete physical model of the scatter and random components into a Monte Carlo calculation of the forward projection. However, this approach is exceedingly computationally expensive and it is not feasible for practical use. A practical and the most common approach of dealing with the additive contaminations is to add their estimate ($\bar{\mathbf{b}} = \bar{\mathbf{s}}+\bar{\mathbf{r}}$) to the forward projection in the matrix model of the iterative reconstruction, i.e., the forward model is given by $\mathbf{A}\boldsymbol{\lambda} + \bar{\mathbf{b}}$, as considered in the derivation of the ML-EM reconstruction (13.67).

A special treatment has to be considered for the clinical scanners in which the random events ($\mathbf{r}$, estimated by delayed coincidences) are on-line subtracted from the acquired data ($\mathbf{y}$, events in the coincidence window - prompts). The most important characteristic of the Poisson data is that their mean equals their variance: $mean(\mathbf{y}_i) = var(\mathbf{y}_i)$. However, after the subtraction of the delays from the prompts (both being Poisson variables) the resulting data ($\boldsymbol{\gamma}$) are not Poisson anymore, since $mean(\boldsymbol{\gamma}_i) = mean(\mathbf{y}_i - \mathbf{r}_i) = mean(\mathbf{y}_i) - mean(\mathbf{r}_i)$, while $var(\boldsymbol{\gamma}_i) = var(\mathbf{y}_i - \mathbf{r}_i) = var(\mathbf{y}_i) + var(\mathbf{r}_i)$. To regain the main characteristic of the Poisson data (at least of the first two moments) the *shifted Poisson* approach can be used, utilizing the fact that adding a (noiseless) constant value to the Poisson variable changes the mean but preserves the variance of the result. To modify the mean of the subtracted data $\boldsymbol{\gamma}$ to be equal to their variance (i.e., $var(\mathbf{y}_i) + var(\mathbf{r}_i)$), we need to add to the subtracted data an estimate (of the mean) of the randoms ($\bar{\mathbf{r}}$) multiplied by 2. This gives us $mean(\boldsymbol{\gamma}_i + 2\bar{\mathbf{r}}_i) = mean(\mathbf{y}_i - \mathbf{r}_i + 2\bar{\mathbf{r}}_i) = mean(\mathbf{y}_i) + mean(\mathbf{r}_i)$, which is equal to the $var(\boldsymbol{\gamma}_i + 2\bar{\mathbf{r}}_i) = var(\mathbf{y}_i)$ + var($\mathbf{r}_i$). The ML-EM algorithm using the shifted Poisson model can then be written as:

$$\boldsymbol{\lambda}_j^{(k+1)} = \frac{\boldsymbol{\lambda}_j^{(k)}}{\sum_i \mathbf{A}_{ij}} \sum_i \mathbf{A}_{ij} \frac{\boldsymbol{\gamma}_i + 2\bar{\mathbf{r}}_i}{\sum_j \mathbf{A}_{ij}\boldsymbol{\lambda}_j^{(k)} + \bar{\mathbf{s}}_i + 2\bar{\mathbf{r}}_i}. \tag{13.85}$$

It is worthwhile to mention here that even in the shifted Poisson case, we cannot completely avoid the negative values in the subtracted data and consequent truncation leading to the bias and artifacts. However, the chance of the negative values decreases since the truncation of the negative values is being done on the "value-shifted" data ($\boldsymbol{\gamma}_i + 2\bar{\mathbf{r}}_i$). Examples of reconstructions from data with subtracted additive term, using regular ML-EM algorithm and using ML-EM with shifted Poisson model, are shown in Figure 13.19. Because the counts were relatively high in this simulation, the subtraction did not produce negatives. ML-EM of ($\mathbf{y} - \mathbf{r}$) creates streaks because the reliability of the subtracted data is overestimated.

Note that in the reconstruction model (as well as in the pre-correction approaches) the estimates of the scatter and randoms have to be treated in the same way as the estimates of the true events in the forward projection, including consideration of the normalized or un-normalized events, attenuation corrected or uncorrected data, gaps in the data, etc. Various challenges exist for the scatter and randoms estimations in general, such as modeling of the out-of-FOV scatter, but this is a topic of another chapter.

13.3.6.3 Finite spatial resolution

There are a number of physical and geometrical effects and limitations (such as positron range, acolinearity, depth of interaction, size of detector crystal elements, inter-crystal scatter, collimator geometry, etc.) affecting PET and SPECT resolution as described in more details in other chapters of this book. To get the most out of the acquired data and to correct for the resolution degradation, these effects have to be properly modeled in the system matrix of statistical reconstruction, as considered in the components ($\mathbf{A}_{det.blur}$, $\mathbf{A}_{geom}$, $\mathbf{A}_{positron}$) of the factorized system matrix outlined in subsection 1.3.6.1. This step does not influence the mathematical definition of the reconstruction algorithm (such as ML-EM, as given by equation 13.67); only the form of its system matrix is changed.

