Gengsheng Lawrence Zeng

Medical Image Reconstruction

A Conceptual Tutorial

高等教育出版社
HIGHER EDUCATION PRESS

Springer

Gengsheng Lawrence Zeng

Medical Image Reconstruction

A Conceptual Tutorial

Gengsheng Lawrence Zeng

Medical Image Reconstruction

A Conceptual Tutorial

With 163 Figures

高等教育出版社
HIGHER EDUCATION PRESS

Springer

Author
Prof. Dr. Gengsheng Lawrence Zeng
Utah Center for Advanced Imaging Research
Department of Radiology
University of Utah
Salt Lake City, UT 84108, USA
E-mail: larry@ucair.med.utah.edu

ISBN 978-7-04-020437-7
Higher Education Press, Beijing

ISBN 978-3-642-05367-2 e-ISBN 978-3-642-05368-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2009937574

Cover design: Frido Steinen-Broo, EStudio Calamar, Spain

Printed on acid-free paper

Springer is part of Springer Science + Business Media (www.springer.com)

This book is dedicated to Ya, Andrew,
Kathy, and Megan

Preface

The first time I heard about image reconstruction was twenty years ago I came to the University of Utah as a post-doctoral fellow in the Department of Radiology. Dr. Grant Gullberg and Dr. Rolf Clackdoyle gave many lectures on image reconstruction and I took notes. Even today I still go back to those notes from time to time. I benefit from those notes significantly. This book is complied together with parts of those notes and some current research papers with most mathematical proofs removed. I am grateful to Dr. Gullberg and Dr. Clackdoyle for introducing me to the wonderful world of image reconstruction. I appreciate Dr. Michel Defrise, Dr. Ge Wang, and Dr. Guang-Hong Chen for their helpful suggestions. I also like to thank my colleagues in the department and in other institutions. I would especially like to thank Kathy Gullberg and Jacob Piatt for proof-reading the drafts.

This tutorial text introduces the classical and modern image reconstruction technologies to the general audience. It covers the topics in two-dimensional (2D) parallel-beam and fan-beam imaging, three-dimensional (3D) parallel ray, parallel plane, and cone-beam imaging. Both analytical and iterative methods are presented. The applications in X-ray CT, SPECT (single photon emission computed tomography), PET (positron emission tomography), and MRI (magnetic resonance imaging) are also discussed. Contemporary research results in exact ROI (region-of-interest) reconstruction with truncated projections, Katsevich's cone-beam filtered backprojection algorithm, and reconstruction with highly undersampled data with l_0-minimization are also included in this book.

This book is written in an easy-to-read style, which lets the diagrams do the most talking. The readers who intend to get into medical image reconstruction will gain the general knowledge of the field in a painless way. I hope you enjoy reading it as much as I enjoy writing (and drawing) it. The first time reader can skip the more challenging materials marked by the "$*$" sign without interrupting the flow of this book.

Gengsheng Lawrence Zeng
Salt Lake City
August 2009

Contents

* The first time reader can skip the more challenging materials marked by the "$*$" sign without interrupting the flow of this book.

1 Basic Principles of Tomography

This is an introductory chapter, which presents the fundamental concepts in tomography. It first defines what the tomography is, and shows how a tomographic image can obtained from its measurements using two simple examples. The concept of projection is then explained. Next, the filtered back-projection image reconstruction method is introduced using a point source example. Finally, the concept of backprojection is discussed.

1.1 Tomography

The Greek word *tomos* means a section, a slice, or a cut. *Tomography* is the process of imaging a cross section. For example, if you are given a watermelon and would like to see inside, the easiest way to do so is to cut it open (see Figure 1.1). Clearly, this approach to obtain a cross-section image is not a

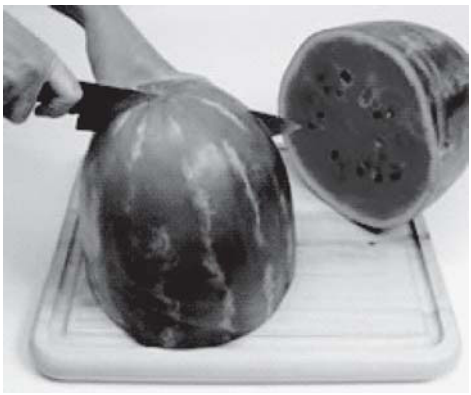


Fig. 1.1. Cutting open to see what is inside.

good idea in medicine. Nobody wants to be cut open in order to see what is inside.

Let us look at another example. You are visiting a small park, which is closed for maintenance. You walk around the park and take a few pictures of it. After you get home, you can use your pictures to make a map of the park. To make your life easier, let us assume that there are two large trees in the park, and you take two pictures from the east and the south, as shown in Figure 1.2 Left. Using these two pictures, you can map out where these two trees are, as shown in Figure 1.2 Right. This can be done by positioning the pictures at the original orientations at which the pictures were taken, drawing a line from each tree, and finding the intersections. If you have enough pictures, it is not hard to find out where the trees are.

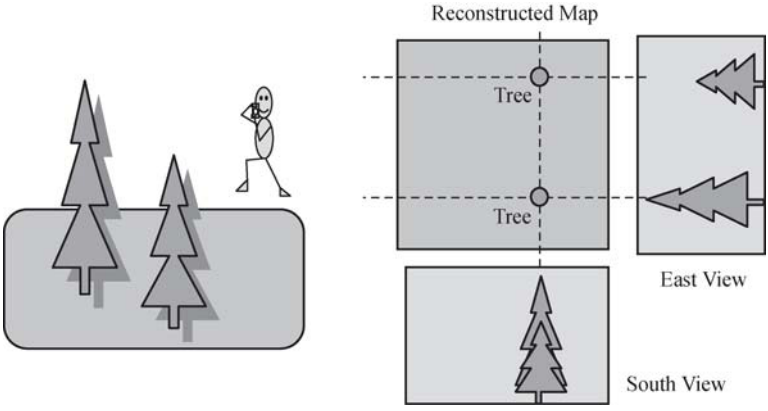


Fig. 1.2. Reconstruct a map from two pictures.

Tomography is a mathematical problem. Let us do a fun mathematical exercise here. We have a 2×2 matrix. We do not tell you what it is yet. Here are the hints: The sum of the first row is 5, the sum of the second row is 4, the sum of the first column is 7, and the sum of the second column is 2 (see Figure 1.3). Now, you figure out what this 2×2 matrix is.

You can solve this puzzle by setting up a system of linear equations with the matrix entries as unknowns:

$$\begin{cases} x_1 + x_2 = 5, \\ x_3 + x_4 = 4, \\ x_1 + x_3 = 7, \\ x_2 + x_4 = 2. \end{cases} \tag{1.1.1}$$

Solving these equations, you will get

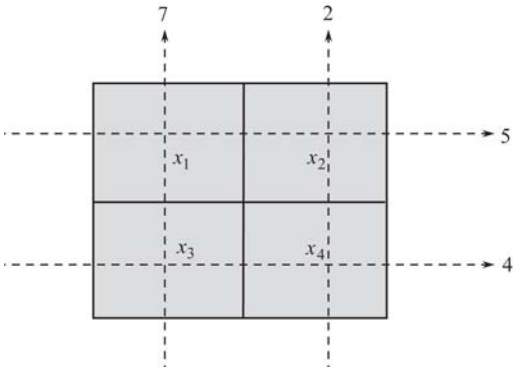


Fig. 1.3. A 2×2 matrix puzzle.

$$\begin{cases} x_1 = 3, \\ x_2 = 2, \\ x_3 = 4, \\ x_4 = 0. \end{cases} \tag{1.1.2}$$

Congratulations! You have just mathematically solved a tomography problem. Usually, a tomography problem is solved mathematically, hence the term CT(computed tomography). The row sum or column sum in this example can be generalized as a *ray sum*, a *line integral*, or a *projection*. The procedure to produce a tomographic image from projections is called *image reconstruction*.

What if the tomography problem gets more complicated? If there are many more trees in the park, taking only two pictures may not provide us enough information to map out the park. If the matrix size is larger than 2×2, the row sum and column sum alone do not form enough equations to solve for the matrix entries.

We need more views! For the matrix identification case, we need to sum the matrix diagonally at various angles. In turn, more sophisticated mathematics is required to solve the tomography problem.

1.2 Projection

In order to understand the concept of projection (ray sum, line integral, or Radon transform), we will present more examples here.

In the first example, the object is a uniform disc on the x-y plane, the

center of the disc is at the origin, and the (linear) density of the disc is ρ (see Figure 1.4). The projection (i.e., the line integral) of this object can be calculated as the chord length t times the linear density ρ. That is

$$p(s) = \begin{cases} \rho t = 2\rho\sqrt{R^2 - s^2}, & |s| < R, \\ 0, & |s| \geqslant R. \end{cases} \tag{1.2.1}$$

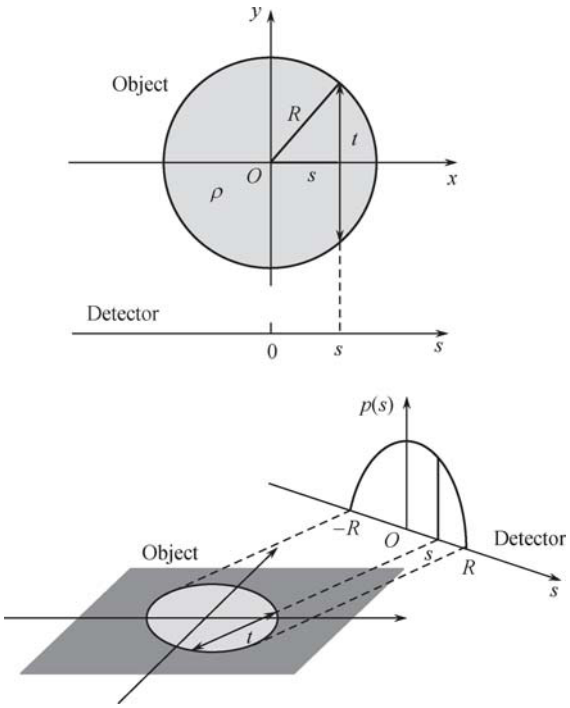


Fig. 1.4. The line integral across the disc is the length of a chord times the density.

In this particular example, the projection $p(s)$ is the same for any view angle θ, which is the orientation of the detector.

If the object is more complicated, the projection $p(s, \theta)$ is angle θ dependent (see Figure 1.5).

In the next example we use a point source on the y-axis to further illustrate the angle θ dependency of the projection $p(s, \theta)$ (see Figure 1.6). Here we pay attention to the location s of the spike on the 1D detector, which can be evaluated as

$$s = r \sin\theta. \tag{1.2.2}$$

This is a sine function with respect to θ. If you display this point source projection data set $p(s, \theta)$ in the s-θ coordinate system (see Figure 1.6 Right),

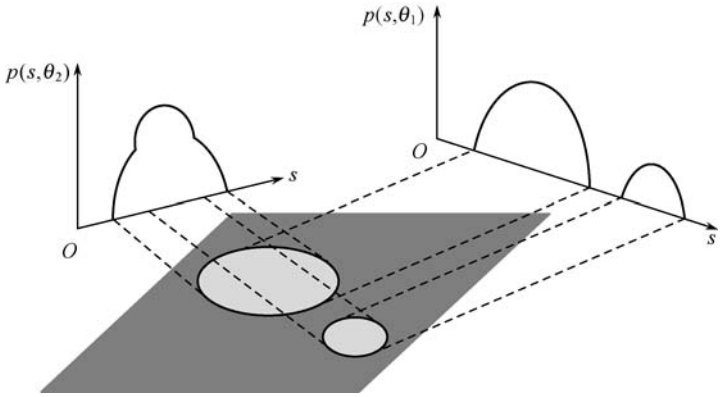


Fig. 1.5. The projections are usually different at a different view-angle.

you will see the trajectory of a sine wave. Because of this observation, people refer to the projection data set as a *sinogram*.

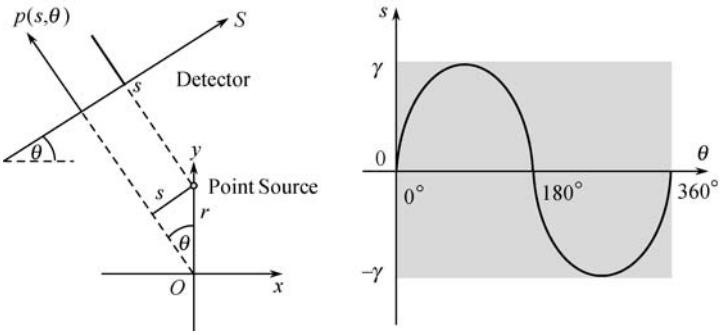


Fig. 1.6. A sinogram is a representation of the projections on the s-θ plane.

The fourth example is a discrete object of a 2×2 matrix, similar to that in Figure 1.3. The detector is also discrete with 4 detector bins (see Figure 1.7). Each matrix element represents a uniform pixel, and x_i $(i = 1, 2, 3, 4)$ is the linear density in the ith pixel. Here we would like to find the line-integral $p(s, \theta)$ of the matrix at a view angle θ. The quantity a_{ij} $(i = 1, 2, 3, 4$, and $j = 1, 2, 3, 4)$ is the segment length of the path towards the detector bin i within the pixel j, and $a_{ij} = 0$ if the jth pixel is not on the path to the ith detector bin. The projection $p(i, \theta)$ is calculated as

$$p(i, \theta) = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + a_{i4}x_4, \qquad i = 1, 2, 3, 4. \tag{1.2.3}$$

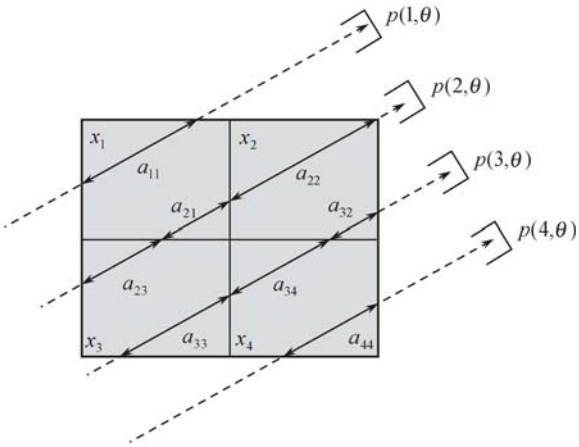


Fig. 1.7. The projections are weighted by the line-length within each pixel.

1.3 Image Reconstruction

In this section, we illustrate the common image reconstruction strategy by considering a point source. Let us consider an empty two-dimensional (2D) plane with an x-y coordinate system, and we place a small dot with a value, say one, somewhere on this plane and not necessarily at the origin (see Figure 1.8). We now imagine that there is a detector (e.g., a camera) rotating around the origin, acquiring images of projections. At a particular angle θ, we denote the projection as $p(s, \theta)$, where s is the coordinate on the detector.

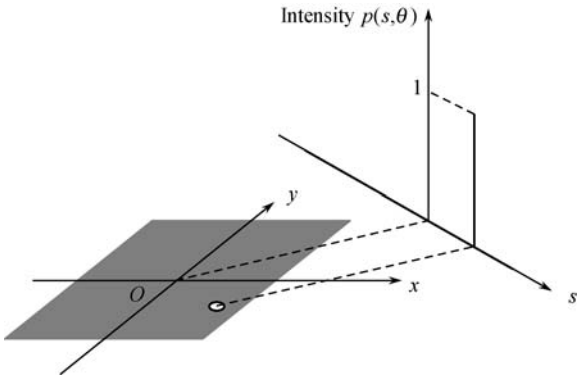


Fig. 1.8. Projection of a point source object.

The projection $p(s, \theta)$ is formed by drawing a line across the x-y plane, orthogonal to the detector, and meeting on the detector at location s. Then we evaluate the line integral along this line, and the integral value is $p(s, \theta)$. In our example, if the line does not touch the point source, $p(s, \theta)$ is zero. If the line passes through the point source, $p(s, \theta)$ is one.

Now we are going to reconstruct the image using the projections $p(s, \theta)$. Our strategy is similar to that in the tree-map example in Section 1.1, where we drew a line from each tree on the detector and found the location of the intersections. In image reconstruction, we not only need to find the location but also the intensity value of the object of interest.

As shown in Figure 1.9 (a), a number of projections are taken from the point source at various view angles. We attempt to reconstruct the point source image in the following manner.

When you look at the projections $p(s, \theta)$ at one view θ, you see a spike of intensity one. This spike is the sum of all activity along the projection path. To reconstruct the image, you must re-distribute the activity in the spike back to its original path. The problem is that you do not know where you need to put more activity along the path and where you put less. Before you give up, you decide to put equal amounts of activity everywhere along the path, and the amount is the magnitude of the projection spike [see Figure 1.9 (b)]. If you do that for few more angles, you will have the situation as shown in Figure 1.9 (c). Due to the superposition effect, there will be a tall spike in the x-y plane at the location of the point source.

What you have just done is a standard mathematical procedure called *backprojection*. If you backproject from all angles from 0° to 360° you will produce an image similar to the one shown in Figure 1.9 (d).

After backprojection, the image is still not quite the same as the original image but rather is a blurred version of it. To eliminate the blurring, we introduce negative "wings" around the spike in the projections before backprojection [see Figure 1.9 (e)]. The procedure of adding negative wings around the spike is called *filtering*. The use of the negative wings results in a clear image [see Figure 1.9 (f)]. This image reconstruction algorithm is very common and is referred to as a Filtered Backprojection (FBP) algorithm.

In this section, we use a point source to illustrate the usefulness of filtering and backprojection with many views in image reconstruction. We must point out that if the object is a point source, we only need two views to reconstruct the image, just like the map making example in Section 1.1.

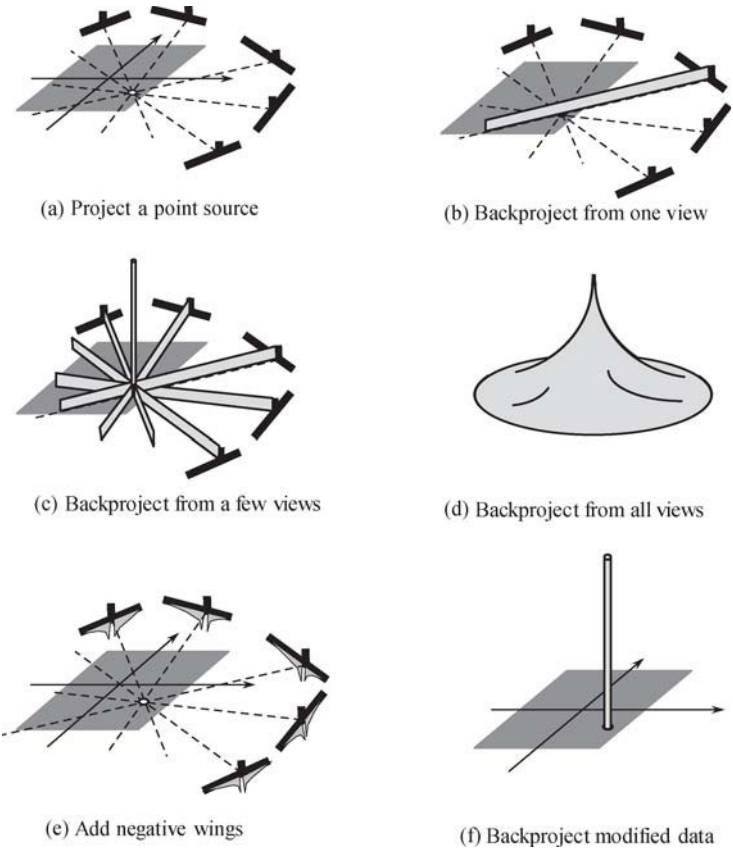


Fig. 1.9. Reconstruction of a point source image by backprojecting unfiltered and filtered data.

1.4 Backprojection

One must first define projection before backprojection can be defined. We must make it clear that backprojection is not the inverse of projection. Backprojection alone is not sufficient to reconstruct an image. After you backproject the data, you do not get the original image back. We will illustrate this point by a simple discrete 2×2 problem below (see Figure 1.10).

The original image is defined by $x_1 = 3$, $x_2 = 2$, $x_3 = 4$, and $x_4 = 0$. The associated projections are $p(1, 0°) = 7$, $p(2, 0°) = 2$, $p(1, 270°) = 5$, and $p(2, 270°) = 4$. The projections are formed one view at a time (see Figure

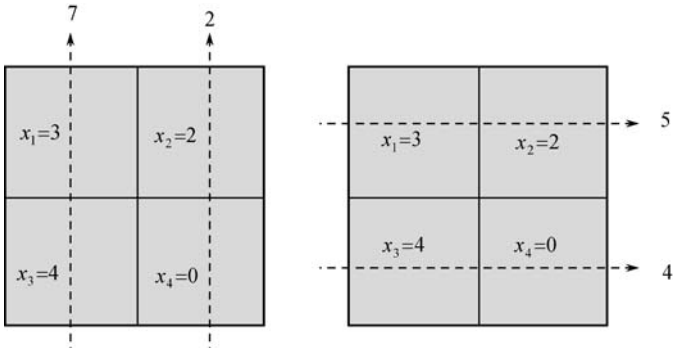


Fig. 1.10. View-by-view projection.

1.10). The backprojected image is also formed one view at a time. The final backprojected image is the summation of the backprojections from all views, as shown in Figure 1.11. Please note that the backprojected image is different from the original image.

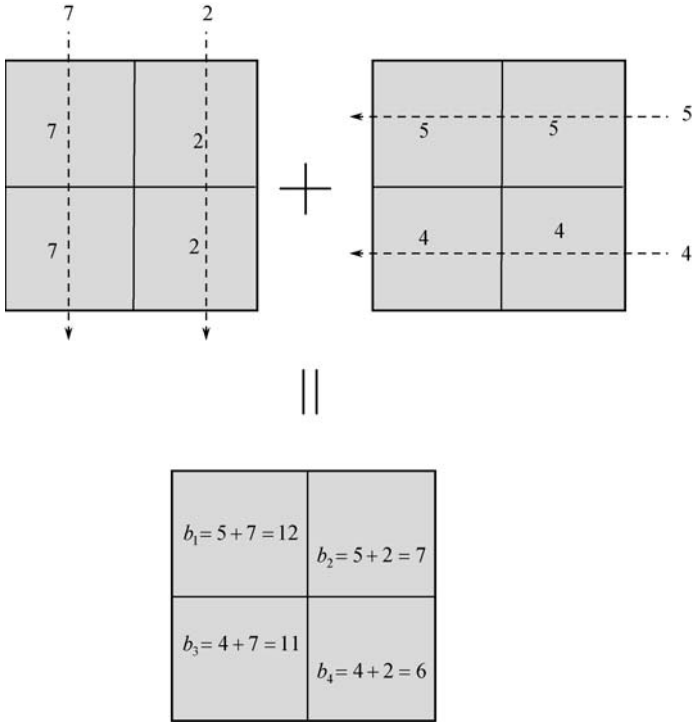


Fig. 1.11. View-by-view backprojection, then sum all backprojected images.

Even though the backprojected image is not the original image, they are closely related. Their relationship will be further discussed in the next chapter.

*1.5 Mathematical Expressions

In every chapter we dedicate a section especially to mathematical expressions. These mathematical expressions help the advanced readers to better grasp the main concepts which are discussed in the chapter. Mathematical expressions of projection and backprojection for 2D parallel-beam imaging are presented in this section. The Dirac δ function has an important role in analytic algorithm development; its definition and some properties are also covered in this section.

1.5.1 Projection

Let $f(x, y)$ be a density function in the x-y plane. The projection (ray sum, line integral, or Radon transform) $p(s, \theta)$ has many equivalent expressions such as

$$p(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - s)\mathrm{d}x\mathrm{d}y, \tag{1.5.1}$$

$$p(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(\boldsymbol{x}\cdot\boldsymbol{\theta} - s)\mathrm{d}x\mathrm{d}y, \tag{1.5.2}$$

$$p(s,\theta) = \int_{-\infty}^{\infty} f(s\cos\theta - t\sin\theta, s\sin\theta + t\cos\theta)\mathrm{d}t, \tag{1.5.3}$$

$$p(s,\theta) = \int_{-\infty}^{\infty} f(s\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp})\mathrm{d}t, \tag{1.5.4}$$

$$p(s,\theta) = \int_{-\infty}^{\infty} f_{\theta}(s,t)\mathrm{d}t, \tag{1.5.5}$$

where $\boldsymbol{x} = (x, y)$, $\boldsymbol{\theta} = (\cos\theta, \sin\theta)$, $\boldsymbol{\theta}^{\perp} = (-\sin\theta, \cos\theta)$, δ is the Dirac delta function, and f_θ is the function f rotated by θ clockwise. We assume that the detector rotates counter-clockwise around the object or that the object rotates clockwise while the detector stays still. The coordinate system is shown in Figure 1.12.

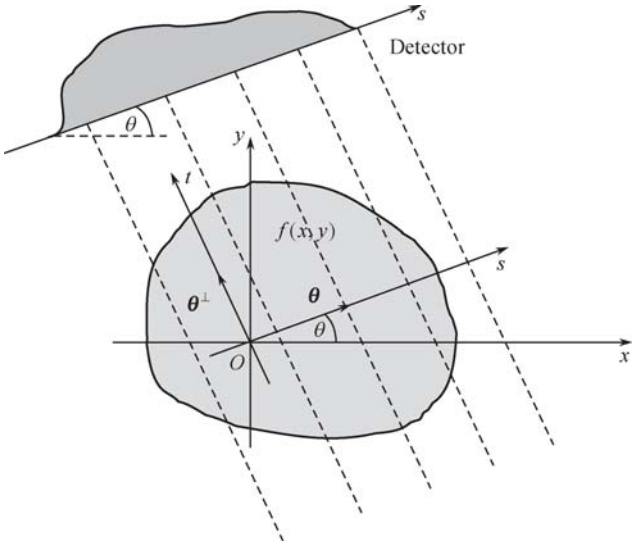


Fig. 1.12. Coordinate system for 2D parallel-beam imaging.

1.5.2 Backprojection

Backprojection is the adjoint of projection. Here "adjoint" is a mathematical term. It refers to the conjugate transpose in linear algebra. For a real matrix A, its adjoint is simply the transposed matrix A^{T}. In the discrete case as in Section 1.4, the projection is

$$P = AX, \tag{1.5.6}$$

where X represents an image, but in a column form. For example, the 2×2 image is expressed as (see Figures 1.3 or 1.10)

$$X = [x_1, x_2, x_3, x_4]^{\mathrm{T}}. \tag{1.5.7}$$

The column matrix P represents the projections. If we use the example in Figure 1.10,

$$P = [p(1, 0^\circ), p(2, 0^\circ), p(1, 270^\circ), p(2, 270^\circ)]^{\mathrm{T}} = [7, 2, 5, 4]^{\mathrm{T}}. \tag{1.5.8}$$

The matrix A is the projection operator. Its entries a_{ij} are defined in Figure 1.7. Using the example of Figure 1.10, the backprojection of P can be

calculated using matrix multiplication as

$$B = A^{\mathrm{T}}P = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} 7 \\ 2 \\ 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 2 \\ 5 \\ 4 \end{bmatrix} = \begin{bmatrix} 12 \\ 7 \\ 11 \\ 6 \end{bmatrix}, \tag{1.5.9}$$

which is the same as the result obtained "graphically" in Figure 1.11.

For the continuous case, the backprojection image $b(x, y)$ can be expressed in the following equivalent ways:

$$b(x,y) = \int_0^{\pi} p(s,\theta)\big|_{s=x\cos\theta+y\sin\theta}\mathrm{d}\theta, \tag{1.5.10}$$

$$b(x,y) = \int_0^{\pi} p(s,\theta)\big|_{s=\boldsymbol{x}\cdot\boldsymbol{\theta}}\mathrm{d}\theta, \tag{1.5.11}$$

$$b(x,y) = \int_0^{\pi} p(\boldsymbol{x}\cdot\boldsymbol{\theta},\theta)\mathrm{d}\theta, \tag{1.5.12}$$

$$b(x,y) = \frac{1}{2}\int_0^{2\pi} p(x\cos\theta + y\sin\theta,\theta)\mathrm{d}\theta. \tag{1.5.13}$$

1.5.3 The Dirac δ-function

The Dirac δ-function is not a regular function that maps a value in the domain to a value in the range. The Dirac δ-function is a generalized function or a distribution function. The δ-function can be defined in many ways. Here, we use a series of Gaussian functions to define the δ-function. Each of the Gaussian functions (see Figure 1.13) has a unit area underneath its curve, and as the parameter n gets larger, the curve gets narrower and taller (see Figure 1.13):

$$\left(\frac{n}{\pi}\right)^{1/2} \mathrm{e}^{-nx^2}. \tag{1.5.14}$$

Let $f(x)$ be a smooth function that is differentiable everywhere with any order and $\lim\limits_{x\to\infty} x^N f(x) = 0$ for all N. Then the δ-function is defined implicitly as

$$\lim_{n\to\infty}\int_{-\infty}^{\infty}\left(\frac{n}{\pi}\right)^{1/2}\mathrm{e}^{-nx^2}f(x)\mathrm{d}x = \int_{-\infty}^{\infty}\delta(x)f(x)\mathrm{d}x = f(0). \tag{1.5.15}$$

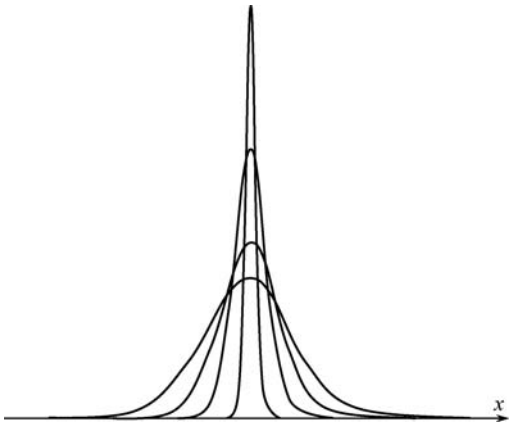


Fig. 1.13. Using a train of Gaussian functions to define the δ-function.