However this step has very practical consequences on the complexity of the algorithm implementation, computational demands and most importantly on the quality of the reconstructed images. By including the resolution effects into the reconstruction model, a larger fraction of the data is being used for the reconstruction within each point of the space, with the true signal component becoming more consistent, while the noise components becoming less consistent with the model. Thus the res-

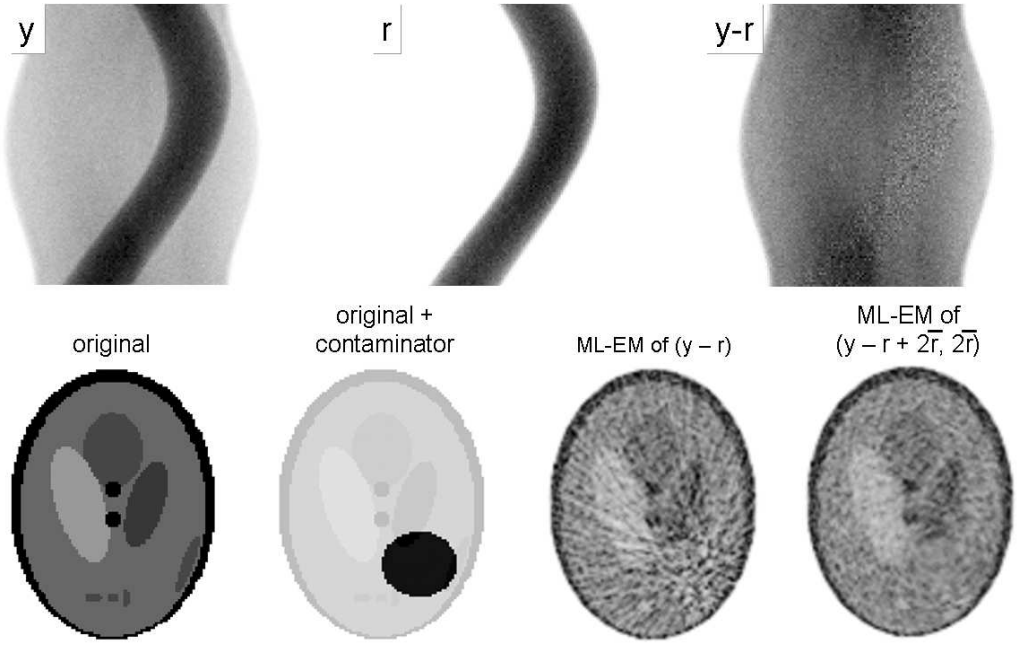

Figure 13.19: *Illustration of (exaggerated case of) reconstructions from contaminated data* $\mathbf{y}$ *from which the additive contamination term* $\mathbf{r}$ *was subtracted (both data and contamination term are Poisson). The top row shows the sinograms. Note the increased noise level in the contaminated area in the sinogram* $\mathbf{y} - \mathbf{r}$*. The bottom row shows the true image without and with the contaminator, the ML-EM reconstruction from the subtracted data (*$\mathbf{y} - \mathbf{r}$*) and the shifted Poisson ML-EM reconstruction, in which the estimated (noiseless) additive term* $2\bar{\mathbf{r}}$ *is added to the subtracted data and forward projection as given by (13.85).*

olution modeling helps us twice, by improving the image resolution while at the same time reducing the image noise, as illustrated in Figure 13.20 for simulated SPECT data. This is quite different from the filtering case, where the noise suppression is always accompanied by resolution deterioration. On the other hand, the resolution modeling costs us a considerable increase of the computational load (both in space/memory and time) since the system matrix is much less sparse, that is, it contains a larger proportion of non-zero elements. This not only leads to more computational load per iteration, but also to a slower convergence of the iterative reconstruction and consequently to the need of more iterations.

Resolution effects can be subdivided into the effects dependent on the particular object, such as the positron range, and the effects influenced by the scanner geometry, design and materials (which can be determined beforehand for the given scanner). The positron range depends on the particular attenuation structures in which the the positrons annihilate, and also varies from isotope to isotope. Furthermore, the shape of the probability function (kernel) of the positron annihilation abruptly changes at the boundaries of two tissues, such as at the boundary of lungs and surrounding soft tissues, and thus it strongly depends on the particular object's morphology and it is quite challenging to model accurately. In general, the positron range has a small effect (compared to the other effects) for clinical scanners, particularly for studies using ^{18}F-labeled tracers, and can be often ignored. However for small animal imaging and for other tracers (such as ^{82}Rb) the positron range becomes an important effect to be considered.

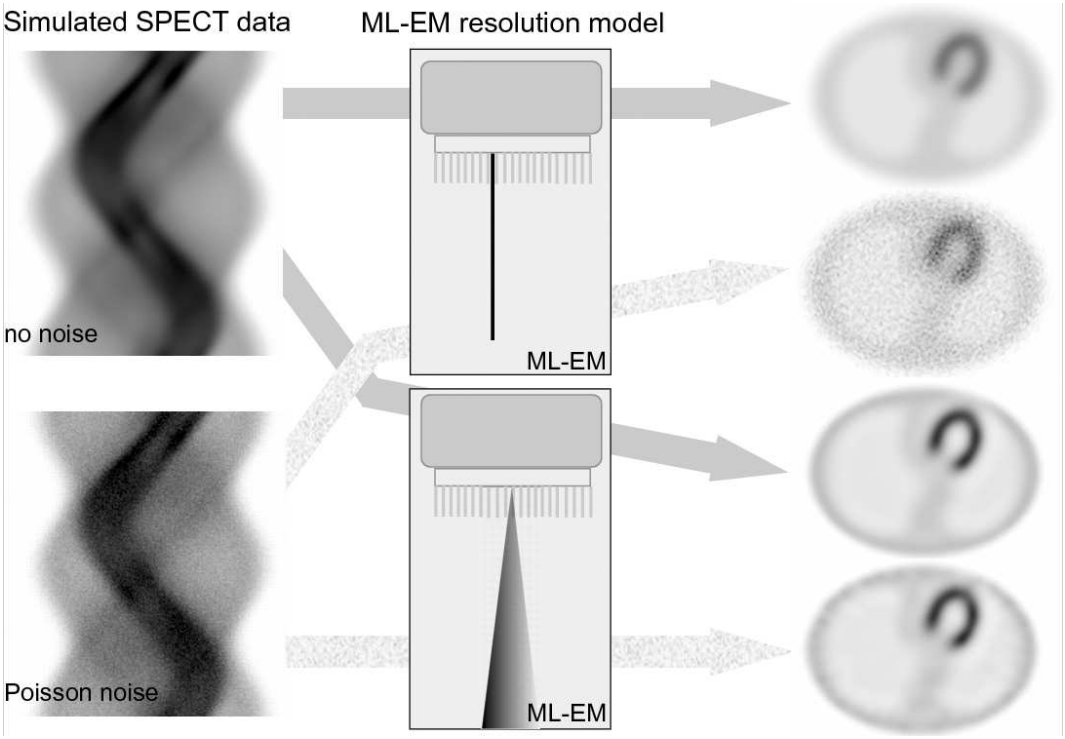

Figure 13.20: *Examples of the effects of resolution modeling within statistical iterative reconstruction. Data were simulated for a SPECT system with depth dependent resolution. It is clearly seen that using the proper resolution model within statistical reconstruction (bottom two images on the right) not only improves resolution of the images, but also helps to efficiently suppress the noise component.*

There is a whole spectrum of approaches how to determine and implement the scanner dependent resolution models. We will mention only the main categories of them. The simplest, but least accurate, approach is to approximate the system resolution model by a spatially invariant resolution kernel, usually a spherically symmetric Gaussian, with the shape (FWHM) estimated from point source measurements at one or more representative locations within the given scanner. This approach typically provides satisfactory results within the central FOV of large whole body PET scanners. However for PET systems with smaller ring diameters (relative to the reconstruction FOV), such as animal systems, and for SPECT systems with depth dependent resolution (and in particular with non-circular orbits), it is desirable to use more accurate spatially variant resolution models.

The second category is using analytically calculated resolution functions (usually spatially variant anisotropic kernels) for each location (LOR) as determined based on analytical models of physical effects affecting the resolution. This approach is usually limited to simple analytical models representing (or approximating) only basic physical characteristics of the system. The resolution kernels are usually calculated on the fly during the reconstruction process when they are needed within the forward and backprojection calculations. In SPECT, the distance dependent collimator blurring requires convolution kernels that become wider, and therefore needing more computation, with increasing distance to the collimator. The computation time can be reduced considerably by integrating an incremental blurring step into the projector (and backprojector), based on Gaussian diffusion. This method, developed by McCarthy and Miller in 1991, is described in more detail in chapter 22 of [5].

A more accurate but computationally very demanding approach is using Monte Carlo simulations

of the resolution functions based on a set of point sources at various (ideally all) image locations. Setting up an accurate mathematical model (transport equations tracing the photon paths through the detector system/crystals) is relatively easy within the Monte Carlo simulations, compared to the analytical approach of determining the resolution function. However to obtain sufficient statistics to get the desired accuracy of the shape of the resolution functions is extremely time consuming. Consequently, simplifications often have to be done in practice, such as determining the resolution kernels only at a set of representative locations and interpolating/extrapolating from them the resolution kernels at other locations.