The δ-function has some properties:

$$\int_{-\infty}^{\infty} \delta(x-a)f(x)\mathrm{d}x = \int_{-\infty}^{\infty} \delta(x)f(x+a)\mathrm{d}x = f(a), \tag{1.5.16}$$

$$\int_{-\infty}^{\infty} \delta(ax)f(x)\mathrm{d}x = \frac{1}{|a|}f(0), \tag{1.5.17}$$

$$\int_{-\infty}^{\infty} \delta^{(n)}(x)f(x)\mathrm{d}x = (-1)^n f^{(n)}(0) \text{ [the } n\text{th order derivative]}, \tag{1.5.18}$$

$$\delta(g(x))f(x) = \sum_n \frac{1}{|g'(\lambda_n)|}\delta(x-\lambda_n), \tag{1.5.19}$$

where λ_n's are the zeros of $g(x)$.

In 2D and 3D cases, $\delta(\boldsymbol{x}) = \delta(x)\delta(y)$ and $\delta(\boldsymbol{x}) = \delta(x)\delta(y)\delta(z)$, respectively. In the last property, $|g'|$ will be replaced by $|grad(g)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$ and $|grad(g)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2 + \left(\frac{\partial g}{\partial z}\right)^2}$, respectively, in 2D and 3D.

In 2D imaging, we use a 2D δ-function $\delta(\boldsymbol{x}-\boldsymbol{x}_0)$ to represent a point source at location $\boldsymbol{x} = \boldsymbol{x}_0$. The Radon transform of $f(\boldsymbol{x}) = \delta(\boldsymbol{x}-\boldsymbol{x}_0) = \delta(x-x_0)\delta(y-y_0)$ is given as

$$p(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\boldsymbol{x})\delta(\boldsymbol{x}\cdot\boldsymbol{\theta}-s)\mathrm{d}\boldsymbol{x}, \tag{1.5.20}$$

$$p(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(x-x_0)\delta(y-y_0)\delta(x\cos\theta + y\sin\theta - s)\mathrm{d}x\mathrm{d}y, \tag{1.5.21}$$

$$p(s,\theta) = \int_{-\infty}^{\infty} \delta(y - y_0) \left[\int_{-\infty}^{\infty} \delta(x - x_0)\delta(x\cos\theta + y\sin\theta - s)\mathrm{d}x \right] \mathrm{d}y, \tag{1.5.22}$$

$$p(s,\theta) = \int_{-\infty}^{\infty} \delta(y - y_0)\delta(x_0\cos\theta + y\sin\theta - s)\mathrm{d}y, \tag{1.5.23}$$

$$p(s,\theta) = \delta(x_0\cos\theta + y_0\sin\theta - s). \tag{1.5.24}$$

which is a sinogram similar to that shown in Figure 1.6.

1.6 Worked Examples

Example 1 If you see two separate trees on both views, can you uniquely reconstruct the map of trees (see Figure 1.14)? If not, you may need to take more pictures. If you are only allowed to take one more picture, at which direction should you take the picture?

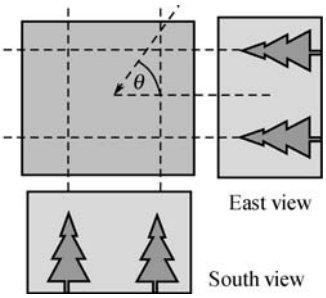


Fig. 1.14. Two trees can be seen on both views.

Solution

Both of the following two situations can satisfy the two views:

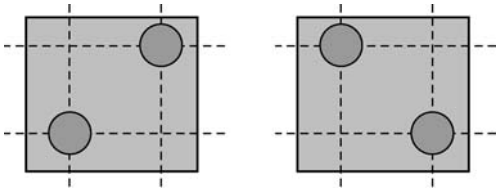


Fig. 1.15. Two potential solutions for the mapping problem.

If we take another picture at 45°, we are able to solve the ambiguity.

Example 2 Find the projections of a uniform disc. The center of the disc is not at the center of detector rotation.

Solution

We already know that if the center of the disc is at the center of detector rotation, the projection can be evaluated as

$$p(s) = \begin{cases} \rho t = 2\rho\sqrt{R^2 - s^2}, & |s| < R; \\ 0, & |s| \geqslant R. \end{cases} \tag{1.6.1}$$

Without loss of generality, we now assume that the center of the disc is on the positive x-axis with the coordinates $(r, 0)$.

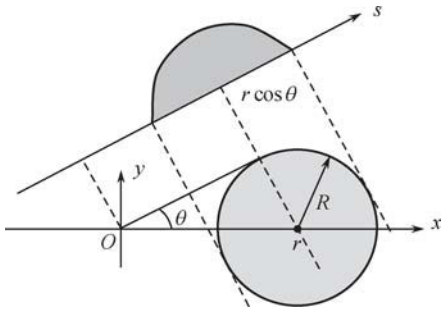


Fig. 1.16. Imaging of an off-centered disc.

For this new setup, we need to shift the projection data on the s-axis. The shifting distance is $r\cos\theta$. That is

$$p(s) = \begin{cases} 2\rho\sqrt{R^2 - (s - r\cos\theta)^2}, & |s - r\cos\theta| < R; \\ 0, & |s - r\cos\theta| \geqslant R. \end{cases} \tag{1.6.2}$$

Example 3* Show that the parallel-beam data redundancy condition is $p(s, \theta) = p(-s, \theta + \pi)$.

Proof

Using the projection definition in Section 1.5, we have

$$\begin{aligned} p(-s, \theta + \pi) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(x\cos(\theta + \pi) + y\sin(\theta + \pi) - (-s))\mathrm{d}x\mathrm{d}y \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\delta(-x\cos\theta - y\sin\theta + s)\mathrm{d}x\mathrm{d}y \end{aligned}$$

$$
\begin{aligned}
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(-(x\cos\theta + y\sin\theta - s))\mathrm{d}x\mathrm{d}y \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - s)\mathrm{d}x\mathrm{d}y \\
&= p(s,\theta). \qquad (1.6.3)
\end{aligned}
$$

(The δ function is an even function.)

Example 4* Show that the point spread function of the projection/backprojection operator is $1/r$, where $r = \left\|\boldsymbol{x} - \boldsymbol{x}_0\right\|$ and the point source object is $f(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_0)$.

Proof

Using the definition of the backprojection, we have

$$
b(\boldsymbol{x}) = \int_0^{\pi} p(\boldsymbol{x}\cdot\boldsymbol{\theta},\theta)\mathrm{d}\theta = \int_0^{\pi}\int_{-\infty}^{\infty} f((\boldsymbol{x}\cdot\boldsymbol{\theta})\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp})\mathrm{d}t\mathrm{d}\theta. \qquad (1.6.4)
$$

We realize that the line integral $\displaystyle\int_{-\infty}^{\infty} f((\boldsymbol{x}\cdot\boldsymbol{\theta})\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp})\mathrm{d}t$ is along the line that passes through the point $\boldsymbol{x}$ and in the direction of $\boldsymbol{\theta}^{\perp}$ (see Figure 1.17), we have

$$
\int_{-\infty}^{\infty} f((\boldsymbol{x}\cdot\boldsymbol{\theta})\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp})\mathrm{d}t = \int_{-\infty}^{\infty} f(\boldsymbol{x} - \hat{t}\boldsymbol{\theta}^{\perp})\mathrm{d}\hat{t}. \qquad (1.6.5)
$$

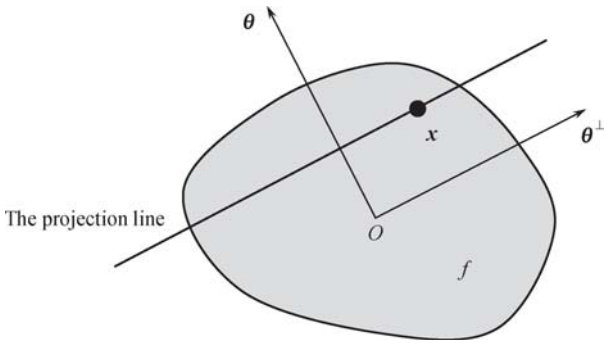


Fig. 1.17. The line integral is performed on a line passing through the backprojection point.

Therefore the projection/backprojection image can be obtained as

$$
b(\boldsymbol{x}) = \int_0^{\pi}\int_{-\infty}^{\infty} f(\boldsymbol{x} - \hat{t}\boldsymbol{\theta}^{\perp})\mathrm{d}\hat{t}\mathrm{d}\theta. \qquad (1.6.6)
$$

Let $\hat{\boldsymbol{x}} = \hat{t}\boldsymbol{\theta}^{\perp}$ with $\|\hat{\boldsymbol{x}}\| = |\hat{t}|$, $\mathrm{d}\hat{\boldsymbol{x}} = |\hat{t}|\,\mathrm{d}\hat{t}\mathrm{d}\theta$. The above expression becomes

$$b(\boldsymbol{x}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{f(\boldsymbol{x}-\hat{\boldsymbol{x}})}{\|\hat{\boldsymbol{x}}\|}\mathrm{d}\hat{\boldsymbol{x}}. \tag{1.6.7}$$

Let $f(\boldsymbol{x}) = \delta(\boldsymbol{x}-\boldsymbol{x}_0)$. The point spread function of the projection/backprojection operator is

$$b(\boldsymbol{x}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{\delta(\boldsymbol{x}-\boldsymbol{x}_0-\hat{\boldsymbol{x}})}{\|\hat{\boldsymbol{x}}\|}\mathrm{d}\hat{\boldsymbol{x}} = \frac{1}{\|\boldsymbol{x}-\boldsymbol{x}_0\|} = \frac{1}{r}. \tag{1.6.8}$$

Example 5* Evaluate $\displaystyle\int_{-\infty}^{\infty} \delta(\mathrm{e}^{2(x-3)(x+4)}-1)f(x)\mathrm{d}x$.

Solution

Let

$$g(x) = \mathrm{e}^{2(x-3)(x+4)} - 1. \tag{1.6.9}$$

Solving $g(x) = \mathrm{e}^{2(x-3)(x+4)} - 1 = 0$, we obtain the zeros of $g(x)$ as

$$\lambda_1 = 3 \text{ and } \lambda_2 = -4. \tag{1.6.10}$$

The derivative of $g(x)$ is

$$g'(x) = 2[(x-3)+(x+4)]\mathrm{e}^{2(x-3)(x+4)} = 2(2x+1)\mathrm{e}^{2(x-3)(x+4)}. \tag{1.6.11}$$

Thus, at the two zeros of $g(x)$, we have

$$g'(3) = 14 \text{ and } g'(-4) = -14. \tag{1.6.12}$$

We have

$$\begin{aligned}
&\int_{-\infty}^{\infty} \delta(\mathrm{e}^{2(x-3)(x+4)}-1)f(x)\mathrm{d}x \\
&= \int_{-\infty}^{\infty} \frac{\delta(x-3)}{|14|}f(x)\mathrm{d}x + \int_{-\infty}^{\infty} \frac{\delta(x+4)}{|14|}f(x)\mathrm{d}x \\
&= \frac{f(3)+f(-4)}{14}.
\end{aligned} \tag{1.6.13}$$

1.7 Summary

- Tomography is a process of taking projection data and converting the data into cross-section images. Projection data from multiple views are required.

- A projection is a line integral (or ray-sum, Radon transform) of an object. Projection data are acquired with detectors. Objects overlap on the detectors.
- Backprojection is a superposition procedure and it sums the data from all projection views. Backprojection evenly distributes the projection domain data back along the same lines from which the line-integrals were formed.
- Image reconstruction is a mathematical procedure that dissolves the overlapping effect in the projection data and creates a non-overlapped image of the original image. A mathematical procedure is called an algorithm.
- Dirac's δ-function usually acts as a point source in algorithm development.
- The readers are expected to understand two main concepts in this chapter: projection and backprojection.

Problems

Problem 1.1 If a 2D object is a point source, it is sufficient to use the projection data from two different detector views to obtain an exact reconstruction. Let us consider a 2D object that consists of three point sources which are not on a same straight line (i.e., they are not co-linear). Determine the smallest number of detector views so that sufficient projection data are available to obtain an exact reconstruction.

Problem 1.2 It is known that the Radon transform of a shifted point source $\delta(x - x_0, y - y_0)$ is $\delta(x_0 \cos\theta + y_0 \sin\theta - s)$. This result can be extended to a general object $f(x, y)$. If $p(s, \theta)$ is the Radon transform of the un-shifted object $f(x, y)$, determine the Radon transform of the shifted object $f(x - x_0, y - y_0)$.

Problem 1.3 Use the definition of the δ function to prove that the following two definitions of the Radon transform are equivalent:

$$p(s,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - s)\mathrm{d}x\mathrm{d}y,$$

$$p(s,\theta) = \int_{-\infty}^{\infty} f(s\cos\theta - t\sin\theta, s\sin\theta + t\cos\theta)\mathrm{d}t.$$

Problem 1.4 The backprojection in the Cartesian coordinate system is defined as

$$b(x,y) = \int_0^{\pi} p(x\cos\theta + y\sin\theta, \theta)\mathrm{d}\theta.$$

Give an equivalent expression $b_{polar}(r, \varphi)$ of the backprojection in the polar coordinate system.

References

1. Barrett H, Swindell W (1988) Radiological imaging. Academic, New York
2. Bracewell RN (1955) Two-dimensional imaging. Prentice Hall, Englewood Cliffs, New Jersey
3. Bushberg JT, Seibert JA, Leidoldr EM, Boone JM (2002) The essential physics of medical imaging, 2nd edn. Lippincott Williams and Wilkins, Philadelphia
4. Carlton RR, Adler AM (2001) Principles of radiographic imaging: An Art and a Science, 3rd edn. Delmar, New York
5. Cho ZH, Jones JP, Signh M (1993) Foundations of medical imaging. Wiley, New York
6. Cormack AM (1963) Representation of a function by its line integrals, with some radiological applications. J Appl Phys 34: 2722 – 2727
7. Herman G (1981) Image reconstruction from projections: the fundamentals of computerized tomography. Academic, New York
8. Kak AC, Staney M (1987) Principles of computerized tomography. IEEE Press, Piscataway, New Jersey
9. Macovski A (1983) Medical imaging systems. Prentice Hall, Englewood Cliffs, New Jersey
10. Natterer F (1986) The mathematics of computerized tomography. Wiley-Teubner, New York
11. Prince JL, Links JM (2006) Medical imaging signals and system. Pearson Prentice Hall, Upper Saddle River, New Jersey
12. Shung KK, Smith MB, Tsui BMW (1992) Principles of medical imaging. Academic, San Diego, California
13. Zeng GL (2001) Image reconstruction — a tutorial. Computerized Medical Imaging and Graphics 25: 97 – 103

2 Parallel-Beam Image Reconstruction

This chapter introduces the central slice theorem, which is the function of tomography. This theorem relates the 2D image with its 1D projections in the Fourier domain. From this theorem, many image reconstruction algorithms are derived. Among these algorithms, the filtered backprojection (FBP) algorithm is the most popular one, which consists of a ramp-filtering step and a backprojection step. The filtering can be implemented as a multiplication in the Fourier domain or as a convolution in the spatial domain. A recent hot topic of region-of-interest (ROI) reconstruction with truncated projections is included in this chapter.

2.1 Fourier Transform

The concept of the Fourier transform is based on the fact that it is possible to form a function $p(s)$ as a weighted summation of a series of sine and cosine terms of various frequencies, ω, with a weighting function $P(\omega)$. You can use a prism to decompose the sunlight into a spectrum of different colors; you can also re-produce the original light by recombining the spectrum of different colors (see Figure 2.1).

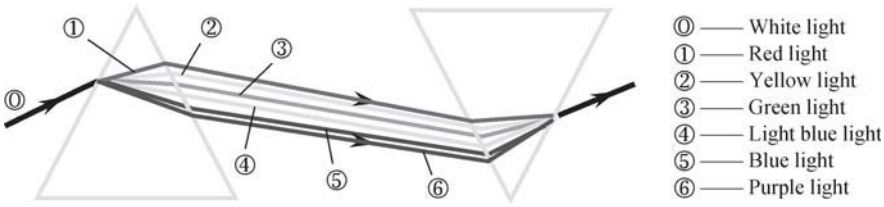


Fig. 2.1. The white light can be decomposed into color lights, which can be converted back the original white light.

The weighting function $P(\omega)$ for each frequency ω is called the Fourier transform of $p(s)$. One can easily use mathematical formulas to find $P(\omega)$ from $p(s)$ and to recover $p(s)$ from $P(\omega)$. If you know one function (either $p(s)$ or $P(\omega)$), you know the other. In this pair, one function is denoted by a lower case letter with a variable s, and the other function is denoted by an upper case letter with a variable ω.

A Fourier transform pair $p(s)$ and $P(\omega)$ is shown in Figure 2.2, where $P(\omega)$ is the Fourier transform of $p(s)$. The function $P(\omega)$ tells us that the triangle function $p(s)$ has rich low frequency components because the central lobe has a large amplitude. As the frequency gets higher (that is, as $|\omega|$ gets larger), the amplitude of the lobes becomes smaller. We also see some notches in $P(\omega)$; those notched frequencies are not in the triangle function $p(s)$. Using the Fourier transform can help us understand some hidden mathematical relationships, which are not easy to see without the Fourier transform.

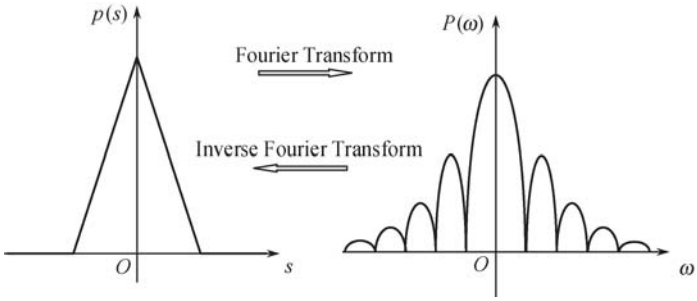


Fig. 2.2. A Fourier transform pair.

One can also find the Fourier transform for a function with two or more variables. We will denote the Fourier transform of the function $f(x, y)$ as $F(\omega_x, \omega_y)$, where ω_x is the frequency in the x direction and ω_y is the frequency in the y direction.

2.2 Central Slice Theorem

Central slice theorem is the foundation of tomography. It has other names: projection slice theorem and Fourier slice theorem. The central slice theorem in two-dimensions (2D) states that the 1D Fourier transform $P(\omega)$ of the projection $p(s)$ of a 2D function $f(x, y)$ is equal to a slice (i.e., a 1D profile) through the origin of the 2D Fourier transform $F(\omega_x, \omega_y)$ of that function which is parallel to the detector (see Figure 2.3).

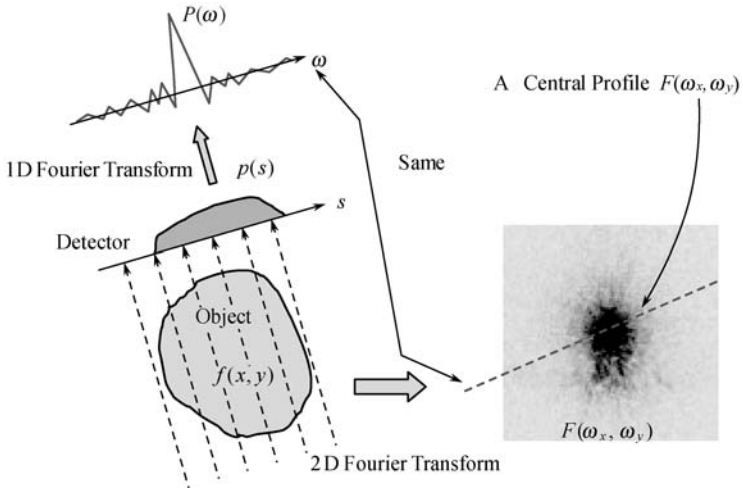


Fig. 2.3. Illustration of the 2D central slice theorem.

If we rotate the detector around the object at least for 180°, the corresponding "central slice" in the 2D Fourier transform $F(\omega_x, \omega_y)$ will rotate synchronously and will cover the entire 2D Fourier space, that is, the ω_x-ω_y plane (see Figure 2.4). In other words, by rotating the detector 180°, the entire 2D Fourier transform $F(\omega_x, \omega_y)$ is "measured." Once $F(\omega_x, \omega_y)$ is available, the original 2D function $f(x, y)$ can be readily obtained by a mathematical procedure called the 2D inverse Fourier transform.

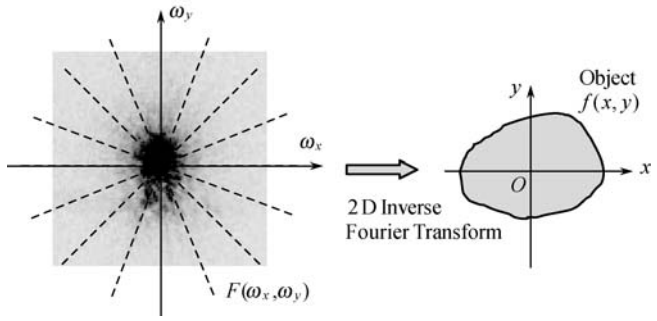


Fig. 2.4. Each view adds a line in the Fourier space. The 2D inverse Fourier transform reconstructs the original image.

Backprojecting projection data at one view are equivalent to adding a "central slice" of $F(\omega_x, \omega_y)$ in the ω_x-ω_y plane (i.e., the Fourier space). Backprojection over 180° fully reconstructs the 2D Fourier transform $F(\omega_x, \omega_y)$. Due to the property of the Fourier pair, the original function $f(x, y)$ can be

readily found from $F(\omega_x, \omega_y)$.

In Chapter 1, we learned that backprojection alone does not get you the original image back; instead, a blurred image is obtained. Is there a contradiction here? No. Let us take another look of the ω_x-ω_y plane in Figure 2.4. After we add the "central slices" to the ω_x-ω_y plane, we see higher density of the "central slices" at the origin of the ω_x-ω_y plane and lower density at the regions away from the origin. The central region of the Fourier space represents low frequencies. Over-weighting with low frequency components blurs the image.

To counter this blurring effect, we must compensate for the non-uniformity in the Fourier space. The non-uniform density in the Fourier space is proportional to

$$\frac{1}{\sqrt{\omega_x^2 + \omega_y^2}}.$$

There are two ways to do the compensation.

One way is to multiply the ω_x-ω_y space Fourier "image" by $\sqrt{\omega_x^2 + \omega_y^2}$, and the resultant Fourier space "image" is $F(\omega_x, \omega_y)$. The 2D inverse Fourier transform of $F(\omega_x, \omega_y)$ gives the exact original image $f(x, y)$.

The other way is to multiply the 1D Fourier transform $P(\omega, \theta)$ of the projection data by $|\omega|$. We then take the 1D inverse Fourier transform of $|\omega|P(\omega, \theta)$. After this special treatment (i.e., filtering) of the projection data, the treated (i.e., filtered) projection data are backprojected and the exact original image $f(x, y)$ is obtained. Note that in the previous discussion (see Figure 2.3) we ignored the second variable in $P(\omega, \theta)$ and $p(s, \theta)$ on purpose to allow the reader to pay attention to the first variable.

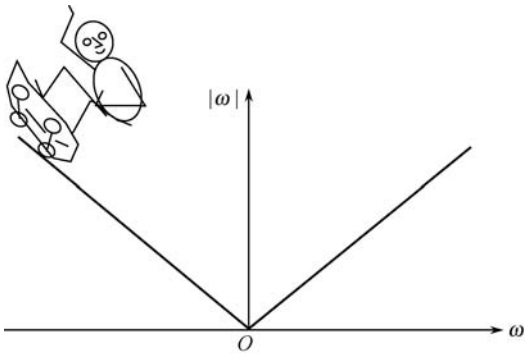


Fig. 2.5. The 1D ramp filter transfer function.

These two methods (i.e., multiplying the Fourier transformed backprojected image by $\sqrt{\omega_x^2 + \omega_y^2}$; multiplying the Fourier transformed projections

by $|\omega|$) will be further discussed later in this book. The second method, which is referred to as the filtered backprojection (FBP) algorithm, is more popular than the first. The function $|\omega|$ is called the *ramp filter* in tomography, named after its appearance (see Figure 2.5). It is the second method that added negative "wings" around the spikes in the projections in Chapter 1.

2.3 Reconstruction Algorithms

Backprojection accomplishes most of the work in image reconstruction; it converts the projection data at various views into an image, which is almost what we wanted except for the blurring effect. The blurring effect is caused by the $1/|\omega|$ non-uniform weighting, with $|\omega| = \sqrt{\omega_x^2 + \omega_y^2}$, in the 2D Fourier space (i.e., the ω_x-ω_y plane). Compensation for this $1/|\omega|$ function can be realized by filtering the projection data $p(s, \theta)$ or equivalently, its 1D Fourier transform $P(\omega, \theta)$ with a ramp-filter $|\omega|$. If the filtered projection data are backprojected, the exact image can be obtained.

2.3.1 Method 1

Precisely, this filtered backprojection (FBP) algorithm can be implemented according to the steps shown in Figure 2.6, in which the ramp filtering is implemented as:

(1) Find the 1D Fourier transform of $p(s, \theta)$ with respect to the first variable s, obtaining $P(\omega, \theta)$.

(2) Multiply $P(\omega, \theta)$ with a ramp filter $|\omega|$, obtaining $Q(\omega, \theta)$.

(3) Find the 1D inverse Fourier transform of $Q(\omega, \theta)$ with respect to the first variable ω, obtaining $q(s, \theta)$.

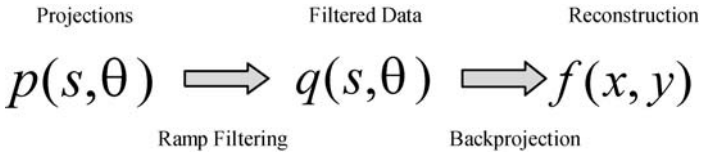


Fig. 2.6. The procedure of the filtered backprojection (FBP) algorithm.

2.3.2 Method 2

There are more ways than one to do ramp filtering. In fact, we can perform ramp filtering without using the Fourier transform at all. According to Fourier transform theory, multiplication in one domain (say, the ω domain) corresponds to convolution in the other domain (say, the s domain) (see Figure 2.7).

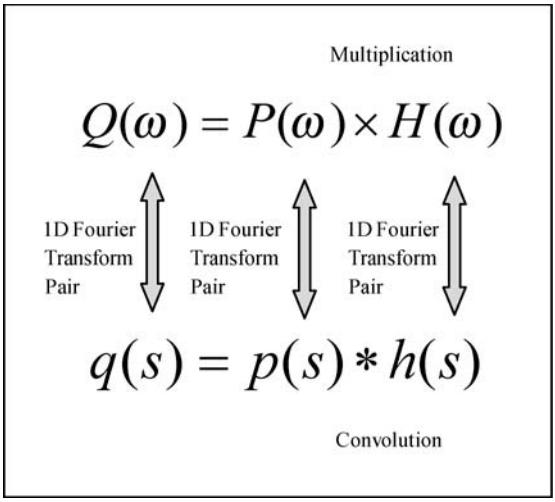


Fig. 2.7. An important property of Fourier transform: multiplication in one domain is equivalent to convolution in the other domain.

Thus Steps (1), (2), and (3) in Method 1 are equivalent to a mathematical procedure called *convolution*. The ramp-filtered data $q(s, \theta)$ can be obtained by convolution as

$$q(s, \theta) = p(s, \theta) * h(s), \tag{2.3.1}$$

where "$*$" denotes the convolution operation, which is an integral with respect to the variable s. Here $h(s)$ is the convolution kernel and is the 1D inverse Fourier transform of $H(\omega) = |\omega|$.

Here we give an example of convolution so that you can understand what convolution can do for you. If the convolution kernel $h(s)$ is not symmetric, you first flip it left-right, making it $h(-s)$. Second, you imagine that your function $p(s)$ can be decomposed into vertical spikes (or Dirac delta functions). Third, replace each spike by $h(-s)$, that is, the $s = 0$ position of $h(-s)$ is at the spike's position and scale $h(-s)$ by the amplitude (which can

be negative) of the spike. Finally, sum up all the shifted and scaled versions of $h(-s)$, obtaining $q(s)$ (see Figure 2.8).

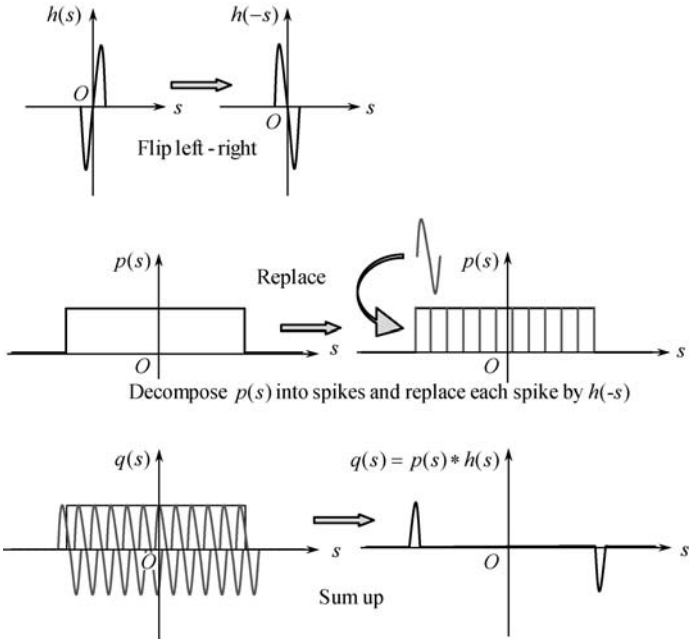


Fig. 2.8. An illustration of the procedure of convolution.

2.3.3 Method 3

There is a third way to implement ramp filtering. Let us factor the ramp filter into two parts:

$$H(\omega) = |\omega| = \mathrm{i}\,2\pi\omega \times \frac{1}{\mathrm{i}\,2\pi}\mathrm{sgn}(\omega), \tag{2.3.2}$$

where , $\mathrm{i} = \sqrt{-1}$ and

$$\mathrm{sgn}(\omega) = \begin{cases} 1, & \omega > 0, \\ 0, & \omega = 0, \\ -1, & \omega < 0. \end{cases} \tag{2.3.3}$$

Here we will use two properties of the Fourier transform.

Fact 1: Multiplication by $\mathrm{i}2\pi\omega$ in the Fourier domain (i.e., the ω domain) corresponds to the derivative with respect to s in the spatial domain (i.e., the s domain).

Fact 2: The inverse Fourier transform of $-\mathrm{i}\ \mathrm{sgn}(\omega)$ is $1/(\pi s)$. Convolution with $1/(\pi s)$ is called the Hilbert transform.

Using the relationship shown in Figure 2.7, the ramp-filtering can be realized as

$$q(s,\theta) = \frac{\mathrm{d}p(s,\theta)}{\mathrm{d}s} * \frac{-1}{2\pi^2 s}, \tag{2.3.4}$$

which is a combination of the derivative and the Hilbert transform.

2.3.4 Method 4

If we switch the order of ramp-filtering and backprojection, we can get another way to reconstruct the image — *backprojection then filtering*. After backprojection, we get a blurred image $b(x,y)$, which is two-dimensional. We need to apply a 2D ramp filter to it. One way to do it is as follows:

(1) Find the 2D Fourier transform of $b(x,y)$, obtaining $B(\omega_x,\omega_y)$.

(2) Multiply $B(\omega_x,\omega_y)$ with a ramp filter $|\omega| = \sqrt{\omega_x^2+\omega_y^2}$, obtaining $F(\omega_x,\omega_y)$.

(3) Find the 2D inverse Fourier transform of $F(\omega_x,\omega_y)$, obtaining $f(x,y)$.

2.3.5 Method 5

Method 3 consists of three components: the derivative, the Hilbert transform, and the backprojection. If we switch the order, we obtain yet another reconstruction algorithm:

(1) Find the derivative of the projection data $p(s,\theta)$ with respect to s, obtaining $\dfrac{\mathrm{d}p(s,\theta)}{\mathrm{d}s}$.

(2) Backproject $\dfrac{\mathrm{d}p(s,\theta)}{\mathrm{d}s}$ over 180°

(3) Perform the line-by-line Hilbert transform, in the direction parallel to the detector at the 90° position.

You must have noticed that we are playing the order changing game to create more and more reconstruction algorithms. We simply change the order of ramp-filtering and backprojection, or change of the order of the performing derivative, the Hilbert transform, and the backprojection. If we keep playing this game, we can make a list of some possible image reconstruction algorithms (see Figure 2.9). Each algorithm has its advantages and disadvantages.

The table in Figure 2.9 does not exhaust all possibilities. For example, the

Hilbert transform can be implemented as convolution in the spatial domain or as multiplication in the Fourier domain. The Hilbert transform has another form, which is not convolution but is an integral over a finite interval. The finite Hilbert transform has an important application in handling truncated projection data.

You may see that the backprojection is used in all algorithms. This does not have to be the case. You do not have to use a spatial domain backprojector in a reconstruction algorithm. You can use the central slice theorem by assigning the Fourier domain projection data $P(\omega, \theta)$ at the proper ω_x-ω_y locations. This is the Fourier domain implementation of the backprojection. However, this Fourier domain backprojection has limited applications because large interpolation errors could be introduced in polar coordinate (ω, θ) system to Cartesian $(\omega_x$-$\omega_y)$ system transformation.

Method	Step 1	Step 2	Step 3
1	1D Ramp filter with Fourier transform	Backprojection	
2	1D Ramp filter with convolution	Backprojection	
4	Backprojection	2D Ramp filter with Fourier transform	
	Backprojection	2D Ramp filter with 2D convolution	
3	Derivative	Hilbert transform	Backprojection
5	Derivative	Backprojection	Hilbert transform
	Backprojection	Derivative	Hilbert transform
	Hilbert transform	Derivative	Backprojection
	Hilbert transform	Backprojection	Derivative
	Backprojection	Hilbert transform	Derivative

Fig. 2.9. A list of parallel-beam analytical image reconstruction algorithms.

2.4 A Computer Simulation

In Figure 2.10, we show an example of the filtered backprojection algorithm in action. The original image $f(x, y)$ is shown in the lower right corner; it consists of a large disc and four small discs. The projection data $p(s, \theta)$ are generated analytically using computer software. The projections of the two outer small discs trace two sine curves in the sinogram, which is an s-θ coordinate display of the projection data.

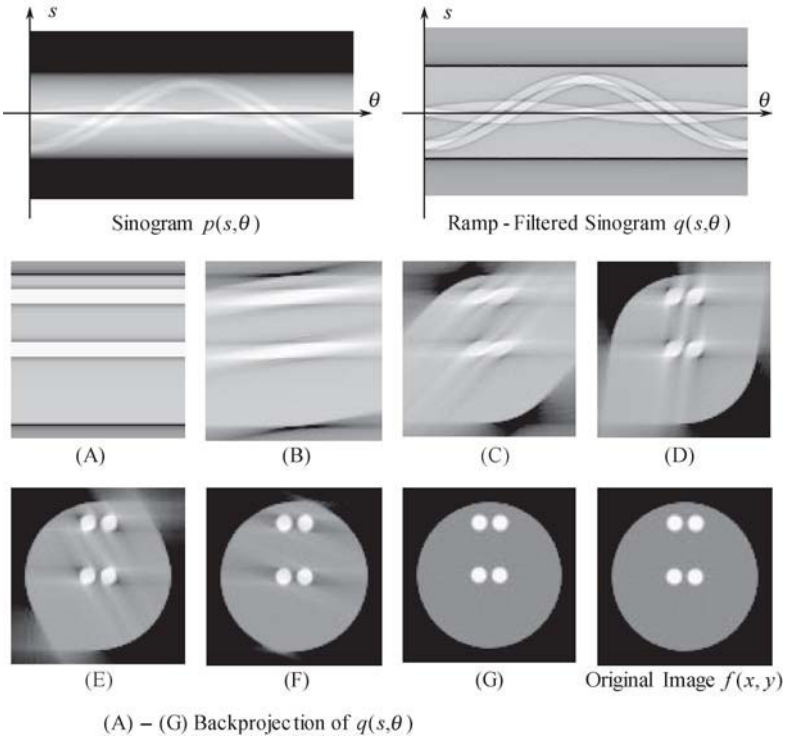


Fig. 2.10. An FBP algorithm in action: filtering and view-by-view backprojection.

After applying the ramp-filter to $p(s, \theta)$, the filtered data $q(s, \theta)$ looks sharper because the ramp-filter, which is a high-pass filter, suppresses the low-frequency components and enhances the high-frequency components. One can see the darker edges around the projection of the discs. These darker edges are the negative "wings" that we discussed in Chapter 1.

The image labeled (A) is the backprojection of $q(s, \theta)$ at the first angle.

The filtered data $q(s, \theta)$ at the first angle are simply copied across the entire image. This action is sometimes described as evenly distribution of a value along the projection path. Progressing through images (A) to (G), as data from more and more angles are backprojected, the image takes shape and gets closer and closer to the original image. Remember that backprojection consists of two actions: smear back and superposition. The image is reconstructed when the backprojection is performed over 180°. The backprojection can also be performed over 360° (then divide the image value by 2), because the data are redundant by observing the fact that $p(s, \theta) = p(-s, \theta + \pi)$ and $q(s, \theta) = q(-s, \theta + \pi)$.

*2.5 ROI Reconstruction with Truncated Projections

The situation of the region-of-interest (ROI) reconstruction with truncated projections is illustrated in Figure 2.11, where the detector is not large enough to cover the entire object, but is large enough to cover the ROI. In tomographic theory, this ROI reconstruction problem is called the interior problem. Only the circular FOV (field-of-view) is fully measured. For a general interior problem, only an approximate solution can be obtained. Exact reconstructions exist only for some special cases.

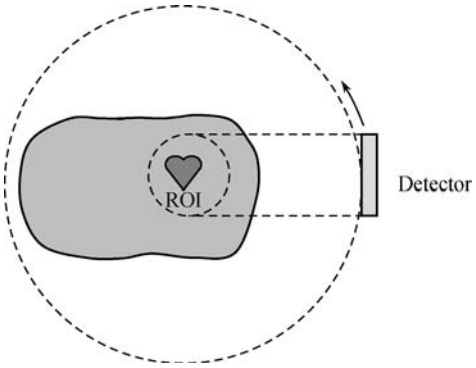


Fig. 2.11. The detector is only large enough to cover the ROI, but not large enough to cover the entire object.

One solvable ROI reconstruction problem is shown in Figure 2.12, and the reconstruction is provided by the "derivative-backprojection-Hilbert transform" algorithm. The ROI is the dark shaded smaller region within the circular field-of-view (FOV). Both the derivative operation and the backprojec-

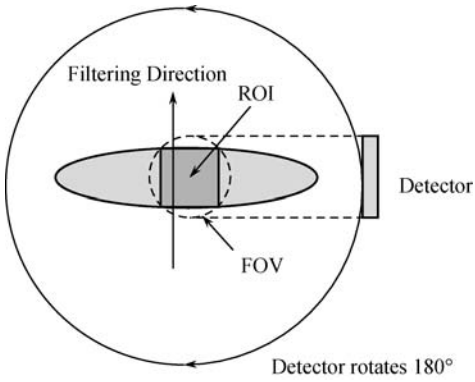


Fig. 2.12. The derivative-backprojection-Hilbert transform algorithm is able to exactly reconstruct the ROI.

tion operation are local. That is, they only use the locally available data to find the results. After the derivative and backprojection, the data within the FOV are exact.

The next step in the reconstruction algorithm is to perform the line-by-line 1D Hilbert transform along the direction indicated as the vertical direction in Figure 2.12. The usual Hilbert transform is in the form of convolution with a convolution kernel $1/s$ that does not vanish. Therefore, the Hilbert transform requires data along the entire filtering line.

Thanks to a finite inverse Hilbert transform formula, we are able to perform the inverse Hilbert transform with finite data. If a function $f(s)$ is non-zero on the interval $[a, b]$ and if the Hilbert transform of $f(s)$ is $g(s)$, then the finite inverse Hilbert transform formula only needs $g(s)$ on $[a, b]$ to recover $f(s)$.

Using this finite inverse Hilbert transform formula, we are now able to evaluate the 1D inverse Hilbert transform along the filtering line to reconstruct the image, because the required exact data are available in the FOV. This finite inverse Hilbert transform formula is not in the form of convolution; therefore, it is not efficient to perform this step using the Fourier transform methods.

It has been shown that an exact ROI reconstruction can be obtained for a less restricted case shown in Figure 2.13. For this case, the ROI has the same size as the FOV. Here, only one end of the filtering line is required to be outside the object. The proof of exact reconstruction for this case is based on analytic continuation. The drawback of this method is that we have not found a closed-form formula to reconstruct the ROI image.

An immediate extension of this analytic continuation method can be applied to the general interior problem, where a small region of the image within the FOV is exactly known in advance (see Figure 2.14). The known

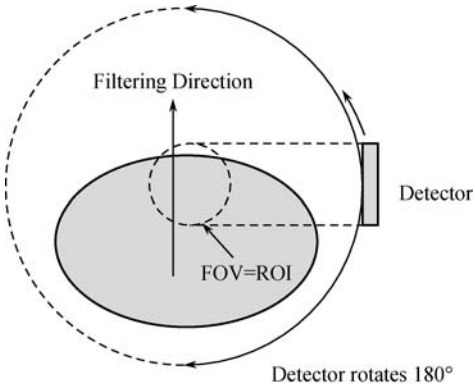


Fig. 2.13. The measured data are sufficient to exactly reconstruct the FOV. A part of the FOV is outside the object.

region in the image can be very small.

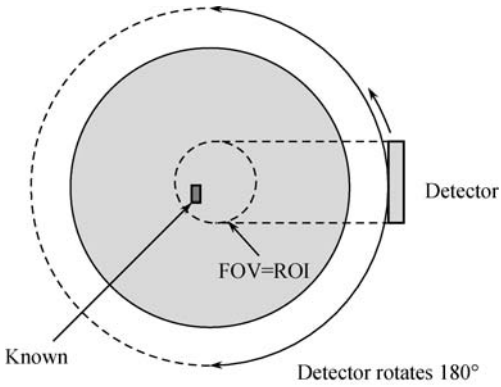


Fig. 2.14. If a small region in the FOV is known, the FOV can be exactly reconstructed.

We now explain what we mean by analytic continuation. The analytic continuation is a subject in a mathematical branch called complex analysis, which studies functions with complex variables.

If x is a real variable, $h(x) = 2x + \mathrm{i}\,(3 + 5x)$, with $\mathrm{i} = \sqrt{-1}$, is not a function with a complex variable. However, $h(z) = 2z$, with $z = x + \mathrm{i}\,y$ and both x and y are real, is a function with a complex variable and is defined in a region (called domain) in the complex plane.

A complex function $h(z)$ is said to be analytic in a complex region R, if and only if the function $h(z)$ is differentiable at every point in R. If a complex function is analytic on a region R, it is *infinitely* differentiable in R. Therefore, you can have a power expansion of an analytic function $h(z)$

anywhere in R. You can imagine that an analytic function is very smooth.

Let $h_1(z)$ and $h_2(z)$ be analytic functions in regions R_1 and R_2, respectively, and suppose that the intersection $R_1 \cap R_2$ is not empty and that $h_1(z) = h_2(z)$ on $R_1 \cap R_2$. Then $h_2(z)$ is called an analytic continuation of $h_1(z)$ to R_2, and vice versa. Moreover, the analytic continuation of $h_1(z)$ to R_2 is unique. This uniqueness of analytic continuation is a rather amazing and extremely powerful statement. If a complex function $h(z)$ is analytic in a region R, knowing the value of $h(z)$ in a very small sub-region of R uniquely determines the value of the function $h(z)$ at every other point in R.

If our 2D image $f(x, y)$ is known in a small region Ω_{known}, we can somehow determine the image $f(x, y)$ in a larger region Ω. Note that $f(x, y)$ is not an analytic function. We do not perform analytic continuation on $f(x, y)$.

First, we draw a line, say, the x-axis, passing through the small region Ω_{known}, as shown in Figure 2.15. Without loss of generality, we only consider the image values on one line: $y = 0$. We can do the same for other lines that pass through the known region Ω_{known}.

It is given that the image $f(x, 0)$ is known when $-r \leqslant x \leqslant r$. In the following, we are going define a function $h(x)$, convert it into a complex function $h(z)$, and apply analytic continuation to $h(z)$ so that the image $f(x, 0)$ can be estimated in a larger interval $(-1, 1)$.

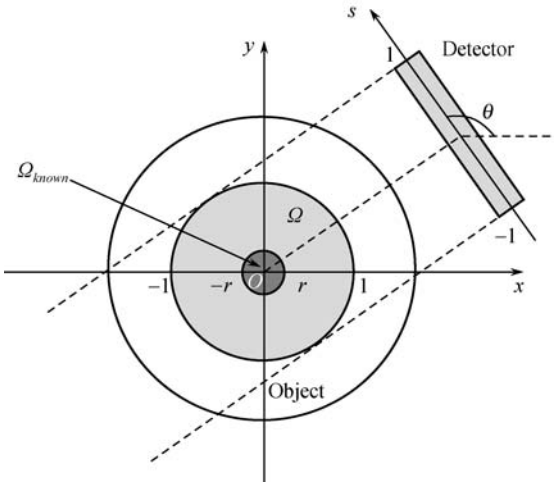


Fig. 2.15. The known image values on $[-r, r]$ together with the truncated projections can be used to determine image values in $(-1, 1)$.

In this particular imaging problem, the detector is smaller than the object. Projection data are only available for $-1 \leqslant s \leqslant 1$. We can use the FBP algorithm to reconstruct the image as

$$f(x,0)=\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{-\infty}^{\infty}\frac{1}{s-x\cos\theta}\frac{\partial p(s,\theta)}{\partial s}\mathrm{d}s. \tag{2.5.1}$$

This expression can be written in two terms, the first term with measured data and the second term with unmeasured data:

$$\begin{aligned}f(x,0)=&\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{-1}^{1}\frac{1}{s-x\cos\theta}\frac{\partial p(s,\theta)}{\partial s}\mathrm{d}s\\&+\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{|s|>1}\frac{1}{s-x\cos\theta}\frac{\partial p(s,\theta)}{\partial s}\mathrm{d}s.\end{aligned} \tag{2.5.2}$$

The first term can be evaluated using the measured data on $[-1,1]$ and it contains the sharp edges of the image f. However, the second term is very smooth (almost a constant) but cannot be evaluated because the data are not available. Let us denote the second term as $h(x)$. We have

$$h(x)=\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{|s|>1}\frac{1}{s-x\cos\theta}\frac{\partial p(s,\theta)}{\partial s}\mathrm{d}s,\quad |x|<1. \tag{2.5.3}$$

In the above expression, $|x|<1$ and $|s|>1$make $1/(s-x\cos\theta)$ differentiable of any order. We further assume that $\partial p/\partial s$ is absolutely integrable, which is usually satisfied in practice. We claim that if we replace the real variable x by a complex variable z, $h(z)$ is an analytic function within $|z|<1$.

The analytic function $h(z)$ is known on $[-r,r]$, because $h(x)$ can also be expressed as

$$h(x)=f(x,0)-\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{-1}^{1}\frac{1}{s-x\cos\theta}\frac{\partial p(s,\theta)}{\partial s}\mathrm{d}s, \tag{2.5.4}$$

$f(x,0)$ is given on $[-r,r]$, and the second term in the above expression can be evaluated with the measured data. By using analytic continuation, $h(z)$ is uniquely determined in $|z|<1$. Thus, $h(x)$ is determined in $(-1,1)$. Once $h(x)$ is found, the image value in $(-1,1)$ can be obtained as

$$f(x,0)=\frac{1}{4\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{-1}^{1}\frac{1}{s-x\cos\theta}\frac{\partial p(s,\theta)}{\partial s}\mathrm{d}s+h(x),\quad |x|<1. \tag{2.5.5}$$

In the above discussion, we know that we should have enough data to determine the image in region Ω. However, doing analytic continuation is easier said than done. Mathematically, the right way to do analytic continuation is by doing Taylor expansions, but this approach is not practical. Currently iterative approaches are used for this purpose.

Another practical problem is the stability issue. In some cases we can prove mathematically that the limited data can uniquely determine an image, but the image reconstruction method can be extremely ill-conditioned and

unstable. The limited angle imaging problem could be such an ill-conditioned problem. In a limited angle imaging problem, the ROI does not have full 180° angular measurements. It may only have, for example, 10° angular measurements.

*2.6 Mathematical Expressions

This section presents the mathematical expressions of the 1D Fourier transform pair, the Hilbert transform pair, and two versions of the inverse finite Hilbert transform. The proof of the central slice theorem is given. This section also derives the filtered backprojection and backprojection-then-filtering algorithms. The spatial domain filtering methods, such as the convolution backprojection algorithm and the Radon inversion formula, are also included.

2.6.1 The Fourier Transform and Convolution

The 1D Fourier transform is defined as:

$$P(\omega) = \int_{-\infty}^{\infty} p(s) \mathrm{e}^{-2\pi \mathrm{i} s\omega} \mathrm{d}s, \tag{2.6.1}$$

and the 1D inverse Fourier transform is

$$p(s) = \int_{-\infty}^{\infty} P(\omega) e^{2\pi \mathrm{i} s\omega} \mathrm{d}\omega. \tag{2.6.2}$$

The convolution of two functions f and g is defined as

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) \mathrm{d}\tau = \int_{-\infty}^{\infty} f(t-\tau) g(\tau) \mathrm{d}\tau. \tag{2.6.3}$$

If we denote the Fourier transform operator as $\mathbf{F}$, the convolution theorem can be stated as

$$\mathbf{F}(f * g) = \mathbf{F}(f) \times \mathbf{F}(g). \tag{2.6.4}$$

2.6.2 The Hilbert Transform and the Finite Hilbert Transform

The Hilbert transform is less well known than the Fourier transform. Unlike the Fourier transform that converts a real function into a complex function, the Hilbert transform converts a real function into another real function.

In this section, we use $\mathbf{H}$ to denote the Hilbert transform operator. If you apply the Hilbert transform twice, you get the original function back, except for a sign change, that is

$$\mathbf{H}(\mathbf{H}f) = \mathbf{H}^2 f = -f. \tag{2.6.5}$$

In other words, the inverse Hilbert transform is the negative value of the Hilbert transform.

A real function and its Hilbert transform are orthogonal. If $g = \mathbf{H}\, f$, then

$$\int_{-\infty}^{\infty} f(t)g(t)\mathrm{d}t = 0. \tag{2.6.6}$$

For example, the Hilbert transform of $\cos(t)$ is $\sin(t)$. The Hilbert transform of $\sin(t)$ is $-\cos(t)$.

The Hilbert transform can be defined in the Fourier domain. Let the Fourier transform of a real function $f(t)$ be $F(\omega)$, the Hilbert transform of $f(t)$ be $g(t)$, and the Fourier transform of $g(t)$ be $G(\omega)$, then

$$G(\omega) = \mathrm{i}\,\mathrm{sgn}(\omega)F(\omega), \tag{2.6.7}$$

with $\mathrm{i} = \sqrt{-1}$ and $\mathrm{sgn}(\omega)$ being the signum:

$$\mathrm{sgn}(\omega) = \begin{cases} 1, & \omega > 0, \\ 0, & \omega = 0, \\ -1, & \omega < 0. \end{cases} \tag{2.6.8}$$

Since the magnitude of $-\mathrm{i}\,\mathrm{sgn}(\omega)$ is, the Hilbert transform is an application of an all-pass filter with a $\pm 90^\circ$ phase shift.

Equivalently, the Hilbert transform can also be expressed as convolution with the convolution kernel

$$h(t) = \frac{1}{\pi t}, \tag{2.6.9}$$

as

$$g(t) = h(t) * f(t) = p.v. \int_{-\infty}^{\infty} f(\tau)\frac{1}{\pi(t-\tau)}\mathrm{d}\tau, \tag{2.6.10}$$

where "$p.v.$" means principle value.

We have already seen that the ramp filter can be decomposed into the Hilbert transform and the derivative. It does not matter whether you perform the derivative first and then perform the Hilbert transform or you perform the Hilbert transform first and then perform the derivative. In fact, we have

$$\boldsymbol{H}(f'(t)) = \frac{\mathrm{d}(\boldsymbol{H}(f(t)))}{\mathrm{d}t}. \tag{2.6.11}$$

One can also change the order of the Hilbert transform and the 180° back-projection. To understand this property better, we will use the central slice

theorem and think in the Fourier domain. Let us consider the backprojection problem

$$b(x,y) = \int_{-\pi/2}^{\pi/2} \mathbf{H}p(s,\theta)\Big|_{s=x\cos\theta+y\sin\theta} \mathrm{d}\theta, \tag{2.6.12}$$

where the Hilbert transform is with respect to the variable s. Let the Fourier transform of $p(s,\theta)$ be $P(\omega,\theta)$. The central slice theorem applied to this particular problem is shown in Figure 2.16, where $P(\omega,\theta)$ is multiplied by either i or $-\mathrm{i}$ before putting it in the 2D Fourier domain.

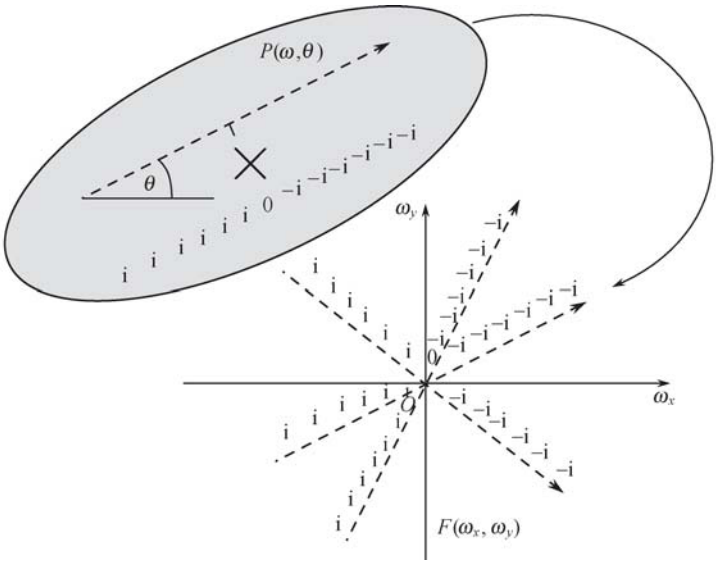


Fig. 2.16. Backprojection of $\mathbf{H}p(s,\theta)$ over $(-\pi/2,\pi/2)$ is equivalent to backprojection of $p(s,\theta)$ over $(-\pi/2,\pi/2)$ then multiply i in the left half (ω_x,ω_y) plane and multiply $-\mathrm{i}$ in the right half (ω_x,ω_y) plane.

An equivalent way to do this backprojection is to backproject $p(s,\theta)$, then multiply the entire left half (ω_x,ω_y) plane by i and multiply the entire right half (ω_x,ω_y) plane by $-\mathrm{i}$. This multiplication action can be achieved by performing line-by-line 1D Hilbert transform in the x-direction. Thus, we have

$$\mathbf{H}_x \int_{-\pi/2}^{\pi/2} p(s,\theta)\Big|_{s=x\cos\theta+y\sin\theta} d\theta = \int_{-\pi/2}^{\pi/2} \mathbf{H}_s p(s,\theta)\Big|_{s=x\cos\theta+y\sin\theta} \mathrm{d}\theta. \tag{2.6.13}$$

In the following, we will briefly introduce the concept of the finite Hilbert transform, which plays an important role in the region-of-interest (ROI) reconstruction with truncated projection data.

Without loss of generality, we assume that a real function $f(t)$ is supported in $(-1, 1)$, that is, $f(t) = 0$ if $|t| \geqslant 1$. Then, the Hilbert transform of $f(t)$ is given as

$$g(t) = p.v. \int_{-1}^{1} f(\tau) \frac{1}{\pi(t-\tau)} \mathrm{d}\tau. \tag{2.6.14}$$

Even though $f(t)$ is supported in a finite interval, $g(t)$ may not have a finite support. There are some formulas that are able to recover $f(t)$ using $g(t)$ only on $[-1, 1]$. For example,

$$f(t) = \frac{1}{\pi} \sqrt{\frac{t-1}{t+1}} p.v. \int_{-1}^{1} \sqrt{\frac{\tau+1}{\tau-1}} \frac{g(\tau)}{t-\tau} \mathrm{d}\tau, \tag{2.6.15}$$

or

$$f(t) = \frac{1}{\pi\sqrt{1-t^2}} \int_{-1}^{1} f(\tau)\mathrm{d}\tau + \frac{1}{\pi} p.v. \int \frac{\sqrt{1-\tau^2}}{\tau - t} g(\tau)\mathrm{d}\tau. \tag{2.6.16}$$

2.6.3 Proof of the Central Slice Theorem

The central slice theorem is given as

$$P\ (\omega, \theta) = F(\omega\cos\theta, \omega\sin\theta). \tag{2.6.17}$$

Proof

We start with the definition of the 1D Fourier transform:

$$P(\omega) = \int_{-\infty}^{\infty} p(s) \mathrm{e}^{-2\pi \mathrm{i} s\omega} \mathrm{d}s, \tag{2.6.18}$$

then use the definition of $p(s, \theta)$ (see Section 1.5), obtaining

$$P(\omega, \theta) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x\cos\theta + y\sin\theta - s) \mathrm{d}x\mathrm{d}y \right] \mathrm{e}^{-2\pi \mathrm{i} s\omega} \mathrm{d}s. \tag{2.6.19}$$

Changing the order of integrals yields

$$P(\omega, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \left[\int_{-\infty}^{\infty} \delta(x\cos\theta + y\sin\theta - s) e^{-2\pi \mathrm{i} s\omega} \mathrm{d}s \right] \mathrm{d}x\mathrm{d}y. \tag{2.6.20}$$

Using the property of the δ function, the inner integral over s can be readily obtained, and we have

$$P(\omega, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \mathrm{e}^{-2\pi \mathrm{i}\,(x\cos\theta + y\sin\theta)\omega} \mathrm{d}x\mathrm{d}y, \tag{2.6.21}$$

that is,

$$P(\omega,\theta)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-2\pi\mathrm{i}\,(xu+yv)}\Big|_{u=\omega\cos\theta,v=\omega\sin\theta}\mathrm{d}x\mathrm{d}y. \tag{2.6.22}$$

Finally, using the definition of the 2D Fourier transform yields

$$P(\omega,\theta)=F(\omega_x,\omega_y)\big|_{\omega_x=\omega\cos\theta,\quad\omega_y=\omega\sin\theta}. \tag{2.6.23}$$

In the polar coordinate system, the central slice theorem can be expressed as

$$P(\omega,\theta)=F_{polar}(\omega,\theta). \tag{2.6.24}$$

2.6.4 Derivation of the Filtered Backprojection Algorithm

We start with the 2D inverse Fourier transform in polar coordinates

$$f(x,y)=\int_0^{2\pi}\int_0^{\infty}F_{polar}(\omega,\theta)\mathrm{e}^{2\pi\mathrm{i}\,\omega(x\cos\theta+y\sin\theta)}\omega\mathrm{d}\omega\mathrm{d}\theta \tag{2.6.25}$$

Because $F_{polar}(\omega,\theta)=F_{polar}(-\omega,\theta+\pi)$, we have

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}F_{polar}(\omega,\theta)\,|\omega|e^{2\pi\mathrm{i}\,\omega(x\cos\theta+y\sin\theta)}\mathrm{d}\omega\mathrm{d}\theta. \tag{2.6.26}$$

By using the central slice theorem, we can replace F by P:

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}P(\omega,\theta)\,|\omega|\mathrm{e}^{2\pi\mathrm{i}\,\omega(x\cos\theta+y\sin\theta)}\mathrm{d}\omega\mathrm{d}\theta. \tag{2.6.27}$$

We recognize that $|\omega|$ is the ramp filter. Let $Q(\omega,\theta)=|\omega|\,P(\omega,\theta)$; then

$$f(x,y)=\int_0^{\pi}\int_{-\infty}^{\infty}Q(\omega,\theta)\mathrm{e}^{2\pi\mathrm{i}\,\omega(x\cos\theta+y\sin\theta)}\mathrm{d}\omega\mathrm{d}\theta. \tag{2.6.28}$$

Using the definition of the 1D inverse Fourier transform and denoting the inverse Fourier transform of Q as q, we have

$$f(x,y)=\int_0^{\pi}q(x\cos\theta+y\sin\theta,\theta)\mathrm{d}\theta, \tag{2.6.29}$$

or

$$f(x,y)=\int_0^{\pi}q(s,\theta)\Big|_{s=x\cos\theta+y\sin\theta}\mathrm{d}\theta. \tag{2.6.30}$$

This is the backprojection of $q(s,\theta)$ (see Section 1.5).