The most accurate, but also most involved approach is based on experimental measurements of the system response by measuring physical point sources at a set of image locations within the scanner. This is a tedious and very time-consuming process, involving point sources with long half-life isotopes and usually requiring the use of accurate robotic stages to move the point source. Among the biggest challenges is to accumulate a sufficient number of counts to obtain an accurate point spread function, even at limited number of locations. Consequently, the actual resolution kernels used in the reconstruction model are often estimated by fitting analytical functions (kernels) to the measured data, rather than directly using the measured point spread functions.

At the conclusion of this subsection it is worth making the following general comment. In the light of the resolution modeling possibilities discussed above, one might wonder if it is worth spending energy and resources on building new PET and SPECT systems with improved resolution properties. However, although it has been shown in the literature that proper system models lead to improved reconstructed image quality, they can never fully recover information that has been lost through the resolution effects and other instrumentation limitations. Furthermore, due to the increased level of the modeling, the system matrix becomes more dense, and consequently the inverse problem (reconstruction) becomes more ill-posed, thus making it impossible to attain perfect recovery for the realistic data. There is no doubt that both developments of improved instrumentation as well as novel and more accurate reconstruction models play an important role in improving image quality and quantitative accuracy, and eventually increasing the general clinical utility of emission tomography systems.

13.3.6.4 Motion corrections

Due to the relatively long acquisition times motion effects, caused by patient movement and organ motion and deformation, cannot be avoided in emission tomography. In the following we cover all of these effects under the simple term “motion.” With the continuous improvements of PET and SPECT technology, leading to improved spatial resolution, signal to noise ratio, image quality and accuracy of quantitative studies, corrections for motion effects become more important. In fact, artifacts caused by motion are becoming the single most important factor of the image degradation, especially in PET or PET/CT imaging of the upper torso region. For example, the motion effects can lead to the loss of small lesions by blurring them completely out in regions with strong motion (such as near the lower lung wall), or to their misplacement into the wrong anatomic region (e.g. into the liver from the lungs, or vice versa). Motion correction has become an important research topic; however its thorough discussion is out of the scope of this chapter and we refer interested readers to the literature on this topic. In the following we will outline just the main concepts of the motion correction as dealt with within the reconstruction process.

The two main sources of motion related artifacts in emission studies are the motion during the emission scan and the discrepancy (caused by the motion) between the attenuation and emission data. The motion during the emission scan means that the emission paths (LORs) through the object (as considered in the system matrix) change during the scan time. If this time-dependent change is not

accounted for, the system model becomes inconsistent with the data, which results in artifacts and motion blurring in the reconstructed images. On the other hand, the transmission scan (CT) is relatively short and it can be done usually in a breath-hold mode. Consequently, the attenuation image is usually motion-free and capturing only one particular patient position and organ configuration (time-frame). If the attenuation factors obtained from this fixed-time-position attenuation image are applied to the emission data acquired at different time frames (or averaged over many time frames) this leads to artifacts in the reconstructed images, which tend to be far more severe in PET than in SPECT. This is, for example, most extremely pronounced at the bottom of the lungs which can typically move several centimeters during the breathing cycle, causing motion between two regions with very different attenuation coefficients.

Emission data motion - Let us first discuss correction approaches for motion during the emission scan. The first step is subdividing the data (in PET typically list-mode data) into a sufficient number of time frames to ensure that the motion within each frame is small. For the organ movement the frames can be distributed over a period of the organ motion (e.g., breathing cycle). For the patient motion the frames would be typically longer and distributed throughout the scan time. The knowledge about the motion can be obtained using external devices, such as cameras with fiducial markers, expansion belts or breathing sensors for respiratory motion, the ECG-signal for cardiac motion etc. There are also a limited number of approaches for estimating the motion directly from the data.

Once the data are subdivided into the set of the frames, the most straightforward approach is to reconstruct data independently in each frame. The problem with this approach is that the resulting images have a poor signal to noise ratio because the acquired counts have been distributed into a number of individual (now low count) frames. To improve the signal to noise ratio, the reconstructed images for individual frames can be combined (averaged) after they are registered (and properly deformed) to the reference time frame image. However, for statistical non-linear iterative reconstruction algorithms, this is not equivalent to (and typically of a lower quality than) the more elaborate motion correction approaches taking into account all of the acquired counts in a single reconstruction, as discussed below.