2.6.5 Expression of the Convolution Backprojection Algorithm

Let the convolution kernel $h(s)$ be the inverse Fourier transform of the ramp filter $|\omega|$; that is, $h(s) = \int_{-\infty}^{\infty} |\omega| \mathrm{e}^{2\pi \mathrm{i}\, \omega s} \mathrm{d}s$. It is readily obtained from Section 2.6.4 that

$$\begin{aligned} f(x,y) &= \int_0^{\pi} [h(s) * p(s,\theta)]\Big|_{s=x\cos\theta+y\sin\theta} \mathrm{d}\theta \\ &= \int_0^{\pi} \int_{-\infty}^{\infty} h(x\cos\theta + y\sin\theta - s) p(s,\theta) \mathrm{d}s \mathrm{d}\theta. \end{aligned} \tag{2.6.31}$$

2.6.6 Expression of the Radon Inversion Formula

The DHB (derivative, Hilbert transform, backprojection) algorithm is also referred to as the Radon inversion formula, which can be obtained by factoring the ramp filter $|\omega|$ into the derivative part and the Hilbert transform part:

$$|\omega| = (2\pi \mathrm{i}\, \omega) \times \left\{ \frac{1}{2\pi} [-\mathrm{i}\, \mathrm{sgn}(\omega)] \right\} \tag{2.6.32}$$

The inverse Fourier transform of $-\mathrm{i}\, \mathrm{sgn}(\omega)$ is $1/(\pi s)$ and the inverse Fourier transform of $2\pi \mathrm{i}\, \omega$ is the derivative operator. Thus,

$$q(s,\theta) = \frac{\partial p(s,\theta)}{\partial s} * \frac{1}{2\pi^2 s}, \tag{2.6.33}$$

and

$$f(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} \frac{\partial p(s,\theta)}{\partial s} \frac{1}{2\pi^2 (x\cos\theta + y\sin\theta - s)} \mathrm{d}s \mathrm{d}\theta. \tag{2.6.34}$$

2.6.7 Derivation of the Backprojection-then-Filtering Algorithm

Let us first look at the backprojection $b(x,y)$ of the original data $p(s,\theta)$ (without filtering). The definition of the backprojection is

$$b(x,y) = \int_0^{\pi} p(s,\theta)\Big|_{s=x\cos\theta+y\sin\theta} \mathrm{d}\theta. \tag{2.6.35}$$

Using the definition of the inverse Fourier transform, $p(s,\theta)$ can be represented with its Fourier transform:

$$b(x,y) = \int_0^{\pi} \left[\int_{-\infty}^{\infty} P(\omega,\theta) \mathrm{e}^{2\pi \mathrm{i}\, \omega(x\cos\theta + y\sin\theta)} \mathrm{d}\omega \right] \mathrm{d}\theta. \tag{2.6.36}$$

Using the central slice theorem, we can replace P by F:

$$b(x,y) = \int_0^{\pi} \left[\int_{-\infty}^{\infty} F_{polar}(\omega,\theta) \mathrm{e}^{2\pi \mathrm{i}\, \omega(x\cos\theta + y\sin\theta)} \mathrm{d}\omega \right] \mathrm{d}\theta, \tag{2.6.37}$$

or

$$b(x,y) = \int_0^{\pi} \int_{-\infty}^{\infty} \frac{F_{polar}(\omega,\theta)}{|\omega|} \mathrm{e}^{2\pi \mathrm{i}\, \omega(x\cos\theta + y\sin\theta)} |\omega| \, \mathrm{d}\omega \mathrm{d}\theta. \tag{2.6.38}$$

This is the 2D inverse Fourier transform in polar coordinates. If we take the 2D Fourier transform (in the polar coordinate system) on both sides of the above equation, we have

$$B_{polar}(\omega,\theta) = \frac{F_{polar}(\omega,\theta)}{|\omega|}, \tag{2.6.39}$$

or in the Cartesian system,

$$B(\omega_x,\omega_y) = \frac{F(\omega_x,\omega_y)}{\sqrt{\omega_x^2+\omega_y^2}}; \tag{2.6.40}$$

that is

$$F(\omega_x,\omega_y) = \sqrt{\omega_x^2+\omega_y^2} B(\omega_x,\omega_y). \tag{2.6.41}$$

The backprojection-then-filtering algorithm follows immediately.

2.7 Worked Examples

Example 1 Write a short Matlab program to illustrate the importance of using sufficient view angles in a tomography problem. The Matlab function "*phantom*" generates a mathematical Shepp-Logan phantom. The function "*radon*" generates the projection data, that is, the Radon transform of the phantom. The function "*iradon*" performs the filtered backprojection reconstruction.

Solution

The Matlab code:

```
P = phantom(128); %Generate the Shepp-Logan phantom in a 128x128
    array
angle = linspace(0,179,180); %Sampling angles
R = radon(P, angle); %Generate the Radon transform over 180°
    I1 = iradon(R, angle); %Inverse Radon transform, i.e., FBP recon-
        struction
    I2 = iradon(R, angle,'linear','none'); %Backprojection without ramp-
        filtering
    subplot(1,3,1), imshow(P), title('Original')
    subplot(1,3,2), imshow(I1), title('Filtered backprojection')
    subplot(1,3,3), imshow(I2,[]), title('Unfiltered backprojection')
```

We get the following results (see Figure 2.17).

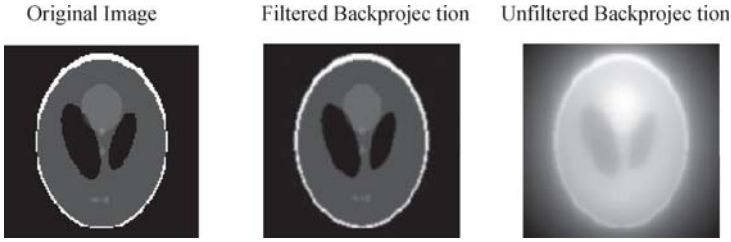


Fig. 2.17. The true image and the reconstructed images with the Matlab function "iradon."

If we change the line angle = linspace(0,179,180) to
angle = linspace(0,179,10)
angle = linspace(0,179,20)
angle = linspace(0,179,40)
angle = linspace(0,179,80)
respectively, we get the following results (see Figure 2.18).

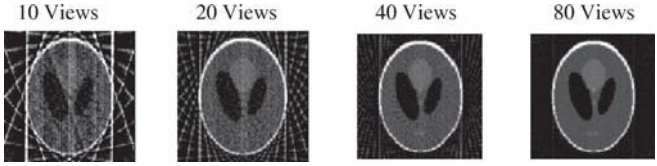


Fig. 2.18. Insufficient views cause artifacts.

Example 2 Run a simple Matlab code to show the noise effect. Apply three different window functions to the ramp filter to control noise.

Solution

The Matlab code:

```
P = phantom(128);
angle = linspace(0,179,180);
R = radon(P,angle);
R = 1e12*imnoise(1e-12*R,'Poisson'); %Add Poisson Noise
    I1 = iradon(R,angle,'Ram-Lak');
subplot(1,3,1), imshow(I1,[]), title('w/ Ram-Lak filter')
    I1 = iradon(R,angle,'Cosine');
subplot(1,3,2), imshow(I1,[]), title('w/ Cosine filter')
    I1 = iradon(R,angle,'Hann');
subplot(1,3,3), imshow(I1,[]), title('w/ Hann filter')
```

The FBP reconstruction results with different window functions are shown in Figure 2.19.

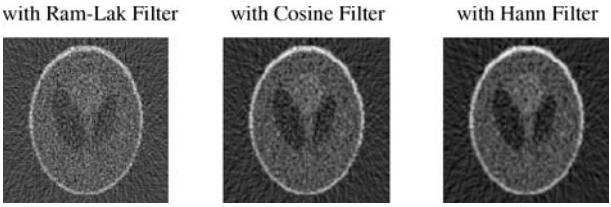


Fig. 2.19. Using different window functions to control noise.

Example 3* Find the discrete "Ramachandran-Lakshminarayanan" convolver, which is the inverse Fourier transform of the ramp filter. The cut-off frequency of the ramp filter is $\omega = \frac{1}{2}$.

Solution

Let us first use the inverse Fourier transform to find the continuous version of the convolver $h(s)$:

$$h(s) = \int_{-1/2}^{1/2} |\omega| \mathrm{e}^{2\pi \mathrm{i}\, \omega s} d\omega = \frac{1}{2} \frac{\sin(\pi s)}{\pi s} - \frac{1}{4} \left(\frac{\sin(\frac{\pi s}{2})}{\frac{\pi s}{2}} \right)^2 . \tag{2.7.1}$$

To convert it to the discrete form, let $s = n$ (integer) and we have

$$h(n) = \begin{cases} \frac{1}{4}, & n = 0 \\ 0, & n \quad \text{even} \\ \frac{-1}{n^2 \pi^2}, & n \quad \text{odd} \end{cases} \tag{2.7.2}$$

The continuous and discrete convolvers are displayed in Figure 2.20.

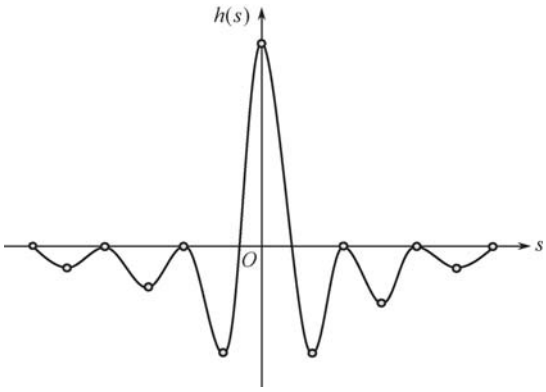


Fig. 2.20. The continuous and sampled "Ramachandran-Lakshminarayanan" convolution kernels.

2.8 Summary

- The Fourier transform is a useful tool to express a function in the frequency domain. The inverse Fourier transform brings the frequency representation back to the original spatial domain expression.
- In the Fourier domain, the projection data and the original image are clearly related. This relationship is stated in the popular central slice theorem. The 1D Fourier transform of the projection data at one view is one line in the 2D Fourier transform of the original image. Once we have a sufficient number of projection view angles, their corresponding lines will cover the 2D Fourier plane.
- The backprojection pattern in the 2D Fourier domain indicates that the sampling density is proportional to $1/|\omega|$. Therefore, the backprojection itself can only provide a blurred reconstruction. An exact reconstruction requires a combination of ramp-filtering and backprojection.
- The most popular image reconstruction algorithm is the FBP (Filtered Backprojection) algorithm. The ramp-filtering can be implemented as multiplication in the frequency domain or as convolution in the spatial domain.
- One has the freedom to switch the order of ramp-filtering and backprojection.
- Ramp-filtering can be further decomposed to the Hilbert transform and the derivative operation. Therefore, we have even more ways to reconstruct the image.

- Under certain conditions, the region-of-interest (ROI) can be exactly reconstructed with truncated projection data.
- The readers are expected to understand two main concepts in this chapter: the central slice theorem and the FBP algorithm.

Problems

Problem 2.1 Let a 2D object be defined as

$$f(x, y) = \cos(2\pi x) + \cos(2\pi y).$$

Find its 2D Fourier transform $F(\omega_x, \omega_y)$. Determine the Radon transform of this object $f(x, y)$ using the central slice theorem.

Problem 2.2 Let $f_1(x, y)$ and $f_2(x, y)$ be two 2D functions, and their Radon transforms are $p_1(s, \theta)$ and $p_2(s, \theta)$, respectively. If $f(x, y)$ is the 2D convolution of $f_1(x, y)$ and $f_2(x, y)$, use the central slice theorem to prove that the Radon transform of $f(x, y)$ is the convolution of $p_1(s, \theta)$ and $p_2(s, \theta)$ with respect to variable s.

Problem 2.3 Let the Radon transform of a 2D object $f(x, y)$ be $p(s, \theta)$, the 2D Fourier transform of $f(x, y)$ be $F(\omega_x, \omega_y)$, and the 1D Fourier transform of $p(s, \theta)$ with respect to the first variable s be $P(\omega, \theta)$. What is the physical meaning of the value $F(0, 0)$? What is the physical meaning of the value $P(0, \theta)$? Is it possible that $\int_{-\infty}^{\infty} g(s, 0^\circ)\mathrm{d}s \neq \int_{-\infty}^{\infty} g(s, 90^\circ)\mathrm{d}s$?

Problem 2.4 Prove that the Hilbert transform of the function

$$f(t) = \begin{cases} \sqrt{1 - t^2}, & |t| < 1 \\ 0, & |t| \geqslant 1 \end{cases}$$

is

$$g(t) = \begin{cases} t, & |t| < 1 \\ t - \sqrt{t^2 - 1}\mathrm{sgn}(t). & |t| \geqslant 1 \end{cases}$$

References

1. Bracewell RN (1956) Strip integration in radio astronomy. Aust J Phys 9: 198 – 217

2. Parallel-Beam Image ReconstructionBracewell RN (1986) The Fourier transform and its applications, 2nd edn. McGraw-Hill, New York
3. Carrier GF, Krook M, Pearson CE (1983) Functions of a complex variable theory and technique. Hod, Ithaca
4. Clackdoyle R, Noo F, Guo J et al (2004) A quantitative reconstruction from truncated projections of compact support. Inverse Probl 20: 1281 – 1291
5. Deans SR (1983) The radon transform and some of its applications. John Wiley & Sons, New York
6. Defrise M, Noo F, Clackdoyle et al (2006) Truncated Hilbert transform and image reconstruction from limited tomographic data. Inverse Probl 22: 1037 – 1053
7. Hahn SL (1996) Hilbert transforms in signal processing. Artech House, Boston
8. Huang Q, Zeng GL, Gullberg GT (2007) An analytical inversion of the 180° exponential radon transform with a numerically generated kernel. Int J Image and Graphics 7: 71 – 85
9. Kak AC, Slaney M (1998) Principles of computerized tomographic imaging, IEEE Press, New York
10. Kanwal RP (1971) Linear integral equations theory and technique. Academic, New York
11. Netterer F (1986) The mathematics of computerized tomography. Wiley, New York
12. Noo F, Clackdoyle R, Pack JD (2004) A two-step Hilbert transform method for 2D image reconstruction. Phys Med Biol 49: 3903 – 3923
13. Sidky EY, Pan X (2005) Recovering a compactly supported function from knowledge of its Hilbert transform on a finite interval. IEEE Trans Sig Processing Let 12: 97 – 100
14. Shepp L, Logan B (1974) The Fourier reconstruction of a head section. IEEE Trans Nucl Sci 21: 21 – 43
15. Tricomi FG (1957) Integral equations. Interscience, New York
16. Ye Y, Yu H, Wei Y et al (2007) A general local reconstruction approach based on a truncated Hilbert transform. Int J Biom Imag 63634
17. Ye Y, Yu H, Wang G (2007) Exact interior reconstruction with cone-beam CT. Int J Biom Imag 10693
18. You J, Zeng GL (2006) Exact finite inverse Hilbert transforms. Inverse Probl 22: L7 – L10
19. Yu H, Wang G (2009) Compressed sensing based interior tomography. Phys Med Biol 54: 2791 – 2805
20. Yu H, Yang J, Jiang M et al (2009) Interior SPECT: exact and stable ROI reconstruction from uniformly attenuated local projections. Commun Numer Meth Engng 25: 693 – 710
21. Yu H, Ye Y, Wang G (2008) Interior reconstruction using the truncated Hilbert transform via singular value decomposition. J X-Ray Sci Tech 16: 243 – 251
22. Zeng GL (2007) Image reconstruction via the finite Hilbert transform of the derivative of the backprojection. Med Phys 34: 2837 – 2843
23. Zeng GL, You J, Huang Q et al (2007) Two finite inverse Hilbert transform formulae for local tomography. Int J Imag Syst Tech 17: 219 – 223

3 Fan-Beam Image Reconstruction

The image reconstruction algorithms discussed in Chapter 2 are for parallel-beam imaging. If the data acquisition system produces projections that are not along parallel lines, the image reconstruction algorithms presented in Chapter 2 cannot be applied directly. This chapter uses the flat detector and curved detector fan-beam imaging geometries to illustrate how a parallel-beam reconstruction algorithm can be converted to user's imaging geometry for image reconstruction.

3.1 Fan-Beam Geometry and Point Spread Function

The fan-beam imaging geometry is common in X-ray CT, where the fan-beam focal point is the X-ray source. A fan-beam imaging geometry and a parallel-beam imaging geometry are compared in Figure 3.1.

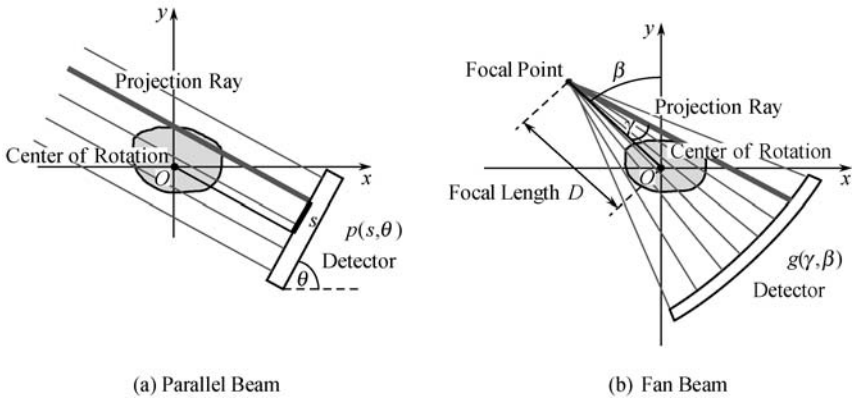


Fig. 3.1. Comparison of the parallel-beam and the fan-beam imaging geometries.

For the parallel-beam geometry, we have a central slice theorem to derive reconstruction algorithms. We do not have an equivalent theorem for the fan-beam geometry. We will use a different strategy — converting the fan-beam imaging situation into the parallel-beam imaging situation and modifying the parallel-beam algorithms for fan-beam use.

In Chapter 2, when we discussed parallel-beam imaging problems, it was not mentioned, but we always assumed that the detector rotates around at a constant speed and has a uniform angular interval when data are taken. We make the same assumption here for the fan-beam.

For the parallel-beam imaging geometry, this assumption results in a shift-invariant point spread function (PSF) for projection/backprojection. In other words, if you put a point source in the x-y plane (it does not matter where you put it), calculate the projections, and perform the backprojection, then you will always get the same star-like pattern (see Figure 3.2). This pattern is called the point spread function (PSF) of the projection/backprojection operation.

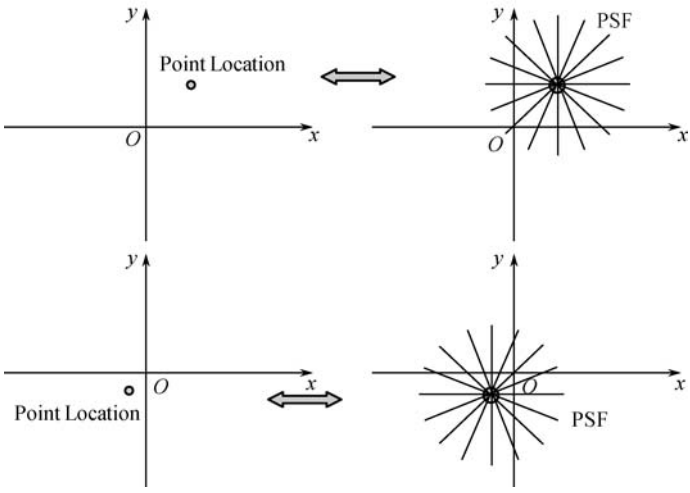


Fig. 3.2. The projection/backprojection PSF is shift invariant.

In the parallel-beam case, when you find the backprojection at the point (x, y), you draw a line through this point and perpendicular to each detector. This line meets the detector at a point, say, s^*. Then add the value $p(s^*, \theta)$ to the location (x, y).

In the fan-beam case, when you find the backprojection at the point (x, y), you draw a line through this point and each focal-point location. This line has an angle, say, γ^*, with respect to the central ray of the detector. Then add the value $g(\gamma^*, \beta)$ to the location (x, y).

It can be shown that if the fan-beam focal-point trajectory is a complete

circle, the PSF is shift-invariant (that is, the pattern does not change when the location of the point-source changes) and has the same PSF pattern as that for the parallel-beam case (see Figure 3.3).

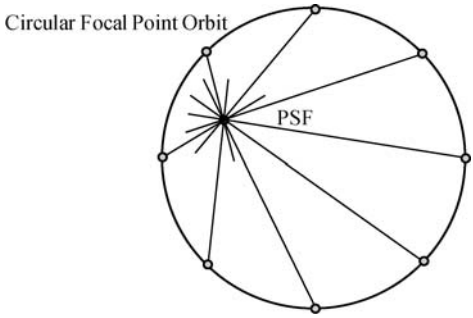


Fig. 3.3. The fan-beam 360° full-scan PSF is the same as that for the parallel-beam scan.

This observation is important. It implies that if you project and backproject an image, you will get the same blurred version of that image, regardless of the use of parallel-beam or fan-beam geometry.

If the original image is $f(x, y)$ and if the backprojection of the projection data is $b(x, y)$, then the PSF can be shown to be $1/r$, where $r = \sqrt{x^2 + y^2}$. Then $f(x, y)$ and $b(x, y)$ are related by

$$b(x, y) = f(x, y) * * \frac{1}{r} \tag{3.1.1}$$

where "$**$" denotes 2D convolution. The Fourier transform of b is B, and the Fourier transform of f is F. Thus the above relationship in the Fourier domain becomes

$$B(\omega_x, \omega_y) = F(\omega_x, \omega_y) \times \frac{1}{\sqrt{\omega_x^2 + \omega_y^2}}, \tag{3.1.2}$$

because the 2D Fourier transform of $1 \Big/ \sqrt{x^2 + y^2}$ is $1 \Big/ \sqrt{\omega_x^2 + \omega_y^2}$.

We already know that if a 2D ramp filter is applied to the backprojected image $b(x, y)$, the original image $f(x, y)$ can be obtained. The same technique can be used for the fan-beam backprojected image $b(x, y)$. The backprojection-then-filtering algorithm is the same for the parallel-beam and fan-beam imaging geometries. If we apply the 2D ramp filter $\sqrt{\omega_x^2 + \omega_y^2}$ to both sides of the above Fourier domain relationship, the Fourier transform $F(\omega_x, \omega_y)$ of the original image $f(x, y)$ is readily obtained:

$$F(\omega_x, \omega_y) = B(\omega_x, \omega_y) \times \sqrt{\omega_x^2 + \omega_y^2}. \tag{3.1.3}$$

Finally, the original image $f(x, y)$ is found by taking the 2D inverse Fourier transform.

3.2 Parallel-Beam to Fan-Beam Algorithm Conversion

If you want to reconstruct the image by a filter-then-backproject (i.e., filtered backprojection) algorithm, a different strategy must be used.

A straightforward approach would be to rebin every fan-beam ray into a parallel-beam ray. For each fan-beam ray-sum $g(\gamma, \beta)$, we can find a parallel-beam ray-sum $p(s, \theta)$ that has the same orientation as the fan-beam ray with the relations (see Figure 3.4)

$$\theta = \gamma + \beta, \tag{3.2.1}$$

and

$$s = D \sin \gamma, \tag{3.2.2}$$

where D is the focal length. We then assign

$$p(s, \theta) = g(\gamma, \beta). \tag{3.2.3}$$

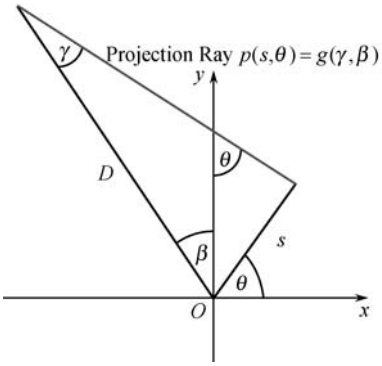


Fig. 3.4. A fan-beam ray can be represented using the parallel-beam geometry parameters.

After rebinning the fan-beam data into the parallel-beam format, we then use a parallel-beam image reconstruction algorithm to reconstruct the image. However, this rebinning approach is not preferred because rebinning requires data interpolation when changing coordinates. Data interpolation introduces errors. The idea of rebinning is feasible, but the results may not be accurate enough.

However, the idea above is not totally useless. Let us do it in a slightly different way. We start out with a parallel-beam image reconstruction algorithm, which is a mathematical expression. On the left-hand-side of the expression is the reconstructed image $f(x, y)$. One the right-hand-side is an integral expression that contains the projection $p(s, \theta)$ and some other factors associated with s and θ (see Figure 3.5).

Next, we replace the parallel-beam projection $p(s, \theta)$ by its equivalent fan-beam counterpart $g(\gamma, \beta)$ on the right-hand-side. Of course, this substitution is possible only if the conditions $\theta = \gamma + \beta$ and $s = D \sin \gamma$ are satisfied. These two relations are easier to see in Figure 3.4, which is derived from Figure 3.1.

In order to satisfy these two conditions, we stick them into the right-hand-side of the expression. This procedure is nothing but changing the variables in the integral, where the variables s and θ are changed into γ and β.

As a reminder, in calculus, when you change the variables in an integral, you need a Jacobian factor, which is a determinant calculated with some partial derivatives. This Jacobian is a function of γ and β.

After substituting the parallel-beam data $p(s, \theta)$ with fan-beam data $g(\gamma, \beta)$, changing variables s and θ to γ and β, and inserting a Jacobian $J(\gamma, \beta)$, a new fan-beam image reconstruction algorithm is born (see Figure 3.5)!

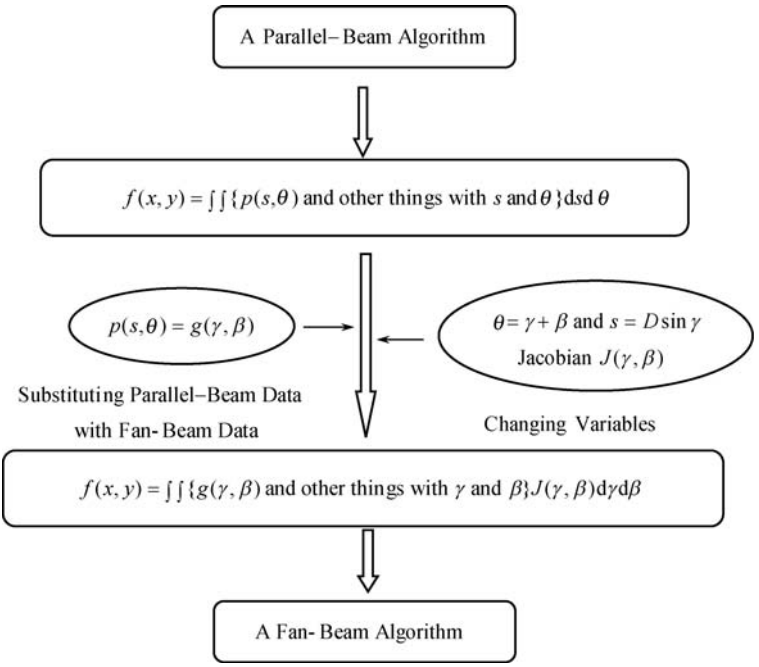


Fig. 3.5. The procedure to change a parallel-beam algorithm into a fan-beam algorithm.

The method outlined in Figure 3.5 is generic. We have a long list of parallel-beam image reconstruction algorithms. They can all be converted into fan-beam algorithms in this way, and accordingly, we also have a long list of fan-beam algorithms. We must point out that after changing the variables, a convolution operation (with respect to variable s) may not turn into a convolution (with respect to variable γ) automatically, and some mathematical manipulation is needed to turn it into a convolution form. Like parallel-beam algorithms, the fan-beam algorithms include combinations of ramp filtering and backprojection or combinations of the derivative, Hilbert transform, and backprojection.

Researchers treat the flat detector fan-beam and curved detector fan-beam differently in reconstruction algorithm development. In a flat detector, the data points are sampled with equal distance Δs intervals, while in a curved detector, the data points are sampled with equal angle $\Delta\gamma$ intervals (see Figure 3.6). A fan-beam algorithm can be converted from one geometry to the other with proper weighting adjustments.

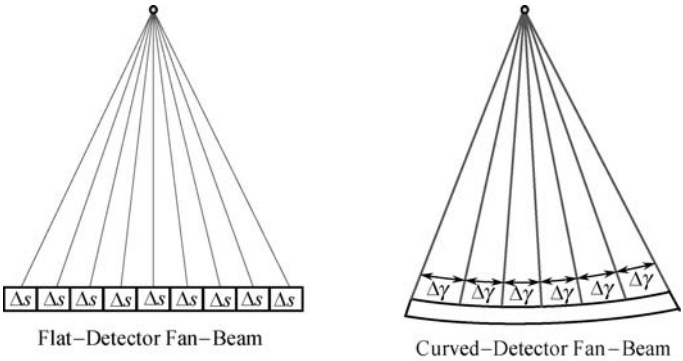


Fig. 3.6. Flat and curved detector fan-beam geometries.

3.3 Short Scan

In parallel-beam imaging, when the detector rotates 2π (i.e., $360°$), each projection ray is measured twice, and the redundant data are related by

$$p(s, \theta) = p(-s, \theta + \pi); \tag{3.3.1}$$

the redundant data are acquired by the two face-to-face detectors (see Figure 3.7). Therefore, it is sufficient to acquire data over an angular range of π.

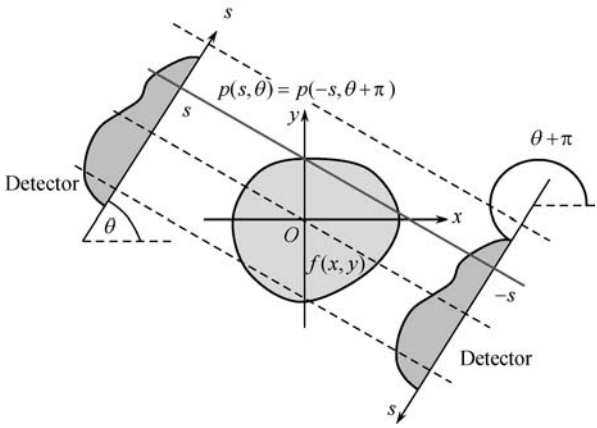


Fig. 3.7. Face-to-face parallel-beam detectors measure the same line integrals.

Likewise, when the fan-beam detector rotates 2π, each projection ray is also measured twice, and the redundant data are related by (see Figure 3.8)

$$g(\gamma, \beta) = g(-\gamma, \beta + 2\gamma + \pi). \tag{3.3.2}$$

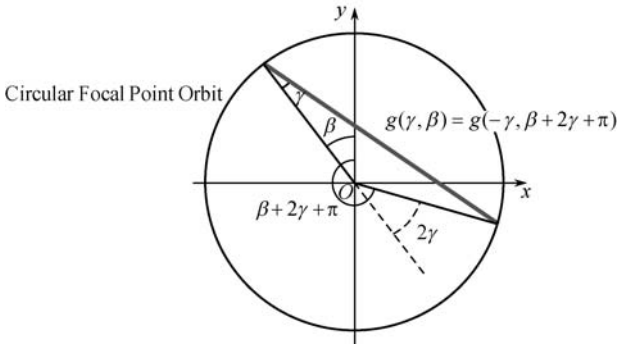


Fig. 3.8. Every ray is measured twice in a fan-beam 360° full scan.

Due to data redundancy, we can use a smaller angular (β) range than 2π for fan-beam data acquisition, hence the term *short scan*. The minimal range of β is determined by how the data are acquired. This required range can be less than π (see Figure 3.9 Left), equal to π (see Figure 3.9 Middle), or larger than π (see Figure 3.9 Right). The criterion is that we need at least 180° angular coverage for each point in the object in which we are interested.

We need to be cautious that in a fan-beam short scan, some rays are measured once, and other rays are measured twice. Even for the case that the angular range of β is less than π, there are still rays that are measured twice.

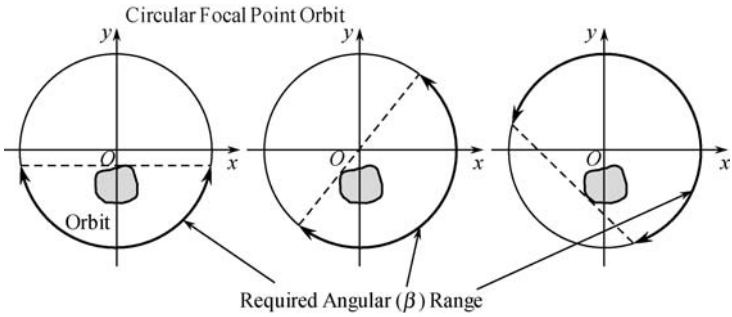


Fig. 3.9. Fan-beam minimum scan angle depends on the location of the object.

In fact, any ray that intersects the measured focal point trajectory twice is measured twice (see Figure 3.10). We require that any ray which passes through the object should be measured at least once. Proper weighting should be used in image reconstruction if data are redundant. For example, if a ray is measured twice, the sum of the weighting factors for these two measurements should be unity. For a fan-beam short-scan, the projection/backprojection PSF is no longer shift-invariant.

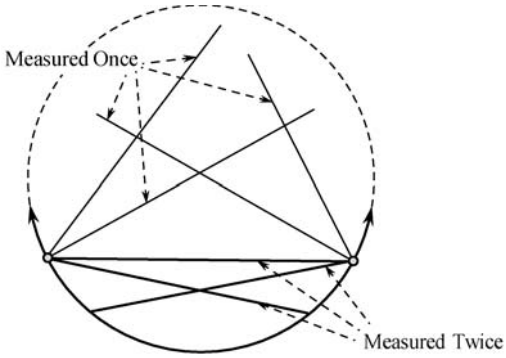


Fig. 3.10. For a fan-beam short scan, some rays are measured once and some rays are measured twice.

*3.4 Mathematical Expressions

This section presents the derivation steps for the filtered backprojection algorithm for the curved fan-beam detector. The ramp-filter used in the filtering step is formulated as a convolution. In this algorithm, the fan-beam

backprojector contains a distance dependent weighting factor, which causes non-uniform resolution throughout the reconstructed image when a window function is applied to the ramp-filter in practice. In order to overcome this problem, the ramp-filter can be replaced by a derivative operation and the Hilbert transform. The derivation of the fan-beam algorithm using the derivative and the Hilbert transform is given in this section.

3.4.1 Derivation of a Filtered Backprojection Fan-Beam Algorithm

We begin with the parallel-beam filtered backprojection algorithm (see Section 2.6.5), using polar coordinates (r, φ) instead of Cartesian coordinates (x, y). Then $x = r\cos\varphi$, $y = r\sin\varphi$, and $x\cos\theta + y\sin\theta = r\cos(\theta - \varphi)$. We have

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi}\int_{-\infty}^{\infty} p(s,\theta)h(r\cos(\theta-\varphi)-s)\mathrm{d}s\mathrm{d}\theta. \tag{3.4.1}$$

Changing variables $\theta = \gamma + \beta$ and $s = D\sin\gamma$ with the Jacobian $D\cos\gamma$ yields

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} g(\gamma,\beta)h(r\cos(\beta+\gamma-\varphi)-D\sin\gamma)D\cos\gamma\mathrm{d}\gamma\mathrm{d}\beta. \tag{3.4.2}$$

This is a fan-beam reconstruction algorithm, but the inner integral over γ is not yet in the convolution form. Convolution is much more efficient than a general integral in implementation. In the following, we are going to convert the integral over γ to a convolution with respect to γ.

For a given reconstruction point (r, φ), we define D' and γ' as in Figure 3.11, then $r\cos(\beta+\gamma-\varphi) - D\sin\gamma = D'\sin(\gamma'-\gamma)$, and

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2} g(\gamma,\beta)h(D'\sin(\gamma'-\gamma))D\cos\gamma\mathrm{d}\gamma\mathrm{d}\beta. \tag{3.4.3}$$

Now, we prove a special property of the ramp filter

$$h(D'\sin\gamma) = \left(\frac{\gamma}{D'\sin\gamma}\right)^2 h(\gamma), \tag{3.4.4}$$

which will be used in the very last step the derivation of the fan-beam formula.

Using the definition of the ramp filter kernel $h(t) = \displaystyle\int_{-\infty}^{\infty} |\omega| \mathrm{e}^{\mathrm{i}\,2\pi\omega t}\mathrm{d}\omega$, we

have

$$\begin{aligned} h(D'\sin\gamma) &= \int_{-\infty}^{\infty} |\omega| \mathrm{e}^{\mathrm{i}\,2\pi\omega D'\sin\gamma} \mathrm{d}\omega \\ &= \left(\frac{\gamma}{D'\sin\gamma}\right)^2 \int_{-\infty}^{\infty} \left|\omega D' \frac{\sin\gamma}{\gamma}\right| \mathrm{e}^{\mathrm{i}\,2\pi\omega \frac{D'\sin\gamma}{\gamma}\gamma} \mathrm{d}\left(\omega D' \frac{\sin\gamma}{\gamma}\right) \\ &= \left(\frac{\gamma}{D'\sin\gamma}\right)^2 \int_{-\infty}^{\infty} |\hat{\omega}| \mathrm{e}^{\mathrm{i}\,2\pi\hat{\omega}\gamma} \mathrm{d}\hat{\omega} \\ &= \left(\frac{\gamma}{D'\sin\gamma}\right)^2 h(\gamma). \end{aligned} \tag{3.4.5}$$

If we denote $h_{fan}(\gamma) = \dfrac{D}{2}\left(\dfrac{\gamma}{\sin\gamma}\right)^2 h(\gamma)$, then the fan-beam convolution backprojection algorithm is obtained as

$$f(r,\varphi) = \int_0^{2\pi} \frac{1}{(D')^2} \int_{-\pi/2}^{\pi/2} (\cos\gamma) g(\gamma,\beta) h_{fan}(\gamma'-\gamma) \mathrm{d}\gamma \mathrm{d}\beta. \tag{3.4.6}$$

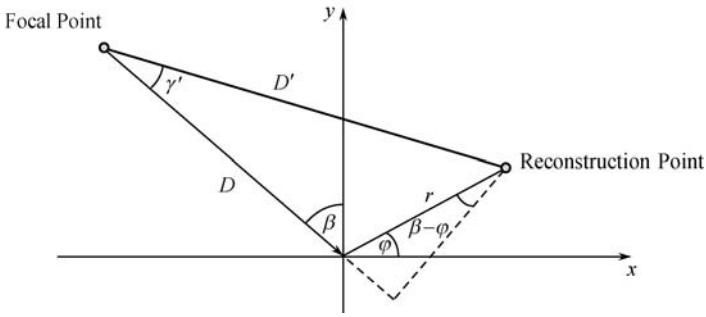


Fig. 3.11. The reconstruction point (r,φ) defines the angle γ' and distance D'.

3.4.2 A Fan-Beam Algorithm Using the Derivative and the Hilbert Transform

The general idea of decomposition of the ramp filter into the derivative and the Hilbert transform can be applied to fan-beam image reconstruction. A derivative-Hilbert transform-backprojection algorithm can be obtained by doing a coordinate transformation on the Radon inversion formula (see Section 2.6.6) as follows. Noo, Defrise, Kudo, Clackdoyle, Pan, Chen, Wang, You and many others have contributed significantly in developing algorithms

using the derivative and the Hilbert transform. We first re-write the Radon inversion formula (see Section 2.6.6) in the polar coordinate system, that is, $f(r,\varphi)$ can be reconstructed as

$$f(r,\varphi)=\frac{1}{2}\int_0^{2\pi}\int_{-\infty}^{\infty}\frac{\partial p(s,\theta)}{\partial s}\frac{1}{2\pi^2(r\cos(\theta-\varphi)-s)}\mathrm{d}s\mathrm{d}\theta. \tag{3.4.7}$$

Changing variables from (s,θ) to (γ,β) and using $\dfrac{\partial p}{\partial s}=\dfrac{1}{D\cos\gamma}\dfrac{\partial g}{\partial\gamma}$, we have

$$\begin{aligned} f(r,\varphi)&=\frac{1}{2}\int_0^{2\pi}\int_{-\pi/2}^{\pi/2}\frac{1}{D\cos\gamma}\frac{\partial g(\gamma,\beta)}{\partial\gamma}\frac{1}{2\pi^2D'\sin(\gamma'-\gamma)}D\cos\gamma\mathrm{d}\gamma\mathrm{d}\beta\\ &=\int_0^{2\pi}\frac{1}{4\pi^2D'}\int_{-\pi/2}^{\pi/2}\frac{\partial g(\gamma,\beta)}{\partial\gamma}\frac{1}{\sin(\gamma'-\gamma)}\mathrm{d}\gamma\mathrm{d}\beta, \end{aligned} \tag{3.4.8}$$

where D' is the distance from the reconstruction point to the focal point at angle β. This D' factor is not desirable in a reconstruction algorithm. A small D' can make the algorithm unstable; this spatially variant factor also costs some computation time. In a 2π scan, each ray is measured twice. If proper weighting is chosen for the redundant measurements, this D' factor can be eliminated.

Let us introduce a weighting function w in the above DHB (derivative, Hilbert transform, backprojection) algorithm:

$$f(r,\varphi)=\frac{1}{4\pi^2}\int_0^{2\pi}\frac{w(\gamma',\beta,r,\varphi)}{D'}\int_{-\pi/2}^{\pi/2}\frac{\partial g(\gamma,\beta)}{\partial\gamma}\frac{1}{\sin(\gamma'-\gamma)}\mathrm{d}\gamma\mathrm{d}\beta. \tag{3.4.9}$$

If we use $\hat{g}$ to denote the result of the derivative and Hilbert transform of the fan-beam data:

$$\hat{g}(\gamma',\beta)=\int_{-\pi/2}^{\pi/2}\frac{\partial g(\gamma,\beta)}{\partial\gamma}\frac{1}{\sin(\gamma'-\gamma)}\mathrm{d}\gamma, \tag{3.4.10}$$

then

$$f(r,\varphi)=\frac{1}{4\pi^2}\int_0^{2\pi}\frac{w(\gamma',\beta,r,\varphi)}{D'}\hat{g}(\gamma',\beta)\mathrm{d}\beta. \tag{3.4.11}$$

It can be shown that $\hat{g}/D'$ has the same redundancy property as the original fan-beam data g (see Figure 3.8). Therefore, it is required that the weighting function satisfy the condition

$$w(\gamma',\beta,r,\varphi)+w(\gamma'_c,\beta_c,r,\varphi)=2, \tag{3.4.12}$$

with $\gamma'_c=-\gamma'$ and $\beta_c=\beta+2\gamma'+\pi$. If we define $w(\gamma',\beta,r,\varphi)=\dfrac{D'}{D\cos\gamma'}$, then the above condition is satisfied because $D'+D'_c=2D\cos\gamma'$ (see Figure

3.12). Finally,

$$f(r,\varphi) = \frac{1}{2}\left\{\frac{1}{4\pi^2}\int_0^{2\pi}\left[\frac{w(\gamma',\beta,r,\varphi)}{D'} + \frac{w(\gamma'_c,\beta_c,r,\varphi)}{D'_c}\right]\hat{g}(\gamma',\beta)\mathrm{d}\beta\right\}$$
$$= \frac{1}{4\pi^2 D}\int_0^{2\pi}\frac{1}{\cos\gamma'}\hat{g}(\gamma',\beta)\mathrm{d}\beta. \tag{3.4.13}$$

This result can be derived without introducing a weighting function w (see Ref. [14]).

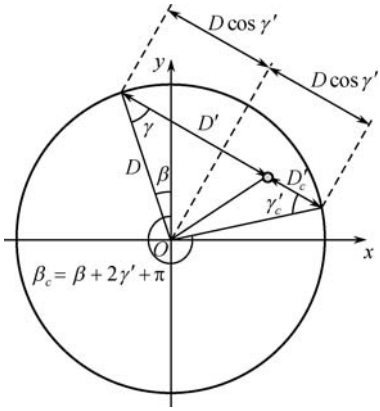


Fig. 3.12. Proper weighting can make the distance dependent factor disappear from the backprojector in a 360° full scan.

3.5 Worked Examples

Example 1 Does the following fan-beam geometry acquire sufficient projection data for image reconstruction? The fan-beam focal-point trajectory consists of three disjoint arcs as shown in Figure 3.13.

Solution

Yes. If you draw any line through the circular object, this line will intersect the focal-point trajectory at least once.

Example 2 On the γ-β plane (similar to the sinogram for the parallel-beam projections), identify the double measurements for a fan-beam 2π scan.

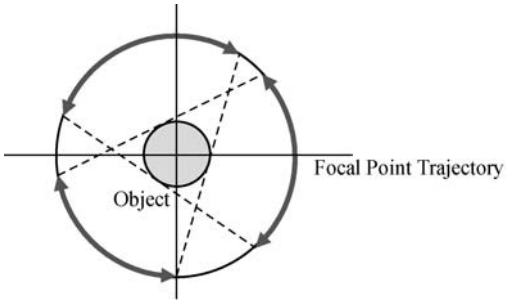


Fig. 3.13. A three-piece fan-beam focal-point trajectory.

Solution

From Figure 3.14, we can readily find the fan-beam data redundancy conditions for

$$g(\gamma_1, \beta_1) = g(\gamma_2, \beta_2) \tag{3.5.1}$$

as

$$\gamma_2 = -\gamma_1 \tag{3.5.2}$$

and

$$\beta_2 = \beta_1 + 2\gamma_1 + \pi. \tag{3.5.3}$$

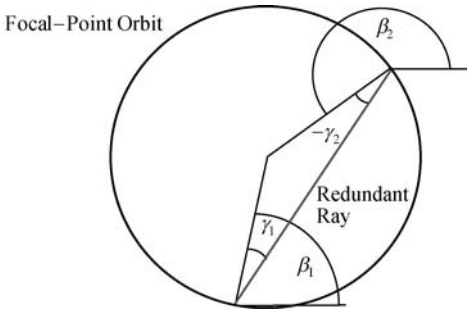


Fig. 3.14. A redundant measurement in a fan-beam scan.

Using these conditions, the fan-bean data redundancy is depicted on the γ-β plane in Figure 3.15, where every vertical line corresponds to a redundant slant line.

Example 3* Derive a filtered backprojection algorithm for the flat-detector fan-beam geometry.

Solution

We begin with the parallel-beam filtered backprojection algorithm, using polar coordinates (r, φ) instead of Cartesian coordinates (x, y). Then $x =$

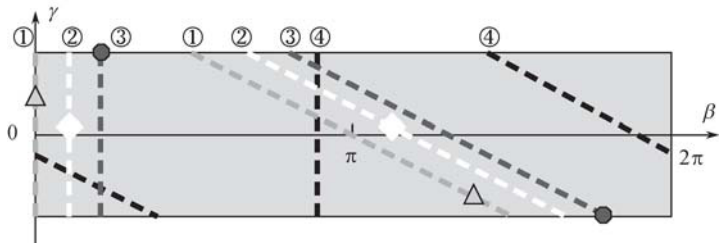


Fig. 3.15. In the γ-β plane representation of the fan-bean data, each vertical line of measurements is the same as the slant line of measurements with the same index number ①, ②, ③, or ④.

$r\cos\varphi$, $y = r\sin\varphi$, and $x\cos\theta + y\sin\theta = r\cos(\theta - \varphi)$. We have

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi}\int_{-\infty}^{\infty} p(s,\theta)h(r\cos(\theta-\varphi)-s)\mathrm{d}s\mathrm{d}\theta. \tag{3.5.4}$$

For the flat-detector fan-beam projection $g(t,\beta) = p(s,\theta)$ if the fan-beam and parallel-beam variables are related by $\theta = \beta + \tan^{-1}\dfrac{t}{D}$ and $s = D\dfrac{t}{\sqrt{D^2+t^2}}$. Changing the parallel-beam variables to the fan-beam variables with the Jacobian $\dfrac{D^3}{(D^2+t^2)^{\frac{3}{2}}}$ yields

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi}\int_{-\infty}^{\infty} g(t,\beta)h\left((\hat{t}-t)\frac{UD}{\sqrt{D^2+t^2}}\right)\frac{D^3}{(D^2+t^2)^{\frac{3}{2}}}\mathrm{d}t\mathrm{d}\beta, \tag{3.5.5}$$

where we have used $r\cos(\theta-\varphi) - s = (\hat{t}-t)\dfrac{UD}{\sqrt{D^2+t^2}}$ with

$$U = \frac{D + r\sin(\beta-\varphi)}{D} \tag{3.5.6}$$

and

$$\hat{t} = \frac{Dr\cos(\beta-\varphi)}{D + r\sin(\beta-\varphi)} \tag{3.5.7}$$

(see Figure 3.16).

This is a fan-beam reconstruction algorithm, but the inner integral over t is not yet in the convolution form. Using a special property of the ramp filter $h(at) = \dfrac{1}{a^2}h(t)$ yields

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi}\frac{1}{U^2}\int_{-\infty}^{\infty}\frac{D}{\sqrt{D^2+t^2}}g(t,\beta)h(\hat{t}-t)\mathrm{d}t\mathrm{d}\beta. \tag{3.5.8}$$

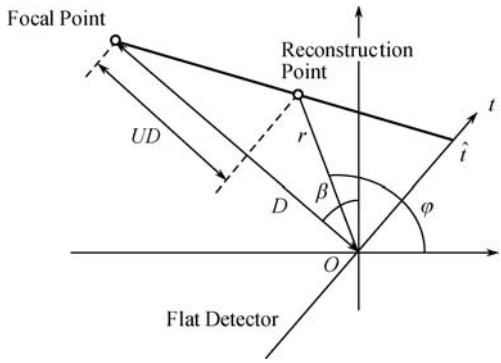


Fig. 3.16. Notation for flat detector fan-beam imaging geometry.

This is a fan-beam convolution backprojection algorithm, where $\frac{D}{\sqrt{D^2+t^2}}$ is the cosine pre-weighting factor, the integral over t is the ramp-filter convolution, and $1/U^2$ is the distance-dependent weighting factor in the backprojection.

Note: The relationship $h(at) = \frac{1}{a^2}h(t)$ does not hold for a *windowed* ramp filter. Therefore, the fan-beam algorithm derived above does not have uniform resolution in the reconstructed image in practice when a window function is applied to the ramp filter.

3.6 Summary

- The fan-beam geometry is popular in X-ray CT imaging.
- The fan-beam image reconstruction algorithms can be derived from their parallel-beam counterparts via changing of variables.
- There are two types of fan-beam detectors, that is, the flat detectors and curved detectors. Each detector type has its own image reconstruction algorithm.
- If the fan-beam focal-point trajectory is a full circle, it is called full-scan. If the trajectory is a partial circle, it is called short-scan. Even for a short-scan, some of the fan-beam rays are measured twice. The redundant measurements need proper weighting during image reconstruction.
- For some fan-beam image reconstruction algorithms, the backprojector contains a distance dependent weighting factor. When a window function is applied to the ramp-filter, this factor is not properly treated by the

window function and the resultant fan-beam FBP is no longer exact in the sense that the reconstructed image has non-uniform resolution. For the same reason, the Parker short-scan method (not discussed in this book) is inexact.

- The modern derivative and Hilbert transform based algorithms are able to weigh the short-scan redundant data in a correct way.
- In this chapter, the readers are expected to understand how a fan-beam algorithm can be obtained from a parallel-beam algorithm.

Problems

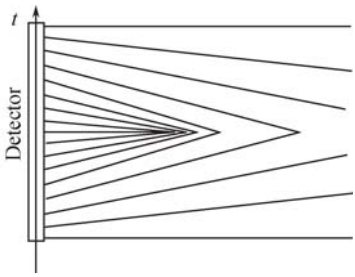
Problem 3.1 The data redundancy condition for the curved detector fan-beam imaging geometry is

$$g(\gamma, \beta) = g(-\gamma, \beta + 2\gamma + \pi).$$

What is the data redundancy condition for the flat detector fan-beam geometry?

Problem 3.2 In Chapter 3, we assume that the X-ray source (i.e., the fan-beam focal point) rotates around the object in a circular orbit. If the focal point orbit is not circular, then the focal length D is a function of the rotation angle β. Extend a fan-beam image reconstruction algorithm developed in this chapter to the situation that the focal point orbit is non-circular.

Problem 3.3 The method of developing an image reconstruction algorithm discussed in this chapter is not restricted to the fan-beam imaging geometry. For example, we can consider a variable focal length fan-beam imaging geometry, where the focal length D can be a function of t, which is the coordinate of the detection bin as shown in the figure below. Extend a fan-beam image reconstruction algorithm in this chapter to this variable focal length fan-beam imaging geometry.



References

1. Besson G (1996) CT fan-beam parameterizations leading to shift-invariant filtering. Inverse Probl 12: 815 – 833
2. Chen GH (2003) A new framework of image reconstruction from fan beam projections. Med Phys 30: 1151 – 1161
3. Chen GH, Tokalkanajalli R, Hsieh J (2006) Development and evaluation of an exact fan-beam reconstruction algorithm using an equal weighting scheme via locally compensated filtered backprojection (LCFBP). Med Phys 33: 475 – 481
4. Dennerlein F, Noo F, Hornegger J et al (2007) Fan-beam filtered-backprojection reconstruction without backprojection weight. Phys Med Biol 52: 3227 – 3240
5. Gullberg GT (1979) The reconstruction of fan-beam data by filtering the backprojection. Comput Graph Image Process 10: 30 – 47
6. Hanajer C, Smith KT, Solmon DC et al (1980) The divergent beam X-ray transform. Rocky Mountain J Math 10: 253 – 283
7. Natterer N (1993) Sampling in fan-beam tomography. SIAM J Appl Math 53: 358 – 380
8. Horn BKP (1979) Fan-beam reconstruction methods. Proc IEEE 67: 1616 – 1623
9. Noo F, Defrise M, Clackdoyle R et al (2002) Image reconstruction from fan-beam projections on less than a short scan. Phys Med Biol 47: 2525 – 2546
10. Pan X (1999) Optimal noise control in and fast reconstruction of fan-beam computed tomography image. Med Phys 26: 689 – 697
11. Pan X, Yu L (2003) Image reconstruction with shift-variant filtration and its implication for noise and resolution properties in fan-beam tomography. Med Phys 30: 590 – 600
12. Parker DL (1982) Optimal short scan convolution reconstruction for fan beam CT. Med Phys 9: 254 – 257
13. Silver MD (2000) A method for including redundant data in computed tomography. Med Phys 27: 773 – 774
14. You J, Liang Z, Zeng GL (1999) A unified reconstruction framework for both parallel-beam and variable focal-length fan-beam collimators by a Cormack-type inversion of exponential Radon transform. IEEE Trans Med Imaging 18: 59 – 65
15. You J, Zeng GL (2007) Hilbert transform based FBP algorithm for fan-beam CT full and partial scans. IEEE Trans Med Imaging 26: 190 – 199
16. Yu L, Pan X (2003) Half-scan fan-beam computed tomography with improved noise and resolution properties. Med Phys 30: 2629 – 2637
17. Wang J, Lu H, Li T et al (2005) An alternative solution to the nonuniform noise propagation problem in fan-beam FBP image reconstruction. Med Phys 32: 3389 – 3394
18. Wei Y, Hsieh J, Wang G (2005) General formula for fan-beam computed tomography. Phys Rev Lett 95: 258102
19. Wei Y, Wang G, Hsieh (2005) Relation between the filtered backprojection algorithm and the backprojection algorithm in CT. IEEE Sig Proc Lett 12: 633 – 636

20. Zeng GL (2004) Nonuniform noise propagation by using the ramp filter in fan-beam computed tomography. IEEE Trans Med Imaging 23: 690 – 695
21. Zeng GL, Gullberg (1991) Short-scan fan beam algorithm for non-circular detector orbits. SPIE Med Imaging V Conf. San Jose, California 332 – 340
22. Zou Y, Pan X, Sidky EY (2005) Image reconstruction in regions-of-interest from truncated projections in a reduced fan-beam scan. Phys Med Biol 50: 13 – 28

4 Transmission and Emission Tomography

This book considers real imaging systems in this chapter. If the radiation source is outside the patient, the imaging system acquires transmission data. If the radiation sources are inside the patient, the imaging system acquires the emission data. For transmission scans, the image to be obtained is a map (or distribution) of the attenuation coefficients inside the patient. For the emission scans, the image to be obtained is the distribution of the injected isotopes within the patient. Even for emission scans, an additional transmission scan is sometimes required in order to compensate for the attenuation effect of the emission photons. Some attenuation compensation methods for emission imaging are discussed in this chapter.

4.1 X-Ray Computed Tomography

In this chapter, we relate transmission and emission tomography measurements to line-integral data so that the reconstruction algorithms mentioned in the previous chapters can be used to reconstruct practical data in medical imaging.

X-ray CT (computed tomography) uses transmission measurements to estimate a cross-sectional image within the patient body. X-rays have very high energy, and they are able to penetrate the patient body. However, not every X-ray can make it through the patient's body. Some X-rays get scattered within the body, and their energy gets weakened. During X-ray scattering, an X-ray photon interacts with an electron within the patient, transfers part of its energy to that electron, and dislodges the electron (see Figure 4.1). The X-ray is then bounced to a new direction with decreased energy.

Some other X-rays completely disappear within the body, converting their energy to the tissues in the body, for example, via the photoelectric conversion. The photoelectric effect is a process in which the X-ray photon energy is completely absorbed by an atom within the patient. The absorbed energy

ejects an electron from the atom (see Figure 4.2).

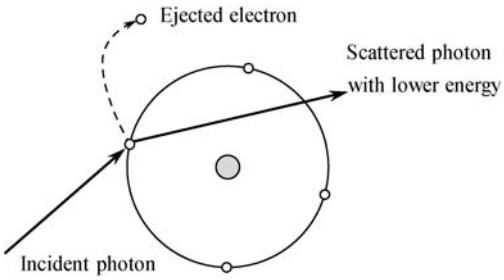


Fig. 4.1. Schematic representation of Compton scattering. The incident photon transfers part of its energy to an electron and is scattered in a new direction.

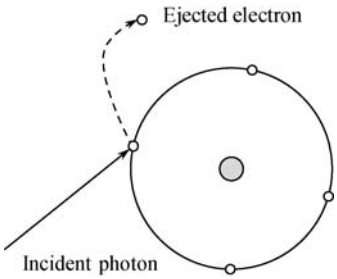


Fig. 4.2. Schematic representation of the photoelectric effect. The incident photon transfers all its energy to an electron and disappears.

Energy deposition within the body can damage DNA if the X-ray dose is too large. Let the X-ray intensity before entering the patient be I_0, and the intensity departing the patient be I_d; I_0 and I_d follow the Beer's law (see Figure 4.3):

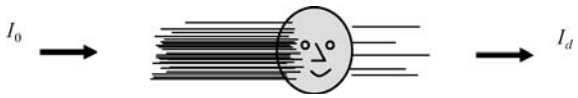


Fig. 4.3. The X-ray intensity is reduced after going through the object.

$$\frac{I_d}{I_0} = \exp(-p), \tag{4.1.1}$$

where p is the line integral of the linear attenuation coefficients along the path of the X-rays. A line integral of the attenuation coefficients is obtained by

$$p = \ln\left(\frac{I_0}{I_d}\right), \tag{4.1.2}$$

which is supplied to the image reconstruction algorithm for image reconstruction.

The goal of X-ray CT is to obtain a cross-section image of various attenuation coefficients within the body. A typical X-ray CT image is shown in Figure 4.4. The attenuation coefficient (commonly denoted by notation μ) is a property of a material; it is the logarithm of the input/output intensity ratio per unit length. Bones have higher μ values, and soft tissue has lower μ values. The attenuation coefficient of a material varies with the incoming X-ray energy; it becomes smaller when the X-ray energy gets higher.