For rigid motion (e.g. in brain imaging) the events on LORs (LOR_i) from each time frame, or time position, can be corrected for motion by translation (using affine transformations) into the new LORs ($LOR_{i'}$) in the reference frame (see Figure 13.21-top right, solid line), in which the events would be detected if there was no motion. Reconstruction is then done in a single reference frame using all acquired counts, leading to a better SNR in the reconstructed images. Care has to be taken with the detector normalization factors so that the events are normalized using the proper factors (N_i) for the LORs on which they were actually detected (and not into which they were translated). Attenuation factors are obtained on the transformed lines ($att_{i'}$) through the attenuation image in the reference frame. Care has also to be given to the proper treatment of data LORs with events being translated into, or out of, the detector gaps or detector ends. This is important in particular for the calculation of the sensitivity matrix, which then becomes a very time consuming process.

For non-rigid (elastic) motion, which is the case for most of the practical applications, the motion correction procedures become quite involved. There are two basic possibilities. The first approach is to derive the transformations of individual paths of events (LORs) from each frame into the reference frame (see Figure 13.21-top right, dotted line). For the non-rigid motion, the transformed paths through the reference object frame are not straight lines anymore, thus leading to very large computational demands for the calculations of the forward and back-projection operations. The same care of normalization, gaps, and detector ends has to be taken as above.

The second, more efficient, approach involves morphing the image estimate (of the reference im-

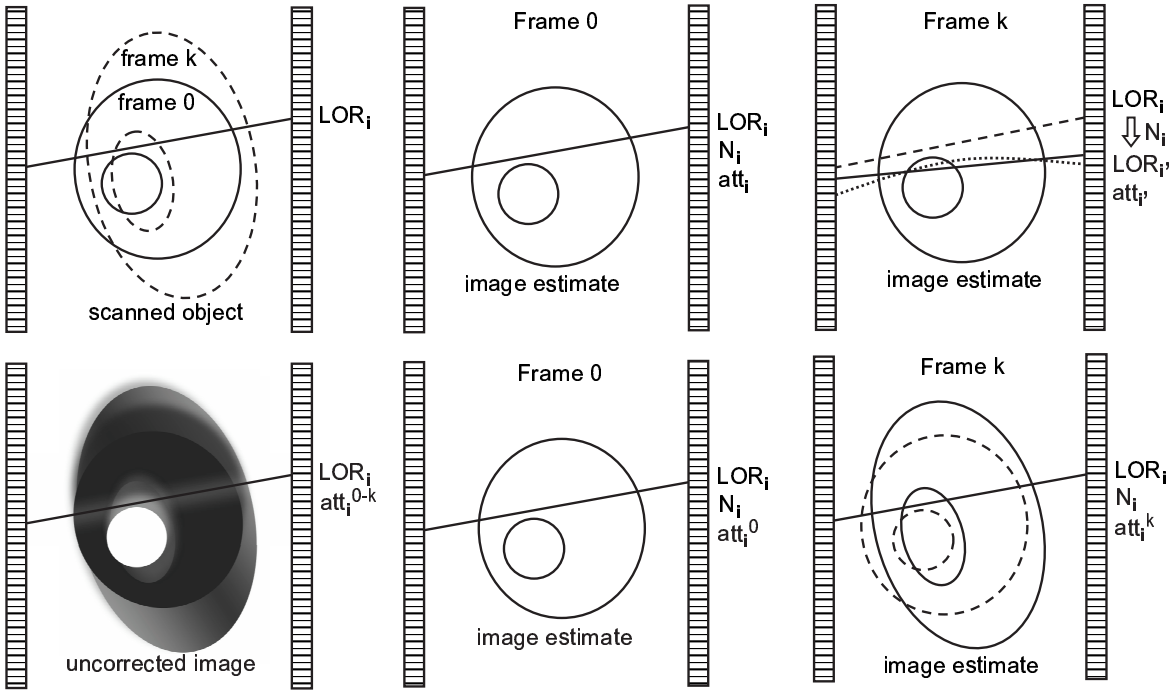

Figure 13.21: *Illustration of motion corrections for events acquired within line-of-response LOR_i, with corresponding normalization N_i and attenuation att_i factors. Left-top: positions and shapes of the object in the reference time* frame 0 *and* frame k. *Left-bottom: illustration of blurring in the reconstruction combining events from all frames without motion correction (attenuation factors are also averaged over the whole range of the frames att_i^{0-k}). Middle column: processing within the reference time frame. Right top: LOR-based motion correction for* frame k - *the LOR_i (dashed line) has to be transformed to the $LOR_{i'}$ (solid line for rigid motion, dotted line for non-rigid motion) which represent the paths that the photons would travel through the reference object if there was no motion. Note that although the LORs are transformed, the normalization factors are used for the crystal pairs (LORs) in which the events were detected (N_i), while the used attenuation factors are for the transformed paths ($att_{i'}$). Right-bottom: image-based motion correction, including image morphing of the estimated image from the reference frame (dashed lines) into the given frame (solid line).*