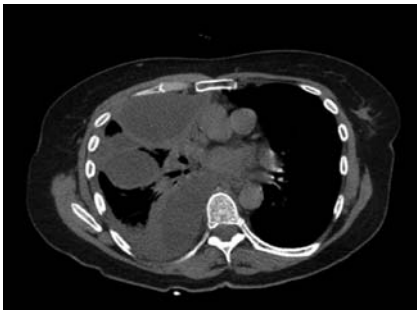


Fig. 4.4. An X-ray CT image.

The first-generation CT, which is no longer in use, had one small detector (see Figure 4.5). The X-ray source and the detector have two motions: linear translation and rotation. The X-ray source sends out a narrow pencil beam to obtain parallel-beam projections. The scanning time was rather long (about 25 minutes).

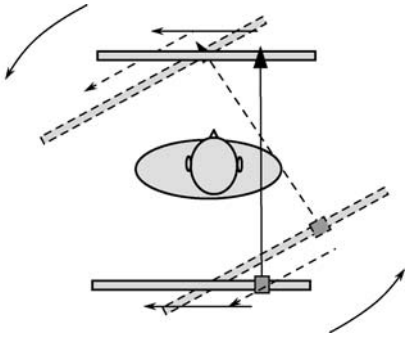


Fig. 4.5. In the first-generation CT, the X-ray tube and the detector translate and rotate.

The second-generation CT used narrow fan-beam geometry, consisting of 12 detectors (see Figure 4.6). Like the first-generation CT, it has two motions:

linear translation and rotation. Due to the fan-beam geometry, the scan time was shortened to about 1 minute.

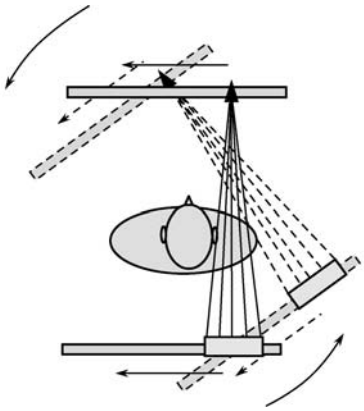


Fig. 4.6. The second-generation CT uses narrow fan-beam X-rays. The X-ray tube and the detector translate across the FOV and rotate around the object.

The third-generation CT uses wide fan-beam geometry, consisting of approximately 1 000 detectors (see Figure 4.7). No linear translation motion is necessary, and the scanning time was further reduced to about 0.5 seconds. The third-generation CT is currently very popular in medical imaging.

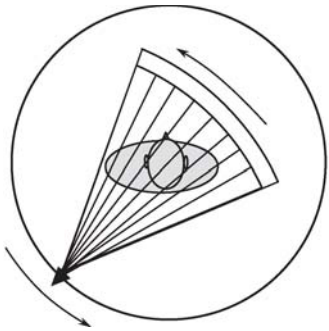


Fig. 4.7. The third-generation CT uses wide fan-beam X-rays. The X-ray tube and the detector rotate around the object; they do not translate anymore.

The fourth-generation CT has a stationary ring detector. The X-ray source rotates around the subject (see Figure 4.8). This scanning method forms a very fast fan-beam imaging geometry; however, it is impossible to collimate the X-rays on the detector, which causes this geometry to suffer from high rates of scattering.

Modern CT can perform helical scans, which is implemented as translating

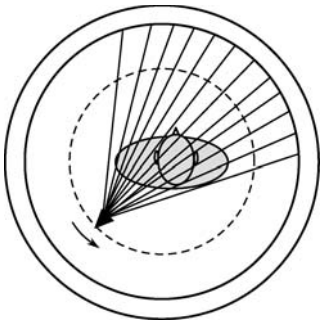


Fig. 4.8. In the fourth-generation CT, the ring detector does not rotate. The X-ray source rotates around the object.

the patient bed in the axial direction as the X-ray source and the detectors rotate. The modern CT has a 2D multi-row detector, and it acquires cone-beam data (see Figure 4.9). Image reconstruction methods for the cone-beam geometry will be covered in the next chapter.

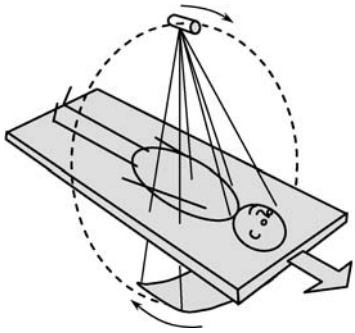


Fig. 4.9. The modern CT can perform cone-beam helix scans with a 2D detector. Some systems have multiple X-ray sources and 2D detectors. The helix orbit is implemented by translating the patient bed while the source and detector rotate.

4.2 Positron Emission Tomography and Single Photon Emission Computed Tomography

In the last section, transmission imaging was discussed. In transmission imaging, the radiation source is placed outside the patient body. The radiation rays (either X-rays or γ-rays) enter the patient from outside, pass through the

patient body, exit the patient, and finally get detected by a detector outside.

This section will change the subject to emission imaging, where the radiation sources are inside the patient body. Radiation is generated inside the patient body, emitted from within, and detected by a detector after it escapes from the patient body.

Radioactive atoms with a short half-life are generated in a cyclotron or a nuclear reactor. Radiopharmaceuticals are then made and injected into a patient (in a peripheral arm vein) to trace disease processes. The patient can also inhale or ingest the radiotracer. Radiopharmaceuticals are carrier molecules with a preference for a certain tissue or disease process. The radioactive substance redistributes itself within the body after the injection. The goal of emission tomography is to obtain a distribution map of the radioactive substance.

Unstable atoms emit gamma rays as they decay. Gamma cameras are used to detect the emitted gamma photons (see Figure 4.10). The cameras detect one photon at a time. These measurements approximate the ray-sums or line-integrals. Unlike the transmission data, we do not need to take the logarithm. SPECT (single photon emission computed tomography) is based on this imaging principle.

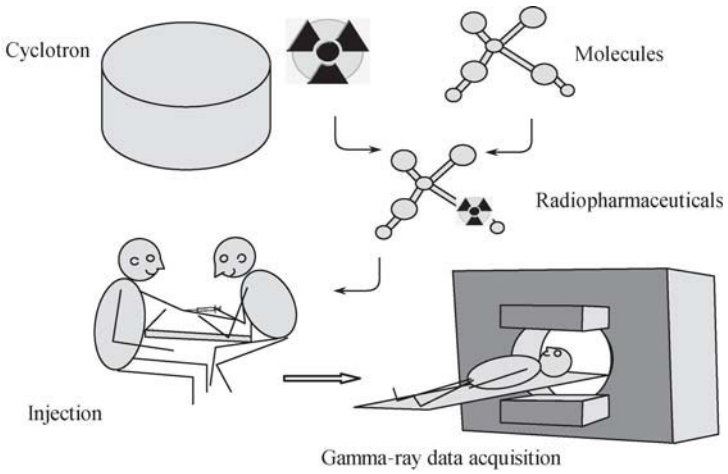


Fig. 4.10. Preparation for a nuclear medicine emission scan.

Some isotopes, for example, O-15, C-11, N-13, and F-18, emit positrons (positive electrons) during radioactive decay. A positron exists in nature only for a very short time before it collides with an electron. When the positron interacts with an electron, their masses are annihilated, creating two gamma photons of 511 keV each. These photons are emitted 180° apart. A spe-

cial gamma camera is used to detect this pair of photons, using coincidence detection technology. Like SPECT, the measurements approximate ray-sums or line-integrals; no logarithm is necessary to convert the data. This is the principle of PET (positron emission tomography) imaging (see Figure 4.11).

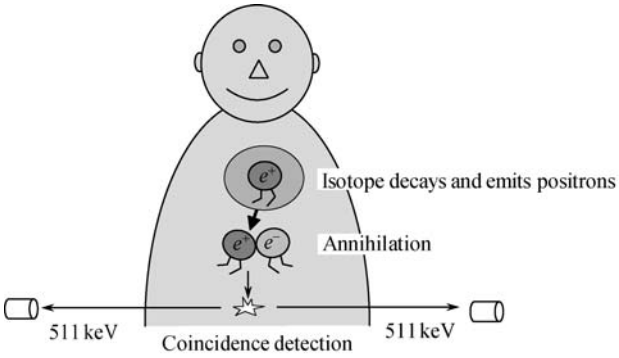


Fig. 4.11. Principle of PET imaging.

The imaging geometry for SPECT (single photon emission computed tomography) is determined by its collimator, which is made of lead septa to permit gamma-rays oriented in certain directions to pass through and stop gamma-rays with other directions. If a parallel-beam or a fan-beam collimator is used, then the data are acquired in the same corresponding form (see Figure 4.12). Similarly, if a cone-beam or a pinhole collimator is used, the imaging geometry is cone-beam (see Figure 4.13). Convergent beam geometries magnify the object so that an image larger than the object can be obtained on the detector.

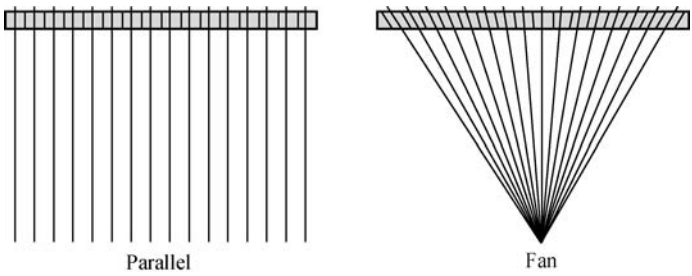


Fig. 4.12. SPECT uses collimators to selected incoming projection ray geometry.

In PET, each measured event determines a line. The imaging geometry is made by sorting or grouping the events according to some desired rules. For example, we can group them into parallel sets (see three different sets in

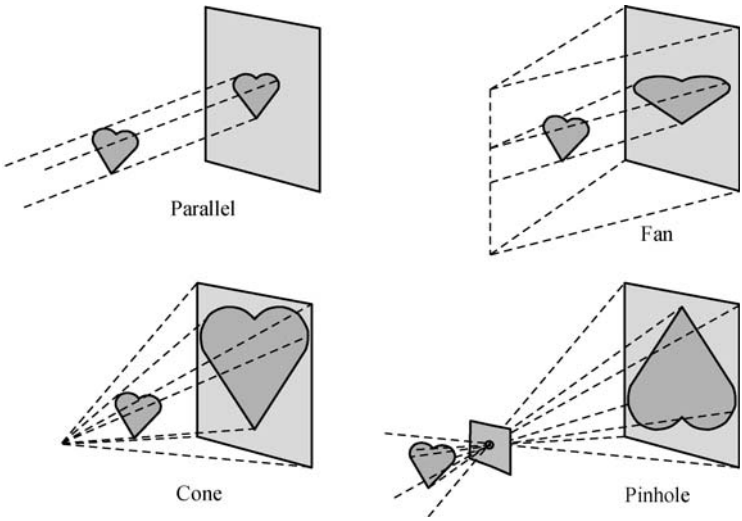


Fig. 4.13. SPECT Collimators can be parallel, convergent, or divergent. They produce different sizes of images.

Figure 4.14). We can also store each event by itself as the list mode.

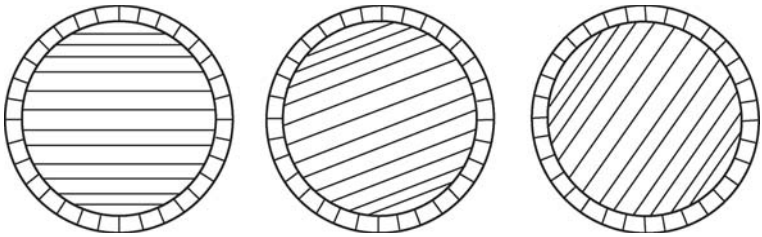


Fig. 4.14. PET data can be grouped into parallel sets.

A typical SPECT image and a PET image are shown in Figure 4.15. When comparing X-ray CT, PET, and SPECT, we observe that X-ray CT has the best resolution and is least noisy, and SPECT has the worst resolution and most noise. Image quality is directly proportional to the photon counts.

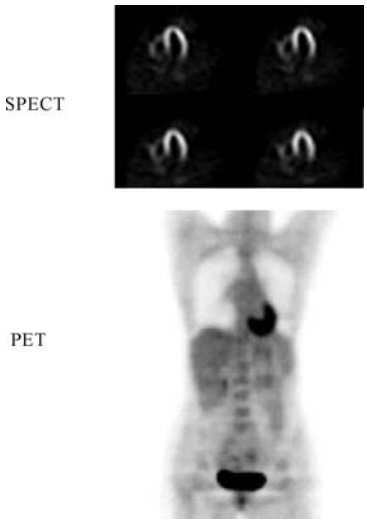


Fig. 4.15. SPECT cardiac images and a PET torso image. The PET image is displayed in the inverse gray scale.

4.3 Attenuation Correction for Emission Tomography

In emission tomography, the gamma-ray photons are emitted from within the patient's body. Not all the photons are able to escape from the patient body; thus they are attenuated when they propagate. The attenuation follows Beer's law, which we have seen in Section 4.1.

In PET, two detectors are required to measure one event with coincidence detection. An event is valid if two detectors simultaneously detect a 511 keV photon. Let us consider the situation depicted in Figure 4.16, where the photons are emitted at an arbitrary location in a non-uniform medium. The photons that reach Detector 1 are attenuated by Path L_1 with an attenuation factor determined by Beer's law. We symbolically represent this attenuation factor as $\exp\left(-\int_{L_1} \mu\right)$. Similarly, the attenuation factor for Path L_2 is $\exp\left(-\int_{L_2} \mu\right)$. The attenuation factor is a number between 0 and 1, and can be treated as a probability. The probability that a pair of photons will be detected by both detectors is the product of the probabilities that each

photon is detected; that is,

$$\exp\left(-\int_{L_1}\mu\right)\times\exp\left(-\int_{L_2}\mu\right)=\exp\left(-\int_{L_1+L_2}\mu\right)=\exp\left(-\int_{L}\mu\right). \tag{4.3.1}$$

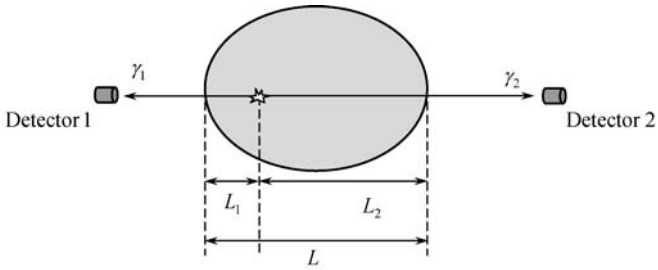


Fig. 4.16. The attenuation of the PET projection is the effect of the total length $L = L_1 + L_2$.

Therefore, the overall attenuation factor is determined by the entire path L, regardless of where the location of the gamma source is along this path.

To do an attenuation correction for PET data, a transmission measurement is required with an external transmission (either X-ray or gamma-ray) source. This transmission measurement gives the attenuation factor $\exp\left(-\int_L \mu\right)$.

The attenuation-corrected line-integral or ray-sum of PET data is obtained as

$$p(s,\theta)=\exp\left(\int_{L(s,\theta)}\mu\right)\times[\text{emission measurement of the path } L(s,\theta)], \tag{4.3.2}$$

where the reciprocal of the attenuation factor is used to compensate for the attenuation effect. Note that there is no need to reconstruct the attenuation map for PET data attenuation correction.

Attenuation correction for SPECT data is much more complicated than this, because an event in SPECT is single photon detection (see Figure 4.17). On the same imaging path, the emission source at a different location has a different attenuation factor. This makes SPECT attenuation correction very difficult. We still do not know how to compensate for the attenuation by processing the projections as done for PET. However, the attenuation effect can be corrected during image reconstruction.

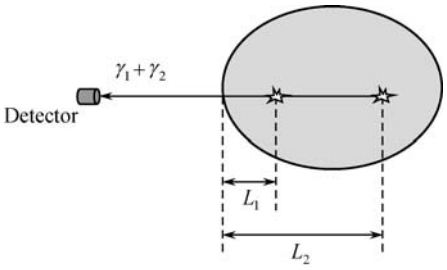


Fig. 4.17. In SPECT the attenuation is a mixed effect of all lengths.

In SPECT, if the attenuator is uniform (i.e., $\mu =$ constant within the body boundary), the filtered backprojection (FBP) image reconstruction algorithm is similar to that for the regular unattenuated data. The attenuation corrected FBP algorithm, developed by Tretiak and Metz, consists of three steps:

(1) Pre-scale the measured projection $p(s,\theta)$ by $e^{\mu d(s,\theta)}$, where the definition of distance $d(s,\theta)$ is given in Figure 4.18. We denote this scaled projection as $\hat{p}(s,\theta)$.

(2) Filter the pre-scaled data with a notched ramp filter (see Figure 4.19).

(3) Backproject the data with an exponential weighting factor $e^{-\mu t}$, where t is defined in Figure 4.20 and is dependent on the location of the reconstruction point as well as the backprojection angle θ.

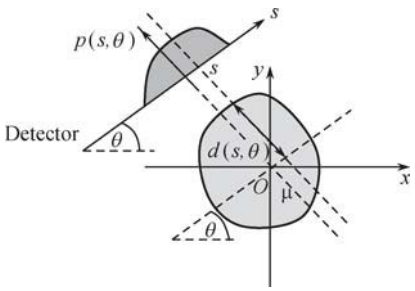


Fig. 4.18. The distance $d(s,\theta)$ is from the boundary of the uniform attenuator to the central line parallel to the detector. A central line is a line passing through the center of rotation.

Similar to the $\mu = 0$ case, the notched ramp filtering can be decomposed into a derivative and a notched Hilbert transform. In the spatial domain, the Hilbert transform is a convolution with a convolution kernel $1/s$. The notched Hilbert transform, on the other hand, is a convolution with a convolution kernel $(\cos \mu s)/s$.

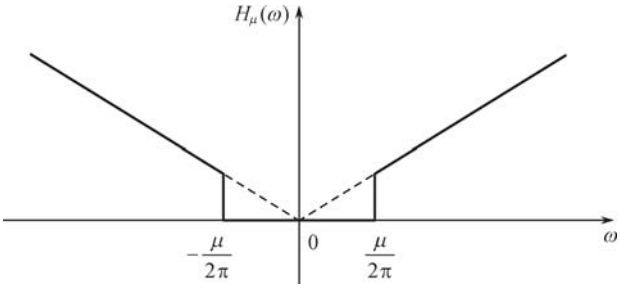


Fig. 4.19. The notched ramp-filter transfer function for image reconstruction in the case of a uniformly attenuated Radon transform. A transfer function is a filter expression given in the Fourier domain, and its inverse Fourier transform is the convolution kernel.

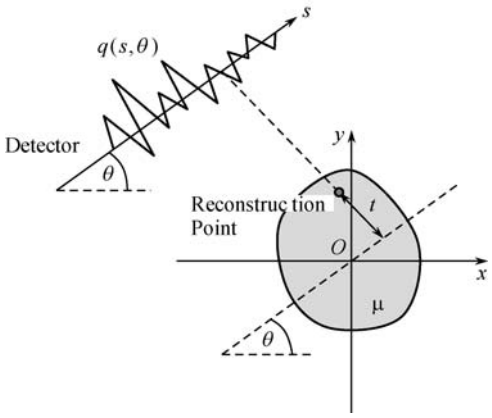


Fig. 4.20. The distance t is from the reconstruction point to the central line parallel to the detector.

In the Fourier domain, the notched Hilbert filter function is shown on the left-hand-side of the graphic equation in Figure 4.21. The cosine function can be decomposed as $\cos(\mu s) = (\mathrm{e}^{\mathrm{i}\,\mu s} + \mathrm{e}^{-\mathrm{i}\,\mu s})/2$. The Fourier transform has a property that multiplication by $\mathrm{e}^{\mathrm{i}\,\mu s}$ in the s domain corresponds to shifting by $\mu/(2\pi)$ in the ω domain (i.e., the Fourier domain). Therefore, the Fourier transform of $(\cos \mu s)/s$ is the combination of two shifted versions (one shifted to the left by $\mu/(2\pi)$ and the other one shifted to the right by $\mu/(2\pi)$) of the Fourier transform of $1/s$ (see Figure 4.21).

If μ is non-uniform, an FBP-type algorithm exists but is rather sophisticated and will be briefly discussed in the next section.

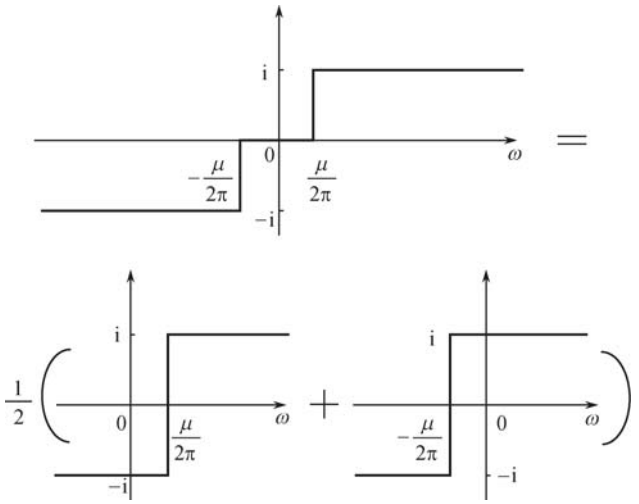


Fig. 4.21. A notched Hilbert transform transfer function can be decomposed as two shifted Hilbert transform transfer functions.

*4.4 Mathematical Expressions

Now we give the mathematical expression of the FBP algorithm for SPECT with uniform attenuation correction. This algorithm has been outlined in Section 4.3. After the pre-scaling step, the scaled projection $\hat{p}(s,\theta)$ can be related to the original image $f(x,y)$ as

$$\hat{p}(s,\theta) = \int_{-\infty}^{\infty} \mathrm{e}^{\mu t} f(s\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp}) \mathrm{d}t. \tag{4.4.1}$$

You can refer to Section 1.5 for notation definitions. The following gives an FBP algorithm using the derivative and the notched Hilbert transform:

$$f(x,y) = \frac{1}{4\pi^2} \int_0^{2\pi} \mathrm{e}^{-\mu(-x\sin\theta + y\cos\theta)} \int_{-\infty}^{\infty} \frac{\cos(\mu(s - x\cos\theta - y\sin\theta))}{s - x\cos\theta - y\sin\theta} \times \frac{\partial \hat{p}(s,\theta)}{\partial s} \mathrm{d}s\mathrm{d}\theta. \tag{4.4.2}$$

There are many other ways to reconstruct SPECT data with uniform attenuation corrections. We can still play the game of switching the order of filtering and backprojection. For example, we can first take the derivative then backproject. This results in an intermediate image $\hat{f}(x,y)$ that is closely

related to the original image $f(x, y)$:

$$\hat{f}(x, y) = f(x, y) * \frac{\cosh(\mu x)}{x}, \tag{4.4.3}$$

which is a 1D convolution (i.e., line-by-line convolution in the x-direction). The deconvolution of this expression to solve for $f(x, y)$ is not an easy task, because the function $\dfrac{\cosh(\mu x)}{x}$ tends to infinity as x goes to infinity. It is impossible to find a function $u(x)$ such that

$$\delta(x) = u(x) * \frac{\cosh(\mu x)}{x} \tag{4.4.4}$$

for $-\infty < x < +\infty$. However, it is possible to find such a function $u(x)$ to make the above expression hold in a small interval, say, $(-1, 1)$. Outside this small interval, $u(x) * \dfrac{\cosh(\mu x)}{x}$ is undefined. This "second best" solution is useful in image reconstruction because our objects are always supported in a small finite region. Unfortunately, we do not yet have a closed-form expression for such a function $u(x)$, and $\hat{f}(x, y) = f(x, y) * \dfrac{\cosh(\mu x)}{x}$ can only be deconvolved numerically.

The advantage of this derivative-then-backprojection algorithm is its ability to exactly reconstruct a region-of-interest with truncated data.

In SPECT imaging with a non-uniform attenuator $\mu(x, y)$, if attenuation correction is required, a transmission scan should be performed in addition to the emission scan. The transmission projections are used to reconstruct the attenuation map $\mu(x, y)$. The following is a reconstruction algorithm that can correct for the non-uniform attenuation. Despite its frightening appearance, it is merely an FBP algorithm:

$$f(x, y) = \frac{1}{4\pi^2}\mathrm{Re}\left\{ \int_0^{2\pi} \frac{\partial}{\partial q}\left[\mathrm{e}^{a_\theta(q,t)-g(q,\theta)} \int_{-\infty}^{\infty} \frac{(\mathrm{e}^g p)(l,\theta)}{q-l}\mathrm{d}l \right]\Bigg|_{q=s} \mathrm{d}\theta \right\}, \tag{4.4.5}$$

where Re means taking the real part, $p(s, \theta)$ is the measured attenuated projection, $s = x\cos\theta + y\sin\theta$, $t = -x\sin\theta + y\cos\theta$, $a_\theta(s, t) = \displaystyle\int_t^\infty \mu(s\boldsymbol{\theta} + \tau\boldsymbol{\theta}^\perp)\mathrm{d}\tau$, $g(s, \theta) = \dfrac{1}{2}[(\mathbf{R} + \mathrm{i}\,\mathbf{HR})\mu](l, \theta)$, $\mathrm{i} = \sqrt{-1}$, $\mathbf{R}$ is the Radon transform operator, and $\mathbf{H}$ is the Hilbert transform operator with respect to variable l. This algorithm was independently developed by Novikov and Natterer.

4.5 Worked Examples

Example 1 In radiology, do the X-ray CT scanners provide images of the distribution of the linear attenuation coefficients within the patient body?

Answer

Not quite. The reconstructed linear attenuation coefficients μ are converted to the so-called CT numbers, defined as

$$\text{CT Number } h = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water}}. \tag{4.5.1}$$

The CT numbers are in Hounsfield units (HU). For water, $h = 0$ HU. For air, $h = -1000$ HU. For bone, $h = 1000$ HU.

Example 2 Is there a central slice theorem for the exponential Radon transform?

Answer

In 1988 Inouye derived a *complex* central slice theorem to relate the uniformly attenuated projections to the object in the Fourier domain. He used a concept of "imaginary" frequency, which was attenuation coefficient. Let the exponential Radon transform be

$$\hat{p}(s,\theta) = \int_{-\infty}^{\infty} \mathrm{e}^{\mu t} f(s\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp})\mathrm{d}t. \tag{4.5.2}$$

Let the 1D Fourier transform of $\hat{p}(s,\theta)$ with respect to s be $P_\mu(\omega,\theta)$ and the 2D Fourier transform of the original object be $F(\omega_x,\omega_y)$. Inouye's complex central slice theorem is expressed as follows:

$$P_\mu(\omega,\theta) = F(\omega\cos(\theta+\nu), \omega\cos(\theta+\nu)) \tag{4.5.3}$$

where $\nu = \dfrac{\mathrm{i}}{2}\ln\dfrac{\omega - \dfrac{\mu}{2\pi}}{\omega + \dfrac{\mu}{2\pi}}$ is an imaginary frequency.

Example 3 In Figure 4.19, the frequency components below $\mu/(2\pi)$ are discarded during image reconstruction. How do the low frequency components of the image get reconstructed?

Answer

We will use Bellini's result to answer this question. Let the attenuation-free Radon transform be

$$p(s,\theta) = \int_{-\infty}^{\infty} f(s\boldsymbol{\theta} s\boldsymbol{\theta}^{\perp})\mathrm{d}t \tag{4.5.4}$$

and the exponential Radon transform be

$$\hat{p}(s,\theta) = \int_{-\infty}^{\infty} \mathrm{e}^{\mu t} f(s\boldsymbol{\theta} s\boldsymbol{\theta}^{\perp})\mathrm{d}t. \tag{4.5.5}$$

The exponential Radon transform is the attenuated projection scaled by the factor $\mathrm{e}^{\mu d(s,\theta)}$.

Then we take the 2D Fourier transform of $p(s,\theta)$ and $\hat{p}(s,\theta)$, and obtain $P(\omega,k)$ and $\hat{P}(\omega,k)$, respectively. Bellini's relationship is given as

$$P(\omega,k) = \left(\frac{\omega^2}{\sqrt{\omega^2 + \left(\frac{\mu}{2\pi}\right)^2} + \frac{\mu}{2\pi}} \right)^k \times \hat{P}\left(\sqrt{\omega^2 + \left(\frac{\mu}{2\pi}\right)^2}, k \right) \text{ for } \omega \geqslant 0 \text{ and } k \geqslant 0. \tag{4.5.6}$$

This relationship implies that the frequency ω has been shifted to $\sqrt{\omega^2 + (\mu/(2\pi))^2}$. The attenuation procedure during data generation actually shifts the "frequency." The lowest frequency in the image corresponds to the frequency at $\mu/(2\pi)$ in the projections. Therefore, all frequency components are preserved.

Example 4 Consider a 2D PET imaging problem shown in Figure 4.22, where the object consists of two compartments R_1 and R_2. The radionuclide concentration is ρ_1 in R_1 and ρ_2 in R_2. The attenuation coefficient μ is the same for both compartments. Calculate the PET coincident measurement p and the attenuation corrected measurement p_c.

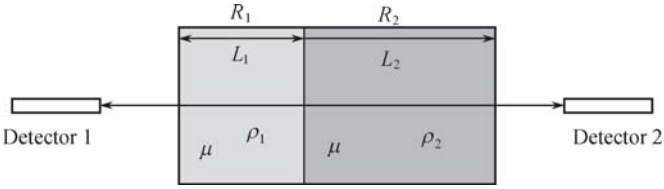


Fig. 4.22. Concentrations are different in the two compartments.

Solutions

The coincidence measurement is

$$p = (L_1\rho_1 + L_2\rho_2)\mathrm{e}^{-\mu(L_1+L_2)}. \tag{4.5.7}$$

The attenuation corrected measurement is

$$p_c = L_1\rho_1 + L_2\rho_2. \tag{4.5.8}$$

4.6 Summary

- The working principle for X-ray CT measurement is Beer's law. One needs to take the logarithm to convert the CT measurements into line-integrals.
- PET and SPECT measure line-integrals directly. However, these measurements suffer from photon attenuation within the patient.
- Attenuation correction for PET can be achieved by the pre-scaling method. The scaling factor is obtained by transmission measurements.
- Attenuation correction for SPECT is difficult and cannot be done by pre-scaling. Attenuation correction is a built-in feature in SPECT reconstruction algorithms. FBP algorithms exist for uniform attenuator and for non-uniform attenuator as well. However, the FBP algorithm for the non-uniform attenuator is very complicated to implement.
- The readers are expected to understand the differences between transmission and emission tomography, and understand the different attenuation effects in PET and SPECT.

Problems

Problem 4.1 The line-integral data p from transmission tomography can be obtained by $p = \ln \frac{N_0}{N}$, where N_0 and N are the numbers of the photons entering and leaving the patient body, respectively. The noise characteristics of the photon numbers are Poisson distributed. In practice, the entering number N_0 is very large and can be treated as a constant. Prove that the mean value (i.e., the expected value) and the variance of the line-integral p can be approximated as p and $1/N$, respectively.

Problem 4.2 The object to be imaged is a 2D uniform disc with a radius R. The linear attenuation coefficient of the disc is a constant μ and its radioactivity line density is a constant ρ. The center of the disc is at the center of the detector rotation. Find the expressions of the attenuated projection data $p(s, \theta)$ for SPECT and PET cases, respectively.

Problem 4.3 Prove that the inverse Fourier transform of the transfer function $H_\mu(\omega)$ of the filter shown in Figure 4.19 is the convolution kernel $\frac{\cos(\mu s)}{s}$.

References

1. Bellini S, Piacenti M, Caffario C et al (1979) Compensation of tissue absorption in emission tomography. IEEE Trans Accoust Speech, Signal Processing ASSP-27: 213 – 218
2. Bukhgeim AA, Kazantsev SG (2002) Inversion formula for the fan-beam attenuated Radon transform in a unit disk. The Sobolev Institute of Mathematics, Russian Academy of Science Siberian Branch
3. Guillement JP, Novikov R (2004) A noise property analysis of single-photon emission computed tomography data. Inverse Probl 20: 175 – 198
4. Gullberg GT (1979) The Attenuated Radon Transform: Theory and Application in Medicine and Biology, Ph.D. Dissertation, Lawrence Berkeley Laboratory, University of California, Berkeley, Culifornia
5. Gullberg GT, Budinger TF (1981) The use of filtering methods to compensate for constant attenuation in single photon emission computed tomography. IEEE Trans Biomed Eng 28: 142 – 157
6. Hawkins WG, Leichner PK, Yang NC (1988) The circular harmonic transform for SPECT reconstruction and boundary conditions on the Fourier transform of the sinogram. IEEE Trans Med Imaging 7: 135 – 148
7. Hsieh J (2003) Computed Tomography: Principles, Design, Artifacts, and Recent Advances, SPIE, Bellingham, Washington
8. Huang Q, Zeng GL, Wu J ((2006) An alternative proof of Bukjgeim and Kazantsev's inversion formula for attenuated fan-beam projections. Med Phys 33: 3983 – 3987
9. Huang Q, You J, Zeng GL et al (2009) Reconstruction from uniformly attenuated SPECT projection data using the DBH method. IEEE Trans. Med. Imaging 28: 17 – 29
10. Inouye T, Kose K, Hasegawa A (1988) Image reconstruction algorithm for single-photon-emission computed tomography with uniform attenuation. Phys Med Biol 34: 299 – 304
11. Kunyansky LA (2001) A new SPECT reconstruction algorithm based upon the Novikov's explicit inversion formula. Inverse Probl 17: 293 – 306
12. Markoe A (1984) Fourier inversion of the attenuated X-ray transform. SIAM J Math Anal 15: 718 – 722
13. Metz CE, Pan X (1995) A unified analysis of exact methods of inverting the 2-D exponential Radon transform, with implications for noise control in SPECT. IEEE Trans Med Imaging 17: 643 – 658
14. Natterer F (2001) Inversion of the attenuated Radon transform. Inverse Probl 17: 113 – 119
15. Noo F, Wagner JM (2001) Image reconstruction in 2D SPECT with 180° acquisition. Inverse Probl 17: 1357 – 1371
16. Novikov RG (2002) An inversion formula for the attenuated X-ray transformation. Ark Math 40: 145 – 167
17. Pan X, Kao C, Metz CE (2002) A family of π-scheme exponential Radon transforms and the uniqueness of their inverses. Inverse Probl 18: 825 – 836
18. Puro A (2001) Cormack-type inversion of exponential Radon transform. Inverse Probl 17: 179 – 188

19. Rullgård H (2002) An explicit inversion formula for the exponential Radon transform using data from 180°. Ark Math 42: 145 – 167
20. Rullgård H (2004) Stability of the inverse problem for the attenuated Radon transform with 180° data. Inverse Probl 20: 781 – 797
21. Tang Q, Zeng GL, Gullberg GT (2007) A Fourier reconstruction algorithm with constant attenuation compensation using 180° acquisition data for SPECT. Phys Med Biol 52: 6165 – 6179
22. Tretiak O, Metz CE (1980) The exponential Radon transform. SIAM J Appl Math 39: 341 – 354
23. You J, Zeng GL, Liang Z (2005) FBP algorithms for attenuated fan-beam projections. Inverse Probl 21: 1179 – 1192
24. Wang G, Lin T, Cheng P et al (1991) Scanning cone-beam reconstruction algorithms for X-ray microtomography. Proc SPIE 1556: 99 – 112
25. Wang G, Yu H, DeMan B (2008) An outlook on X-ray CT research and development. Med Phys 35: 1051 – 1064
26. Webster JG (1998) Medical Instrumentations: Application and Design, 3rd edn, Wiley, New York
27. Wolbarst AB (1993) Physics of Radiology, Appleton and Lange, Norwalk, CT
28. Zeng GL, Gullberg GT (2009) Exact emission SPECT reconstruction with truncated transmission data. Phys Med Biol 54: 3329 – 3340

5 3D Image Reconstruction

This chapter is focused on 3D tomographic imaging. In 3D, we will consider the parallel line-integral projections, parallel plane-integral projections, and cone-beam line-integral projections, separately. For the 3D parallel line-integral projections and parallel plane-integral projections, there exist the central slice theorems, from which the image reconstruction algorithms can be derived. However, for the cone-beam projections the situation is different; there is no central slice theorem for cone-beam. We have to somehow establish a relationship between the cone-beam projections and the 3D image itself. Since the cone-beam image reconstruction is an active research area, this chapter spends a significant effort on discussing cone-beam reconstruction algorithms, among which the Katsevich algorithm is the latest and the best one.

5.1 Parallel Line-Integral Data

In many cases, 3D image reconstruction can be decomposed into a series of slice-by-slice 2D image reconstructions if the projection rays can be divided into groups, where each group contains only those rays that are confined within a transaxial slice (see Figure 5.1 Left).

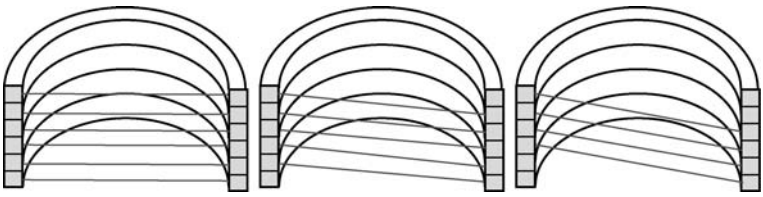


Fig. 5.1. The measurement rays can be in the planes perpendicular to the axial direction and can also be in the slant planes.

In other cases, the projection rays run through transaxial slices, where the slice-by-slice 2D reconstruction approach does not work (see of Figures 5.1 Middle and Right).

The foundation for 2D parallel-beam image reconstruction is the central slice theorem (see Section 2.2). The central slice theorem in 3D states that the 2D Fourier transform $P(\omega_u, \omega_v, \boldsymbol{\theta})$ of the projection $p(u, v, \boldsymbol{\theta})$ of a 3D function $f(x, y, z)$ is equal to a slice through the origin of the 3D Fourier transform $F(\omega_x, \omega_y, \omega_z)$ of that function which is parallel to the detector (see Figure 5.2). Here, $\boldsymbol{\theta}$ is the normal direction of the u-v plane and the ω_u-ω_v plane. The direction $\boldsymbol{\theta}$ represents a group of rays that are parallel to $\boldsymbol{\theta}$.

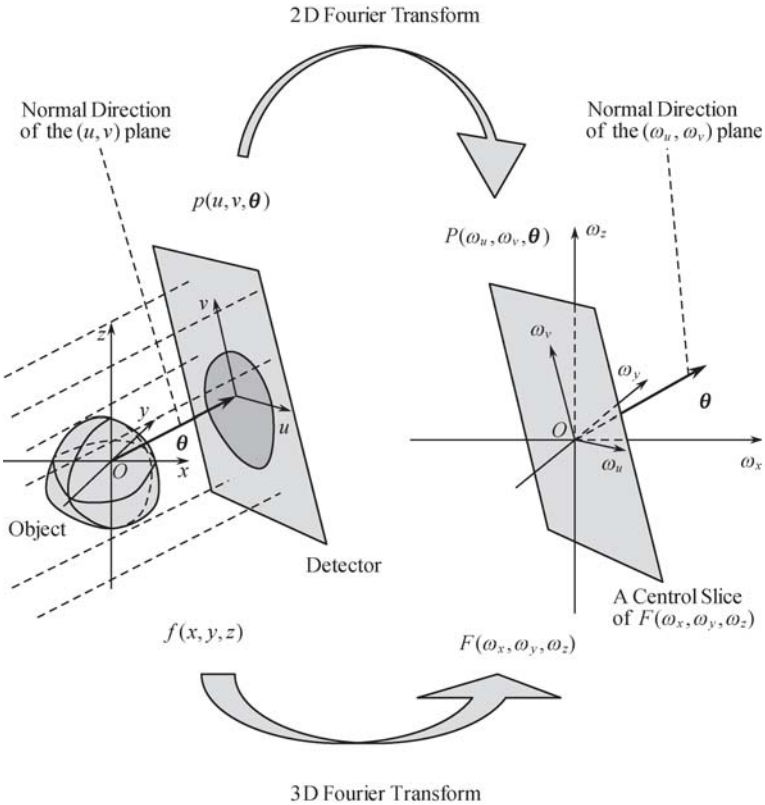


Fig. 5.2. The central slice theorem for the 3D line-integral projections.

Based on this central slice theorem, we can determine some specific trajectories of $\boldsymbol{\theta}$ so that we are able to fill up the $(\omega_x, \omega_y, \omega_z)$ Fourier space. One such option is shown in Figure 5.3, where the trajectory of $\boldsymbol{\theta}$ is a great circle. A great circle is a circle with unit radius that lies on the surface of the unit sphere (see Figure 5.4). Each unit vector $\boldsymbol{\theta}$ on the great circle corresponds to a measured $P(\omega_u, \omega_v, \boldsymbol{\theta})$ plane in the $(\omega_x, \omega_y, \omega_z)$ Fourier space. After the

unit vector $\boldsymbol{\theta}$ completes the great circle, the measured $P(\omega_u, \omega_v, \boldsymbol{\theta})$ planes fill up the entire $(\omega_x, \omega_y, \omega_z)$ Fourier space. In fact, due to symmetry, if the unit vector $\boldsymbol{\theta}$ completes *half* of the great circle, a complete data set is obtained.

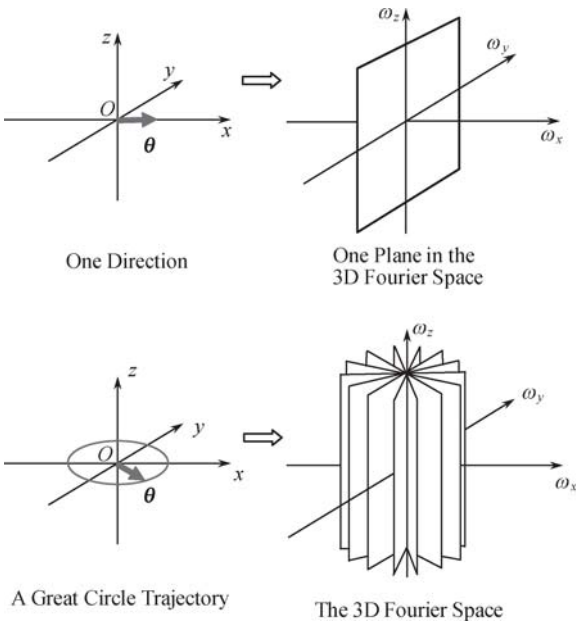


Fig. 5.3. One measuring direction gives a measured plane in the Fourier space. A great circle trajectory provides full Fourier space measurements.

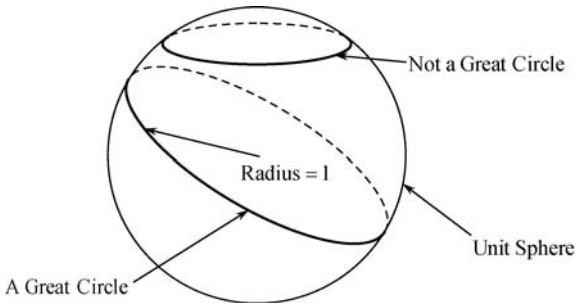


Fig. 5.4. A great circle is a unit circle on the unit sphere.

The above example can be generalized, as stated in Orlov's condition: A complete data set can be obtained if every great circle intersects the trajectory of the unit vector $\boldsymbol{\theta}$, which is the direction of the parallel rays. The trajectory can be curves on the sphere and can also be regions on the sphere. Some examples of the $\boldsymbol{\theta}$ trajectories (shaded area) are shown in Figure 5.5, where

the first three satisfy Orlov's condition, and the last two do not.

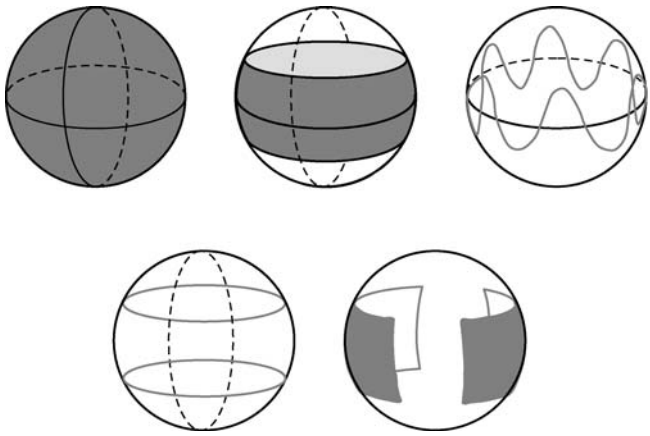


Fig. 5.5. The directional vector trajectories are illustrated as curves or shaded areas on the unit sphere. The trajectories on the top row satisfy Orlov's condition; the trajectories on the bottom row do not satisfy the condition.

The image reconstruction algorithm depends on the trajectory of the direction vector $\boldsymbol{\theta}$ geometry. The basic algorithm development can follow the guidelines below.

5.1.1 Backprojection-then-Filtering

If the data are sufficiently measured, the 3D image can be exactly reconstructed. Like 2D image reconstruction, one can reconstruct an image either by performing the backprojection first or by performing the backprojection last. If the backprojection is performed first, the algorithm is a backprojection-then-filtering algorithm, and it is described in the following steps.

(1) Use an arbitrary point source and find the 3D projection/backprojection PSF h (defined in Figure 3.2). If the original image is $f(x, y, z)$ and the backprojected image is $b(x, y, z)$, then

$$b = f ** * h, \tag{5.1.1}$$

where "$***$" denotes the 3D convolution. For example, if the trajectory of the directional vector $\boldsymbol{\theta}$ is the full unit sphere as shown in the leftmost case in the first row of Figure 5.5, the PSF is

$$h(x, y, z) = \frac{1}{x^2 + y^2 + z^2} = \frac{1}{r^2}, \tag{5.1.2}$$

where r is the distance to the point source. In 2D, this projection/backprojection PSF is $1/r$. This implies that in 3D the PSF is sharper than that in 2D because the PSF falls off at a faster rate as r increases.

(2) Take the 3D Fourier transform of the relationship $b = f ** * h$ and obtain

$$B(\omega_x, \omega_y, \omega_z) = F(\omega_x, \omega_y, \omega_z) H(\omega_x, \omega_y, \omega_z). \tag{5.1.3}$$

After this Fourier transform, b, f, and h become B, F, and H, respectively, and convolution becomes multiplication. If we again use the example in the leftmost case in the upper row of Figure 5.5, the transfer function H is

$$H(\omega_x, \omega_y, \omega_z) = \frac{\pi}{\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}}. \tag{5.1.4}$$

Thus, a 3D ramp-filter

$$\frac{1}{H(\omega_x, \omega_y, \omega_z)} = \frac{\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}}{\pi} \tag{5.1.5}$$

can be used for image reconstruction in this case.

Solve for F as

$$F(\omega_x, \omega_y, \omega_z) = B(\omega_x, \omega_y, \omega_z) \frac{\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}}{\pi}. \tag{5.1.6}$$

Finally, the image $f(x, y, z)$ is obtained by taking the 3D inverse Fourier transform of F.

In general, 3D line-integral data are measured with heavy redundancy. Therefore, the image reconstruction algorithm is not unique because you can always weigh redundant data differently.

5.1.2 Filtered Backprojection

In a filtered backprojection algorithm, the projection $p(u, v, \boldsymbol{\theta})$ is first filtered by a 2D filter (or a 2D convolution), obtaining $q(u, v, \boldsymbol{\theta})$. A backprojection of the filtered data $q(u, v, \boldsymbol{\theta})$ gives the reconstruction of the image $f(x, y, z)$.

Due to data redundancy, the 2D filter is not unique. The filter is usually different depending on different data orientation $\boldsymbol{\theta}$. One way to obtain a Fourier-domain filter is through the central slice theorem.

In Section 5.1.1 above, we had a projection/backprojection PSF $h(x, y, z)$ and its Fourier transform $H(\omega_x, \omega_y, \omega_z)$. If we let

$$G(\omega_x, \omega_y, \omega_z) = 1/H(\omega_x, \omega_y, \omega_z),$$

then G is the Fourier domain filter in the backprojection-then-filtering algorithm. The Fourier domain 2D filter for projection $p(u, v, \boldsymbol{\theta})$ can be selected

as the central slice of $G(\omega_x, \omega_y, \omega_z)$ with the normal direction $\boldsymbol{\theta}$ (see Figure 5.2). Note that the filter in general depends on the direction $\boldsymbol{\theta}$.

5.2 Parallel Plane-Integral Data

In 3D, the parallel plane-integral $p(s, \boldsymbol{\theta})$ of an object $f(x, y, z)$ is referred to as the Radon transform (see Figure 5.6). In 2D, the Radon transform is the parallel line-integral $p(s, \theta)$ of $f(x, y)$. In a general n-D space, the $(n-1)$-D hyperplane-integral of an n-D function f is called the Radon transform. On the other hand, 1D integral of the object is called the line-integral, ray-sum, X-ray transform, or ray transform. In 2D, the Radon transform and the X-ray transform are the same thing.

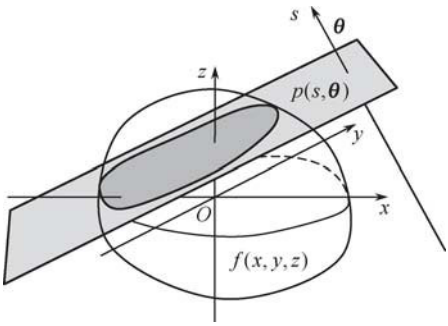


Fig. 5.6. In 3D, the plane integral of an object is the Radon transform.

Unlike the line-integral data, the plane-integral data are not popular in medical imaging. Nevertheless, Radon transform in 3D is still worthwhile to investigate because it has a simple and nice inversion and can be used to solve other related imaging problems.

To study the Radon transform in 3D, we imagine a 1D detector that is able to measure plane-integrals $p(s, \boldsymbol{\theta})$ with the planes orthogonal to the detector. The detector is along the direction $\boldsymbol{\theta}$. The central slice theorem for the Radon transform in 3D states that the 1D Fourier transform $P(\omega, \boldsymbol{\theta})$ of the projection $p(s, \boldsymbol{\theta})$ of a 3D function $f(x, y, z)$ is equal to a 1D profile through the origin of the 3D Fourier transform $F(\omega_x, \omega_y, \omega_z)$ of that function which is parallel to the detector (see Figure 5.7). Here, $\boldsymbol{\theta}$ is the direction of the 1D detector and the 1D profile in the $(\omega_x, \omega_y, \omega_z)$ space.

We observe from Figure 5.7 that each detector position only measures the frequency components along one line in the $(\omega_x, \omega_y, \omega_z)$ space. The direction $\boldsymbol{\theta}$ must go through a half unit-sphere to get enough measurements for image

reconstruction. After the data are acquired, the image reconstruction algorithm is simple and is in the form of filtered backprojection. This form is also referred to as the Radon inversion formula.

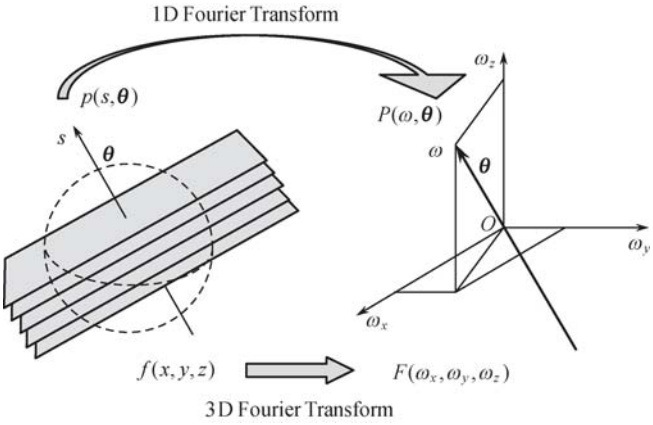


Fig. 5.7. The central slice theorem for the 3D Radon transform.

In order to reconstruct the image, first take the second order derivative of $p(s, \boldsymbol{\theta})$ with respect to variable s. This step is called filtering. Then backproject the filtered data to the 3D image array. You will not find an image reconstruction algorithm simpler than this.

The 3D backprojector in the 3D Radon inversion formula backprojects a point into a 2D plane. There is a trick to perform the 3D backprojection with the Radon data. This trick is to perform the 3D backprojection in two steps, and each step is a 2D backprojection. In the first step, a point is backprojected into a line [see Figure 5.8 (a)]. All data points along a line are backprojected into a set of parallel lines, and these lines are in a 2D plane.

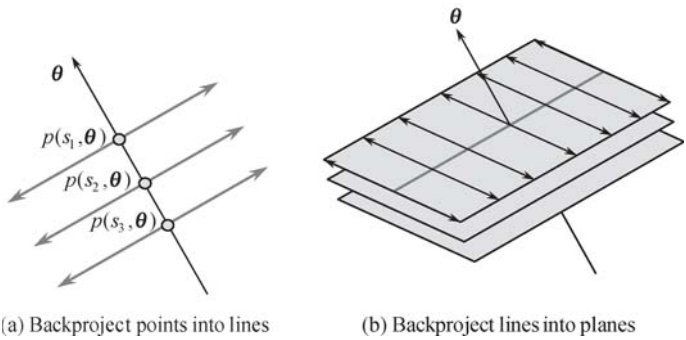


Fig. 5.8. 3D Radon backprojection is implemented as two steps: a point to a line, then a line to a plane.

In the second step, each line is backprojected into a 2D plane [see Figure 5.8 (b)]. The backprojection directions in these two steps are orthogonal to each other.

5.3 Cone-Beam Data

Cone-beam image reconstruction is considerably more complex than that of parallel line-integral and parallel plane-integral data. There is no equivalent central slice theorem known to us. Cone-beam imaging geometry (see the two lower figures in Figure 4.13) is extremely popular, for example, in X-ray CT and in pinhole SPECT; we will spend some effort to talk about its reconstruction methods.

First, we have a cone-beam data-sufficiency condition (known as Tuy's condition): Every plane that intersects the object of interest must contain a cone-beam focal point. This condition is very similar to the fan-beam data sufficiency condition: Every line that intersects the object of interest must contain a fan-beam focal point.

In Figure 5.9, the circular cone-beam focal-point orbit does not satisfy Tuy's condition. If we draw a plane cutting through the object above (or below) the orbit plane and parallel to the orbit plane, this plane will never intersect the circular orbit. The helical and circle-and-lines orbits shown in

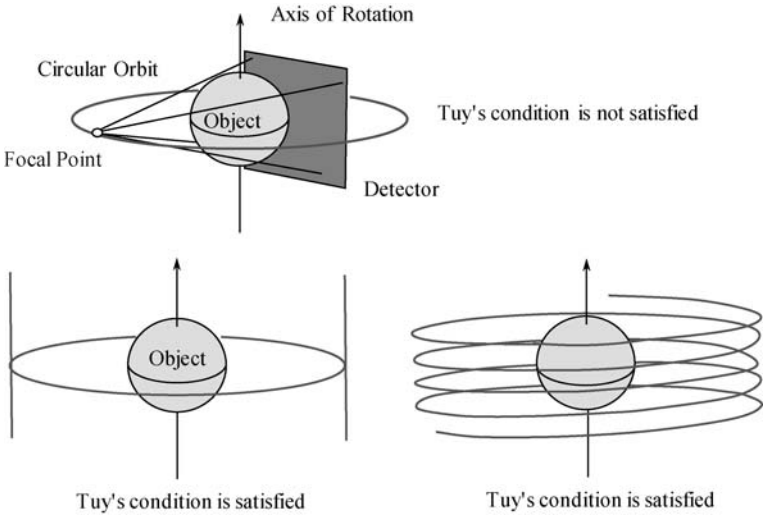


Fig. 5.9. The circular orbit does not satisfy Tuy's conditions. The circle-and-lines and the helix orbits satisfy the conditions.

Figure 5.9 satisfy Tuy's condition, and they can be used to acquire cone-beam projections for exact image reconstruction. Modern CT uses helical orbit to acquire projection data (see Figure 4.8).

5.3.1 Feldkamp's Algorithm

Feldkamp's cone-beam algorithm is dedicated to the circular focal point trajectory. It is a filtered backprojection algorithm and is easy to use. Because the circular trajectory does not satisfy Tuy's condition, Feldkamp's algorithm can only provide approximate reconstructions. Artifacts can appear especially at locations away from the orbit plane. Artifacts include reduction in activity in the regions away from the orbit plane, cross-talk between adjacent slices, and undershoots.

Feldkamp's algorithm is practical and robust. Cone angle, as defined in Figure 5.10, is an important parameter in cone-beam imaging. If the cone angle is small, say less than 10°, this algorithm gives fairly good images. At the orbit plane, this algorithm is exact. This algorithm also gives an exact reconstruction if the object is constant in the axial direction (e.g., a tall cylinder).

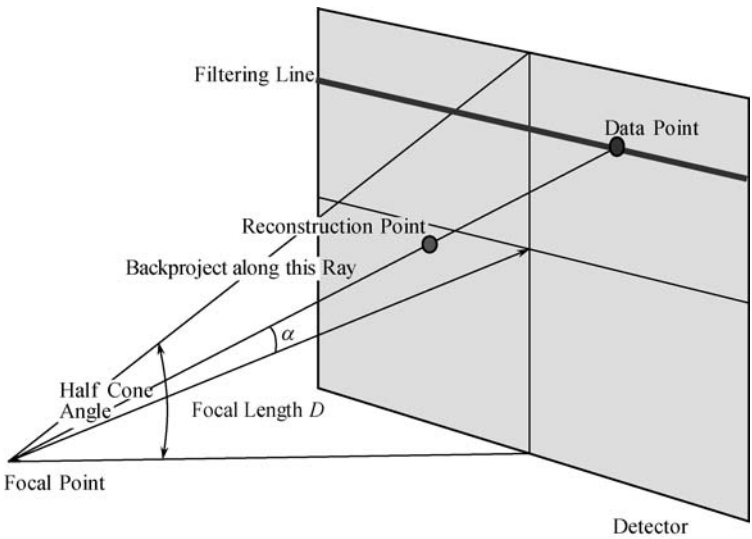


Fig. 5.10. The coordinate system for Feldkamp's cone-beam algorithm.

Feldkamp's cone-beam algorithm (see Section 5.4) is nothing but a modified fan-beam FBP algorithm (see Section 3.4.1). It consists of the following steps:

(1) Pre-scale the projections by a cosine function $\cos\alpha$ (see Figure 5.10 for angle α).

(2) Row-by-row ramp filter the pre-scaled data.

(3) Cone-beam backproject the filtered data with a weighting function of the distance from the reconstruction point to the focal point.

5.3.2 Grangeat's Algorithm

Feldkamp's algorithm converts the cone-beam image reconstruction problem to the fan-beam image reconstruction problem; Grangeat's algorithm, on the other hand, converts it to the 3D Radon inversion problem (see Section 5.2). Feldkamp's algorithm is derived for the circular orbit; Grangeat's algorithm can be applied to any orbit. If the orbit satisfies Tuy's condition, Grangeat's algorithm can provide an exact reconstruction.

Grangeat's method first tries to convert cone-beam ray-sums to plane-integrals, by calculating the line-integrals on the cone-beam detector (see Figure 5.11).

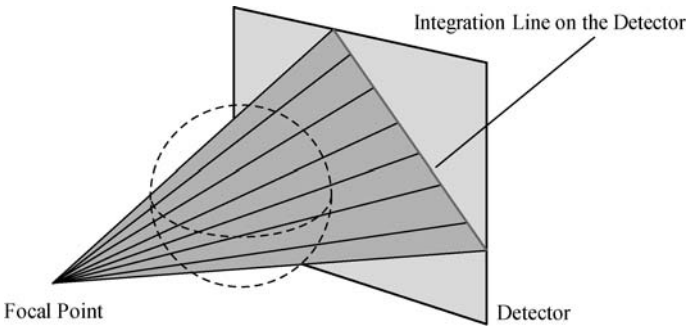


Fig. 5.11. Integration along a line on the cone-beam detector gives a weighted plane integral of the object.

We observe that the line-integral on the detector plane gives a weighted plane-integral of the object with a special non-uniform weighting function $1/r$. Here r is the distance to the cone-beam focal point. We must remove this $1/r$ weighting before we can obtain a regular un-weighted plane-integral.