age) into the frame for which we are processing current events (LORs) (see Figure 13.21-bottom right, solid line). Note that we consider some pre-sorting of the data so that we process events from each frame together (using common image morphing operation). Here the acquired LORs (LOR_i) and their normalization coefficients (N_i) are directly used without modification. But the sensitivity matrix still needs to be carefully calculated taking into consideration update and subset strategy, e.g., including morphing operation if subset data involve several frames. This is however a simpler operation than in the LOR-based case since the morphing is done in the image domain. This image-based approach is not only more efficient, but also better reflects/models the actual data acquisition process during which the acquired object is being changed (morphed).

Attenuation effects - In the following, we consider that we have available either attenuation information for each time frame, e.g., having a sequence of CT scans for different time positions, or have knowledge on the motion and tools to morph a fixed time-position CT image to represent attenua-

tion images at individual time frames. We further consider that we have available tools to obtain the motion transformation of data and/or images between the individual time frames.

If the emission data are stored or binned without any motion gating, they represent motion-blurred emission information over the duration of the scan. Using for them attenuation information for a fixed time position is not correct. It would be better to pre-correct those data using proper attenuation factors for each frame, but then the statistical properties (Poisson character) are lost due to the pre-correction. A good compromise (although not theoretically exact) is to use motion-blurred attenuation factors during the pre-correction or the reconstruction process.

For data stored in multiple time-frames, separate attenuation factors (or their estimates) are used for each frame, such that they reflect attenuation factors (for each LOR) at that particular time-frame. For the case when we have multiple CT images this is simply obtained by calculation (forward projection) of the attenuation coefficients for each frame from the representative CT image for that frame. For the case when we have only one CT image, we have either to calculate attenuation factors on the modified LORs (for each time frame) in the LOR-based corrections, or to morph the attenuation image for each frame and then calculate the attenuation factors from the morphed images in the image based corrections.

13.4 Noise estimation

13.4.1 Noise propagation in FBP

The pixel variance in an image reconstructed with FBP can be estimated analytically, by propagating the uncorrelated Poisson noise in the data through the reconstruction operation. The FBP algorithm can be written as

$$\Lambda(x,y) = \int_0^{\pi} d\phi \int_{-\infty}^{\infty} Y(x\cos\phi + ysin\phi - s)h(s)\; ds, \tag{13.86}$$

where $h(s)$ is the convolution kernel, combining the inverse Fourier transform of the ramp filter and a possible low pass filter to suppress the noise. The variance on the measured sinogram $Y(s,\phi)$ data equals its expectation $\bar{Y}(s,\phi)$, the covariance between two different sinogram values $Y(s,\phi)$ and $Y(s',\phi')$ is zero. Consequently, the covariance between two reconstructed pixel values $\Lambda(x,y)$ and $\Lambda(x',y')$ equals

$$\begin{aligned} \text{covar}(\Lambda(x,y),\Lambda(x',y')) \;\; = \;\; & \int_0^{\pi} d\phi \int_{-\infty}^{\infty} ds\; \bar{Y}(x\cos\phi + y\sin\phi - s) \\ & h(s)\; h\left(s + (x'-x)\cos\phi + (y'-y)\sin\phi\right). \end{aligned} \tag{13.87}$$

This integral is non-zero for almost all pairs of pixels. Because $h(s)$ is a high pass filter, neighboring reconstruction pixels tend to have fairly strong negative correlations. The correlation decreases with increasing distance between the (x,y) and (x',y'). The variance is obtained by setting $x = x'$ and $y = y'$, which produces

$$\text{var}(\Lambda(x,y)) = \int_0^{\pi} d\phi \int_{-\infty}^{\infty} ds\; \bar{Y}(x\cos\phi + y\sin\phi - s)|h(s)|^2 \tag{13.88}$$

Figure 13.22 shows the variance image of the FBP reconstruction of a simulated PET sinogram of a heart phantom. The image was obtained by reconstructing 400 sets of noisy PET data. The figure also shows a noise-free and one of the noisy FBP images. The noise creates streaks that extend to the edge of the image. As a result, the variance is non-zero in the entire image.

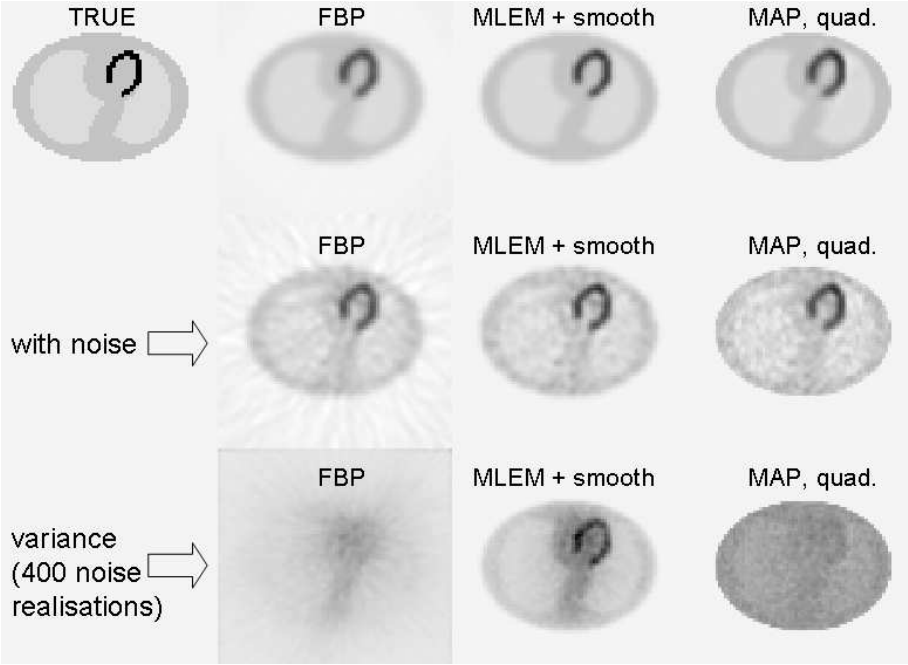

Figure 13.22: *Simulated PET reconstructions of a heart phantom. Reconstructions were done with FBP, MLEM with Gaussian post-smoothing and with MAP using a quadratic prior. For each algorithm a noise free and a noisy reconstruction are shown, and also the pixel variance obtained from 400 independent Poisson noise realisations on the simulated PET data. All reconstructions (first two rows) are shown on the same gray value scale. A second scale was used to display the three variance images. The noisy FBP image contains negative pixels (displayed in white with this scale).*

13.4.2 Noise propagation in MLEM

The noise analysis of MLEM (and MAP) reconstruction is more complicated than that for FBP, because these algorithms are non-linear. However, the MLEM algorithm has some similarity with the weighted least squares algorithm, which can be described with matrix operations. The WLS-reconstruction was described previously; equation (13.45) is repeated here for convenience (the additive term was assumed zero for simplicity):

$$\boldsymbol{\lambda} = (\mathbf{A}'\mathbf{C_y}^{-1}\mathbf{A})^{-1}\ \mathbf{A}'\mathbf{C_y}^{-1}\ \mathbf{y} \tag{13.89}$$

$\mathbf{C_y}$ is the covariance of the data, which is defined as $\mathbf{C_y} = E(\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})'$, where E denotes the expectation, and $\bar{\mathbf{y}}$ is the expectation of $\mathbf{y}$. The covariance of the reconstruction is then

$$\begin{aligned} \mathbf{C}_\lambda &= E\,(\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}})(\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}})' \\ &= (\mathbf{A}'\mathbf{C_y}^{-1}\mathbf{A})^{-1}\ \mathbf{A}'\mathbf{C_y}^{-1}\ E(\mathbf{y} - \bar{\mathbf{y}})(\mathbf{y} - \bar{\mathbf{y}})'\mathbf{C_y}^{-1}\mathbf{A}(\mathbf{A}'\mathbf{C_y}^{-1}\mathbf{A})^{-1} \\ &= (\mathbf{A}'\mathbf{C_y}^{-1}\mathbf{A})^{-1} \end{aligned} \tag{13.90}$$

This matrix gives the covariances between all possible pixel pairs in the image produced by WLS-reconstruction. The projection A and backprojection A' have a low pass characteristic. Consequently, the inverse $(\mathbf{A}'\mathbf{C_y}^{-1}\mathbf{A})^{-1}$ acts as a high pass filter. It follows that neighboring pixels of WLS-reconstructions tend to have strong negative correlations, as is the case with FBP. Because of this, the MLEM variance decreases rapidly with smoothing.

Figure 13.22 shows mean and noisy reconstructions and variance images of MLEM with Gaussian post-smoothing and MAP with a quadratic prior. For these reconstructions, 16 iterations with 8 subsets were applied. MAP with a quadratic prior produces fairly uniform variance, but with a position dependent resolution. In contrast, post-smoothed MLEM produces fairly uniform spatial resolution, in combination with a non-uniform variance.

Bibliography

[1] RM Lewitt and S Matej. “Overview of Methods for Image Reconstruction from Projections in Emission Computed Tomography”, *Proceedings of the IEEE* 2003; 91: 1588-1611.

[2] F Natterer. “The mathematics of computerized tomography”, SIAM, 1986.

[3] AC Kak and M Slaney. “Principles of computerized tomographic imaging”, SIAM, 1988.

[4] HH Barrett and KJ Myers. “Foundations of image science”, Wiley, 2004.

[5] MN Wernick and JN Aarsvold, edts. “Emission tomography, the fundamentals of PET and SPECT”, Elsevier Academic Press, 2004.

[6] F Natterer. “Inversion of the attenuated Radon transform”. *Inverse Problems*. 2001; 17:113-119.

[7] W Xia, RM Lewitt and PR Edholm. “Fourier Correction for Spatially Variant Collimator Blurring in SPECT”, *IEEE Transactions on Medical Imaging* 1995; 14: 100-115.

[8] M Defrise, R Clack, DW Townsend. “Image reconstruction from truncated, two-dimensional, parallel projections” *Inverse Problems*, 1996; 11: 287-313.

[9] M Defrise, S Kuijk, F Deconinck. “A new three-dimensional reconstruction method for positron cameras using plane detectors”, *Phys Med Biol* 1988; 33: 43-51.

[10] PE Kinahan, JG Rogers. “Analytic three-dimensional image reconstruction using all detected events”, *IEEE Trans Nucl Sci* 1990; NS-36: 964-968.

[11] ME Daube-Witherspoon, G Muehllehner. “Treatment of axial data in three-dimensional PET”, *J Nucl Med* 1987, 28: 1717-1724.

[12] RM Lewitt, G Muehllehner, JS Karp. “Three-dimensional image reconstruction for PET by multi-slice rebinning and axial image filtering”, *Phys Med Biol* 1994, 39: 321-339.

[13] M Defrise, PE Kinahan, DW Townsend, C Michel, M Sibomana, DF Newport. “Exact and approximate rebinning algorithms for 3D PET data”, *IEEE Trans Med Imaging*, 1997; 16: 145-158.

[14] M Defrise. “A factorization method for the 3D X-ray transform”, *Inverse Problems* 1995; 11: 983-994.

[15] T Tomitani. “Image reconstruction and noise evaluation in photon time-of-flight assisted positron emission tomography”. *IEEE Trans Nucl Science*, 1981; NS-28: 4582-4589.

[16] J Qi and RM Leahy. “Iterative reconstruction techniques in emission computed tomography”, *Physics in Medicine and Biology* 2006; 51: R541-R578.

[17] JA Fessler and SD Booth. “Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction.” *IEEE Tr. Im. Proc.*, 1999; 8(5):688-699.

[18] WH Press, BP Flannery, SA Teukolsky, WT Vetterling. “Numerical recipes, the art of scientific computing”, Cambridge University Press, 1986.

[19] AR De Pierro. “A modified expectation maximization algorithm for penalizd likelihood estimation in emission tomography”, *IEEE Trans Med Imaging* 1995; 14: 132-137.

[20] LS Shepp, Y Vardi. “Maximum likelihood reconstruction for emission tomography,” *IEEE Trans Med Imaging*, vol MI-1, pp. 113-122, 1982.

[21] J. Qi. “Calculation of the sensitivity image in list-mode reconstruction,” *IEEE Trans. Nucl. Sci.*, vol. 53, no. 5, pp. 2746–2751, 2006.

[22] S. Matej, S. Surti, S. Jayanthi, M. E. Daube-Witherspoon, R. M. Lewitt, J. S. Karp. “Efficient 3-D TOF PET reconstruction using view- grouped histo-images: DIRECT - Direct Image Reconstruction for TOF” *IEEE Trans Med Imaging* 2009; 28: 739-751.

[23] MH Hudson, RS Larkin. “Accelerated image reconstruction using ordered subsets of projection data,” *IEEE Trans Med Imaging* vol 13, pp. 601-609, 1994.

[24] J Browne, AR De Pierro. “A row-action alternative to the EM algorithm for maximizing likelihoods in emission tomography”, *IEEE Trans Med Imaging* 1996; 15: 687-699.

[25] ME Daube-Witherspoon, S Matej, JS Karp, RM Lewitt. “Application of the row action maximum likelihood algorithm with spherical basis functions to clinical PET imaging”, *IEEE Trans Nucl Sci* 2001; 48: 24-30.

[26] DL Snyder, MI Miller, LJ Thomas Jr, DG Politte. “Noise and edge artifacts in maximum-likelihood reconstructions for emission tomography”, *IEEE Trans Med Imaging* 1987; MI-6: 228-238.

[27] RM Leahy and J Qi. “Statistical approaches in quantitative positron emission tomography”, *Statistics and Computing* 2000; 10: 147-165.