From Figure 5.12 we observe that the angular differential $\mathrm{d}\alpha$ multiplied by the distance r equals the tangential differential $\mathrm{d}t$: $r\mathrm{d}\alpha = \mathrm{d}t$. If we perform the angular derivative on the $1/r$ weighted plane-integral, we will cancel out this $1/r$ weighting factor by the factor r, obtaining the derivative of the un-weighted plane-integral with respect to variable t, which is in the normal direction of the plane, that is

$$\frac{\partial\left(\frac{1}{r}\text{ weighted plane integral}\right)}{\partial\alpha} = \frac{\partial\,(\text{Radon transform})}{\partial t}. \tag{5.3.1}$$

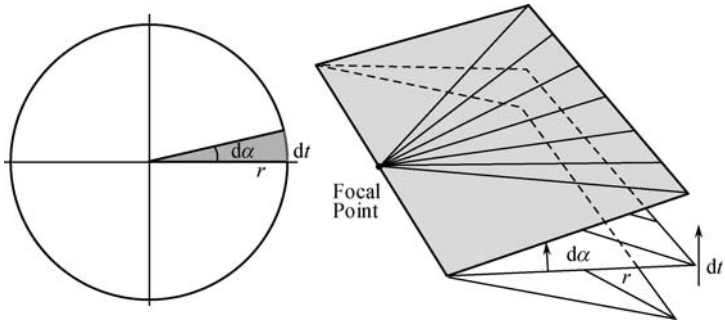


Fig. 5.12. The differential dt in the tangent direction is equal to the angular differential dα times the distance r.

Recall that the Radon inversion formula is the second order derivative of the plane-integral with respect to t, followed by the 3D Radon backprojection. Therefore, a cone-beam image reconstruction algorithm can be implemented as follows:

(1) Form all possible (all orientations and all locations) line-integrals on each detector plane (see Figure 5.11), obtaining $1/r$ weighted plane-integrals.

(2) Perform the angular derivative on results from (1).

(3) Rebin the results from (2) to the $(s, \boldsymbol{\theta})$ Radon space (see Figure 5.6).

(4) Take the derivative of the results of (3) with respect to t, in the normal direction of the plane.

(5) Perform the 3D Radon backprojection (see Figure 5.8).

We now expand on Step (3). For a practical focal point orbit, the $(s, \boldsymbol{\theta})$ Radon space is not uniformly sampled. Data redundancy must be properly weighted. For example, if the value at a particular Radon space location $(s, \boldsymbol{\theta})$ is measured 3 times, then after rebinning this value needs to be divided by 3.

Grangeat's algorithm is not a filtered backprojection algorithm, and it requires data rebinning, which can introduce large interpolation errors.

5.3.3 Katsevich's Algorithm

Katsevich's cone-beam algorithm was initially developed for the helical orbit cone-beam geometry and was later extended to more general orbits. Katse-

vich's algorithm is in the form of FBP, and the filtering can be made shift-invariant. By shift-invariant we mean that the filter is independent of the reconstruction location.

Using a helical orbit (see Figure 5.9), Tuy's data sufficiency condition is satisfied. The main issue in developing an efficient cone-beam FBP algorithm is to properly normalize the redundant data. Katsevich uses two restrictions to handle this issue.

It can be shown that for any point (x, y, z) within the volume surrounded by the helix, there is a unique line segment that passes through the point (x, y, z) where both endpoints touch two different points on the helix and are separated by less than one pitch, say, at s_b and s_t shown in Figure 5.13. This particular line segment is referred to as the π-segment (or π-line). The first restriction is the use of the cone-beam measurements that are measured only from the helix orbit between s_b and s_t.

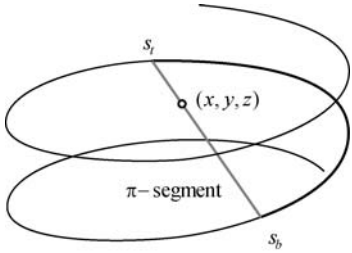


Fig. 5.13. For any point (x, y, z) inside the helix there is one and only one π-segment.

The second restriction is the filtering direction, which handles the normalization of redundant data. In order to visualize the data-redundancy problem, let us look at three cone-beam image reconstruction problems: (1) Data are sufficient and not redundant; (2) Data are insufficient; (3) Data are sufficient but redundant.

(1) The scanning cone-beam focal-point orbit is an arc (i.e., an incomplete circle). The point to be reconstructed is in the orbit plane and on the line that connects the two end-points of the arc. The line connecting the arc's endpoints is the π-segment of the reconstruction point. For our special case that the object is only a point, the cone-beam measurement of this point at one focal-point position can provide a set of plane integrals of the object. Those planes all contain the line that connects the focal point and the reconstruction point. After the focal point goes through the entire arc, all plane integrals are obtained. Recall the central slice theorem for the 3D Radon transform; an exact reconstruction requires that the planar integrals of the object are available for all directions $\boldsymbol{\theta}$, which is indicated in Figure 5.14 Top.

The unit vector $\boldsymbol{\theta}$ traces a circle (let us call it a $\boldsymbol{\theta}$-circle) in a plane that

is perpendicular to the orbit plane, and the line connecting the focal point and the reconstruction point is normal to this plane. Thus, every focal point on the arc orbit corresponds to a $\boldsymbol{\theta}$-circle. Let the focal point travel through the entire arch, then the corresponding $\boldsymbol{\theta}$-circles form a complete unit sphere, which we can call a $\boldsymbol{\theta}$ sphere (see Figure 5.14 Bottom).

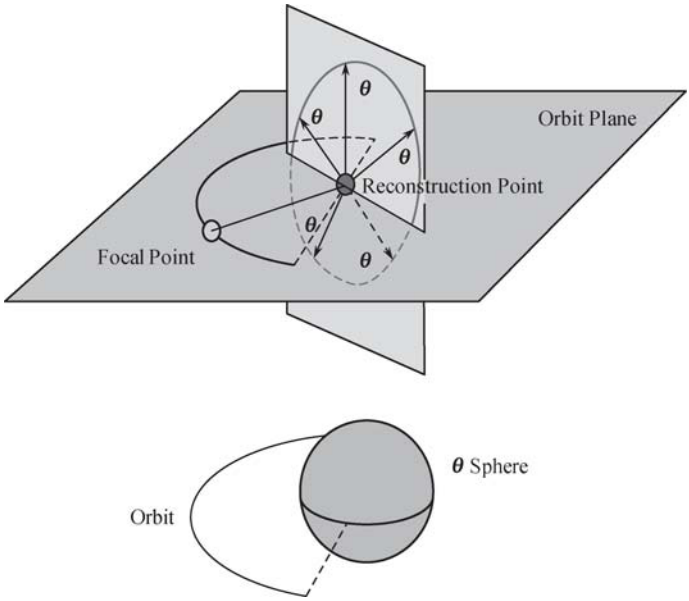


Fig. 5.14. Top: At one focal point position, the directional vectors trace a unit circle. A directional vector represents a measured plane integral. The plane containing this unit circle is perpendicular to the line that connects the reconstruction point and the focal point. Bottom: When the focal point travels through the entire arc, the directional vectors trace a full unit sphere.

(2) If the same arc orbit as in (1) is used and the object to be reconstructed is a point above the orbit plane, then the data are insufficient. When we draw the $\boldsymbol{\theta}$-circle for each focal point position, the $\boldsymbol{\theta}$-circle is not in a vertical plane, but in a plane that has a slant angle. If we let the focal point travel through the entire arc, the corresponding $\boldsymbol{\theta}$-circles do not form a complete unit sphere anymore — both the Arctic Circle and the Antarctic Circle are missing (see Figure 5.15).

(3) Now we consider a helical orbit with the reconstruction point inside. We determine the π-segment for the reconstruction point and find the π-segment endpoints. The segment of the helical orbit between the π-segment endpoints are shown in Figure 5.16.

In Figure 5.16, the $\boldsymbol{\theta}$ sphere is fully measured. In fact, it is over measured. The small triangle-like regions near the North and South Pole are measured 3 times. Let us look at this situation in another way. Draw a plane passing

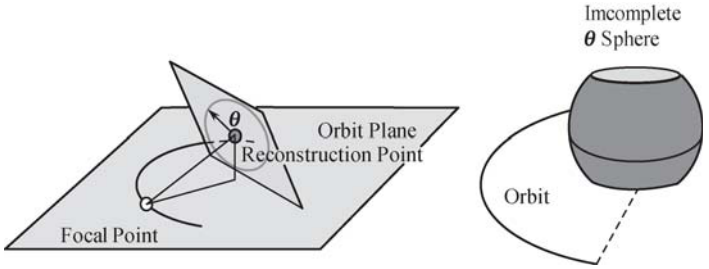


Fig. 5.15. If the reconstruction point is above the orbit plane, the unit sphere is not completely covered by the all the directional vectors.

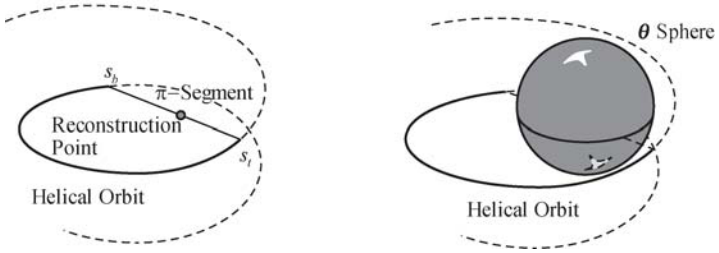


Fig. 5.16. For the helical orbit, the unit sphere is completely measured. At the North Pole and South Pole area, a small region is covered (i.e., measured) three times.

through the reconstruction point. In most cases, the plane will intersect the piece of helix orbit at one point. However, there is a small chance that the plane can intersect the piece of helix at three points (see a side view of the helix in Figure 5.17).

If a plane intersects the orbit three times, we must normalize the data by assigning a proper weight to each measurement. The sum of the weights must be 1. Common knowledge teaches us that we should use all available data, and weigh the redundant measurement by the inverse of its noise variance. However, in order to derive a shift-invariant FBP algorithm, we need to do something against common sense. In Katsevich's algorithm, a measurement is weighted by either $+1$ or -1. If a plane is measured once, we must make sure that it is weighted by $+1$. If a plane is measured 3 times, we need to make sure that two of them are weighted by $+1$ and the third one of them is weighted by -1. In other words, we keep one and throw the other two away. Ouch! Further discussion about the weighting and filtering will be given in the next section.

In Katsevich's algorithm, the normalization issue is taken care of by selecting a proper filtering direction. Here, filtering means performing the Hilbert transform. After the filtering direction and the normalization issues have been taken care of, Katsevich's algorithm is implemented with the following steps:

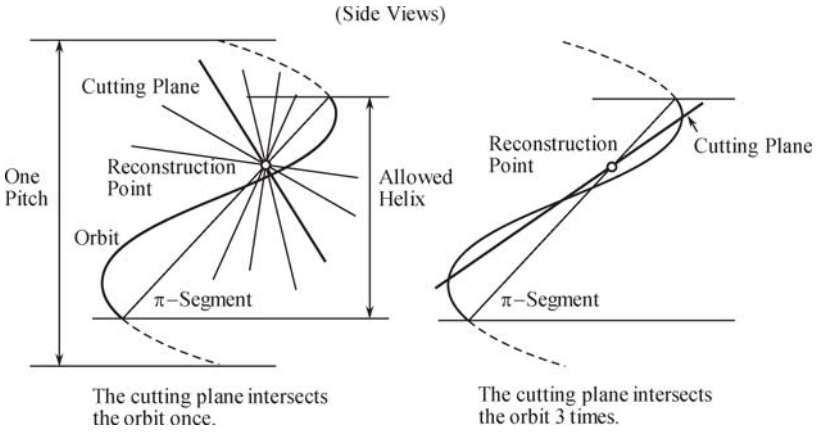


Fig. 5.17. The π-segment defines a section of helix. A cutting plane is a plane passing through the reconstruction point. The cutting plane either intersects the section of helix once or three times.

(1) Take the derivative of the cone-beam data with respect to the orbit parameter along the helix orbit.

(2) Perform the Hilbert transform along the directions that have been carefully selected.

(3) Perform a cone-beam backprojection with a weighting function, similar to the backprojection in Feldkamp's algorithm.

Some later versions of the algorithm have replaced the derivative with respect to the orbit parameters by partial derivative with respect to the detector coordinates.

*5.4 Mathematical Expressions

Some 3D image reconstruction algorithms are presented here without proofs. For the 3D parallel line-integral projections, we have a backprojection-then-filtering algorithm and filtered backprojection algorithms, which are not unique. For the parallel plane-integral projections (i.e., the Radon transform), we also have a backprojection-then-filtering algorithm and a filtered backprojection algorithm which is the Radon inversion formula.

For cone-beam projections, Feldkamp's circular orbit algorithm and the Katsevich's helical orbit algorithm are highlighted, because they are in the form of convolution and cone-beam backprojection. Tuy's relation and Grangeat's relation are also discussed in this section.

5.4.1 Backprojection-then-Filtering for Parallel Line-Integral Data

For this type of algorithm, the projections are backprojected to the image domain first, obtaining $b(x, y, z)$. Then, the 3D Fourier transform is applied on $b(x, y, z)$, obtaining $B(\omega_x, \omega_y, \omega_z)$. Next, a Fourier domain filter $G(\omega_x, \omega_y, \omega_z)$ is used to multiply $B(\omega_x, \omega_y, \omega_z)$, obtaining $F(\omega_x, \omega_y, \omega_z) = B(\omega_x, \omega_y, \omega_z)G(\omega_x, \omega_y, \omega_z)$. Finally, the 3D inverse Fourier transform is applied to $F(\omega_x, \omega_y, \omega_z)$ to obtain the reconstructed image $f(x, y, z)$. Here, the filter transfer function $G(\omega_x, \omega_y, \omega_z)$ is imaging geometry dependent. Some of the imaging geometries are shown in Figure 5.5, where the $\boldsymbol{\theta}$ trajectories are displayed as shaded regions. Let Ω denote the occupied region by the $\boldsymbol{\theta}$ trajectories on the unit sphere.

When $\Omega = \Omega_{4\pi}$ is 4π, that is, $\Omega_{4\pi}$ is the entire unit sphere, G is a ramp-filter:

$$G(\omega_x, \omega_y, \omega_z) = \frac{\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}}{\pi}. \tag{5.4.1}$$

If Ω is not the full sphere, this ramp filter will be modified by the geometry of Ω. Then the general form of G is

$$G(\omega_x, \omega_y, \omega_z) = \frac{\sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}}{D(\boldsymbol{\theta})}, \tag{5.4.2}$$

where $D(\boldsymbol{\theta})$ is half of the arc length of the intersection of a great circle with Ω. The normal direction of the great circle in the Fourier domain is $\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the direction from the origin to the point $(\omega_x, \omega_y, \omega_z)$.

When $\Omega = \Omega_\psi$ is the region shown in Figure 5.18, $D(\boldsymbol{\theta})$ is the arc length γ, which is orientation $\boldsymbol{\theta}$ dependent. Using the geometry, we have $\gamma = \pi$ if $\theta \leqslant \psi$, and we have $\sin\dfrac{\gamma}{2} = \dfrac{\sin\psi}{\sin\theta}$ if $\theta > \psi$.

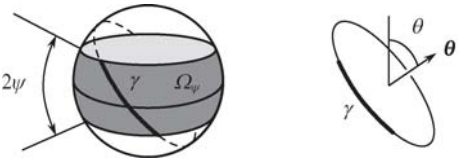


Fig. 5.18. The definition of the Ω_ψ and the arc length γ, which is a part of a great circle.

5.4.2 Filtered Backprojection Algorithm for Parallel Line-Integral Data

In a filtered backprojection algorithm, we need to find a 2D filter transfer function for each orientation $\boldsymbol{\theta} \in \Omega$. If $\Omega = \Omega_{4\pi}$, this filter is a ramp filter

$$Q(\omega_u, \omega_v) = \frac{\sqrt{\omega_u^2 + \omega_v^2}}{\pi}, \tag{5.4.3}$$

which is the same for all orientations $\boldsymbol{\theta} \in \Omega$.

If Ω is not the full sphere $\Omega_{4\pi}$, this ramp filter $Q(\omega_u, \omega_v)$becomes orientation $\boldsymbol{\theta}$ dependent as $Q_{\boldsymbol{\theta}}(\omega_u, \omega_v)$ (note: a subscript is added) and can be obtained by selecting the "central slice" of $G(\omega_x, \omega_y, \omega_z)$ with the normal direction $\boldsymbol{\theta}$. Here, the u-axis and v-axis are defined by unit vectors $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$, respectively. The three vectors, $\boldsymbol{\theta}, \boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ form an orthogonal system in 3D, where $\boldsymbol{\theta}$ represents the direction of a group of parallel lines perpendicular to a detector, $\boldsymbol{\theta}_u$ and $\boldsymbol{\theta}_v$ are on the detector plane, and $\boldsymbol{\theta}_u$ is also in the x-y plane of the global (x, y, z) system in 3D.

If we consider the case of $\Omega = \Omega_\psi$ shown in Figure 5.18, $Q_{\boldsymbol{\theta}}(\omega_u, \omega_v)$ has two expressions in two separate regions (see Figure 5.19):

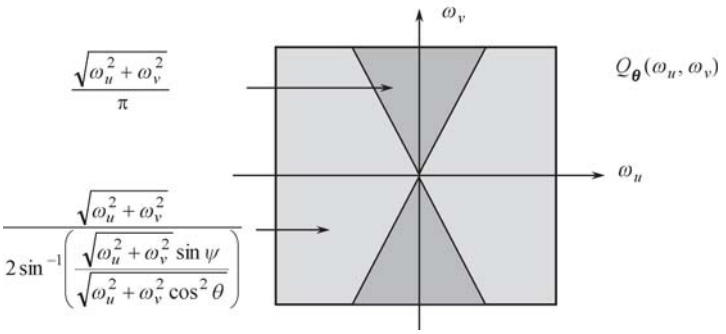


Fig. 5.19. The 2D filter transfer function Q for the imaging geometry Ω_ψ.

$$Q_{\boldsymbol{\theta}}(\omega_u, \omega_v) = \frac{\sqrt{\omega_u^2 + \omega_v^2}}{\pi} \tag{5.4.4}$$

if

$$0 \leqslant \sqrt{\omega_u^2 + \omega_v^2 \cos^2\theta} \leqslant \sqrt{\omega_u^2 + \omega_v^2}\sin\psi; \tag{5.4.5}$$

$$Q_{\boldsymbol{\theta}}(\omega_u, \omega_v) = \frac{\sqrt{\omega_u^2 + \omega_v^2}}{2\sin^{-1}\left(\dfrac{\sqrt{\omega_u^2 + \omega_v^2}\sin\psi}{\sqrt{\omega_u^2 + \omega_v^2\cos^2\theta}}\right)} \tag{5.4.6}$$

if

$$\sqrt{\omega_u^2+\omega_v^2}\sin\psi < \sqrt{\omega_u^2+\omega_v^2\cos^2\theta} \leqslant \sqrt{\omega_u^2+\omega_v^2}. \tag{5.4.7}$$

5.4.3 3D Radon Inversion Formula

The 3D Radon inversion formula (filtered backprojection algorithm) can only be applied to 3D plane integral data:

$$f(x,y,z) = \frac{-1}{8\pi^2}\iint\limits_{4\pi} \frac{\partial^2 p(s,\boldsymbol{\theta})}{\partial s^2}|_{s=\boldsymbol{x}\cdot\boldsymbol{\theta}} \sin\theta \mathrm{d}\theta\mathrm{d}\phi, \tag{5.4.8}$$

where $\boldsymbol{\theta} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ and $\boldsymbol{x} = (x,y,z)$. The coordinate systems are shown in Figure 5.20. If we backproject over a 2π solid angle, the factor $\dfrac{-1}{8\pi^2}$ should be replaced by $\dfrac{-1}{4\pi^2}$.

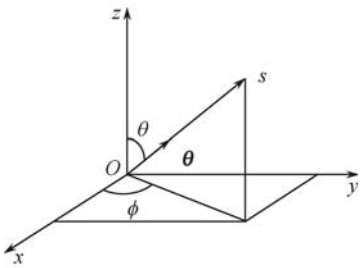


Fig. 5.20. The coordinate system for the 3D Radon inversion formula.

5.4.4 3D Backprojection-then-Filtering Algorithm for Radon Data

Let the backprojected image be

$$b(x,y,z) = \iint\limits_{2\pi} p(s,\boldsymbol{\theta})|_{s=\boldsymbol{x}\cdot\boldsymbol{\theta}} \sin\theta \mathrm{d}\theta\mathrm{d}\phi. \tag{5.4.9}$$

For the 3D Radon case, the Fourier transform of the projection/backprojection PSF is $\dfrac{1}{(\omega_x^2+\omega_y^2+\omega_z^2)}$. Therefore, the Fourier transform of the image $f(x,y,z)$ can be obtained as

$$F(\omega_x,\omega_y,\omega_z) = B(\omega_x,\omega_y,\omega_z) \times (\omega_x^2+\omega_y^2+\omega_z^2). \tag{5.4.10}$$

In the spatial domain, the backprojection-then-filtering algorithm for Radon data can be expressed as

$$f(x,y,z) = \Delta b(x,y,z) = \frac{\partial^2 b(x,y,z)}{\partial x^2} + \frac{\partial^2 b(x,y,z)}{\partial y^2} + \frac{\partial^2 b(x,y,z)}{\partial z^2} \quad (5.4.11)$$

where Δ is the Laplacian operator.

5.4.5 Feldkamp's Algorithm

First let us write down the fan-beam FBP reconstruction algorithm for the flat detector and express the image in polar coordinates (see Figure 5.21):

$$f(r,\varphi) = \frac{1}{2}\int_0^{2\pi} \frac{D}{D-s}\int_{-\infty}^{\infty} \frac{D}{\sqrt{D^2+l^2}} g(l,\beta)h(l'-l)\mathrm{d}l\mathrm{d}\beta, \quad (5.4.12)$$

where $h(l)$ is the convolution kernel of the ramp filter, D is the focal length, $g(l,\beta)$ is the fan-beam projection, l is the linear coordinate on the detector, $s = r\sin(\varphi-\beta)$, and $l' = \dfrac{Dr\cos(\varphi-\beta)}{D-r\sin(\varphi-\beta)}$. In this formula, $\dfrac{D}{\sqrt{D^2+l^2}}$ is the cosine of the incidence angle. When this algorithm is implemented, we first multiply the projections by this cosine function. Then we apply the ramp-filter to the pre-scaled data. Finally, we perform the fan-beam backprojection with a distance dependent weighting $D/(D-s)$, where s is the distance from the reconstruction point to the virtual detector, which is placed at the center of rotation for convenience.

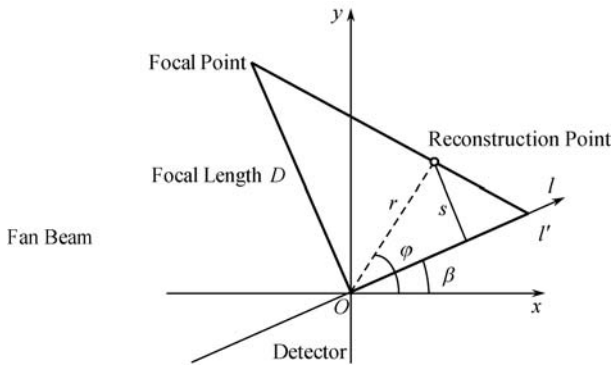


Fig. 5.21. The coordinate system for the flat-detector fan-beam imaging geometry.

Feldkamp's algorithm is almost the same as this fan-beam algorithm, except that the backprojection is a cone-beam backprojection. The ramp-

filtering is performed in the row-by-row fashion. There is no filtering performed in the axial direction. Let the axial direction be the z direction (see Figure 5.22); then

$$f(r,\varphi,z)=\frac{1}{2}\int_0^{2\pi}\frac{D}{D-s}\int_{-\infty}^{\infty}\frac{D}{\sqrt{D^2+l^2+\hat{z}^2}}g(l,\hat{z},\beta)h(l'-l)\mathrm{d}l\mathrm{d}\beta. \quad (5.4.13)$$

In this formula, $g(l,\hat{z},\beta)$ is the cone-beam projection, $D/\sqrt{D^2+l^2+\hat{z}^2}$ is the cosine of the incidence angle, and $\hat{z}$ and l' are defined in Figure 5.22.

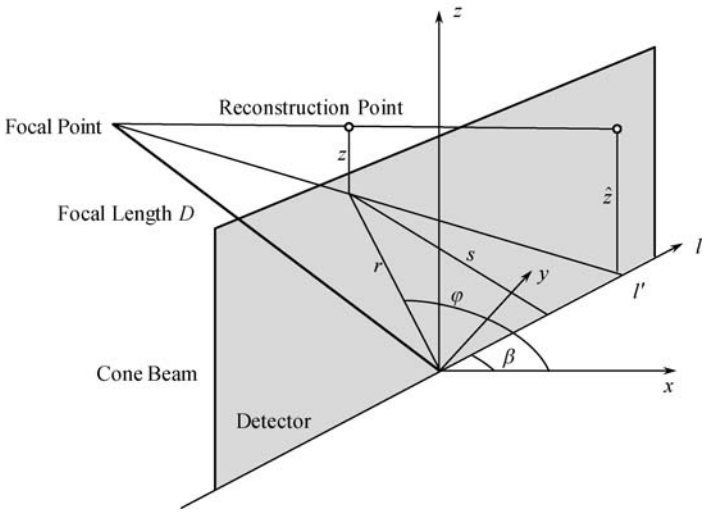


Fig. 5.22. The coordinate system for Feldkamp's cone-beam algorithm.

5.4.6 Tuy's Relationship

Tuy published a paper in 1983. In this paper, he established a relationship between the cone-beam data and the original image, which plays a similar role to the central slice theorem. Let us derive this relationship in the section.

The object to be imaged is f. Let the cone-beam focal point trajectory be denoted by a vector $\boldsymbol{\Phi}$ and let $\boldsymbol{\alpha}$ be a unit vector, indicating the direction of a projection ray. Therefore, the cone-beam data can be express by the following expression

$$g(\boldsymbol{\Phi},\boldsymbol{\alpha})=\int_0^{\infty}f(\boldsymbol{\Phi}+t\boldsymbol{\alpha})\mathrm{d}t,\quad \|\boldsymbol{\alpha}\|=1. \quad (5.4.14)$$

We now replace the unit vector $\boldsymbol{\alpha}$ by a general 3D vector $\boldsymbol{x}$, and the above

2D projection becomes an extended 3D function:

$$g(\boldsymbol{\Phi}, \boldsymbol{x}) = \int_0^\infty f(\boldsymbol{\Phi} + t\boldsymbol{x})\mathrm{d}t. \tag{5.4.15}$$

Taking the 3D Fourier transform of this function with respect to $\boldsymbol{x}$ and using notation $\boldsymbol{\beta}$ as the frequency domain variable, we have

$$\begin{aligned} G(\boldsymbol{\Phi}, \boldsymbol{\beta}) &= \int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty g(\boldsymbol{\Phi}, \boldsymbol{x}) \mathrm{e}^{-2\pi \mathrm{i}\, \boldsymbol{x}\cdot\boldsymbol{\beta}} \mathrm{d}\boldsymbol{x} \\ &= \int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \int_0^\infty f(\boldsymbol{\Phi} + t\boldsymbol{x}) \mathrm{e}^{-2\pi \mathrm{i}\, \boldsymbol{x}\cdot\boldsymbol{\beta}} \mathrm{d}t\mathrm{d}\boldsymbol{x}. \end{aligned} \tag{5.4.16}$$

Let $\boldsymbol{y} = \boldsymbol{\Phi} + t\boldsymbol{x}$; $\mathrm{d}\boldsymbol{y} = |t|^3\, \mathrm{d}\boldsymbol{x}$. The above expression becomes

$$G(\boldsymbol{\Phi}, \boldsymbol{\beta}) = \int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \int_0^\infty f(\boldsymbol{y}) \mathrm{e}^{-\frac{2\pi \mathrm{i}}{t}(\boldsymbol{y}-\boldsymbol{\Phi})\cdot\boldsymbol{\beta}} \frac{1}{|t|^3} \mathrm{d}t\mathrm{d}\boldsymbol{y}. \tag{5.4.17}$$

Let $s = \dfrac{1}{t}$; $\mathrm{d}s = -\dfrac{1}{t^2}\mathrm{d}t$. We then have

$$G(\boldsymbol{\Phi}, \boldsymbol{\beta}) = \int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty \int_0^\infty f(\boldsymbol{y}) \mathrm{e}^{-2\pi \mathrm{i}\, s(\boldsymbol{y}-\boldsymbol{\Phi})\cdot\boldsymbol{\beta}} |s|^3 \frac{1}{s^2} \mathrm{d}\boldsymbol{y}\mathrm{d}s, \tag{5.4.18}$$

$$G(\boldsymbol{\Phi}, \boldsymbol{\beta}) = \int_0^\infty \left(\int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty f(\boldsymbol{y}) \mathrm{e}^{-2\pi \mathrm{i}\, \boldsymbol{y}\cdot(s\boldsymbol{\beta})} \mathrm{d}\boldsymbol{y} \right) |s|\, \mathrm{e}^{2\pi \mathrm{i}\, s(\boldsymbol{\Phi}\cdot\boldsymbol{\beta})} \mathrm{d}s. \tag{5.4.19}$$

Recognizing that the inner triple integral is the 3D Fourier transform of f, that is

$$F(s\boldsymbol{\beta}) = \int_{-\infty}^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty f(\boldsymbol{y}) \mathrm{e}^{-2\pi \mathrm{i}\, \boldsymbol{y}\cdot(s\boldsymbol{\beta})} \mathrm{d}\boldsymbol{y}, \tag{5.4.20}$$

we have

$$G(\boldsymbol{\Phi}, \boldsymbol{\beta}) = \int_0^\infty F(s\boldsymbol{\beta})\, |s|\, \mathrm{e}^{2\pi \mathrm{i}\, s(\boldsymbol{\Phi}\cdot\boldsymbol{\beta})} \mathrm{d}s. \tag{5.4.21}$$

Next, we change the limit from $[0, \infty)$ to $(-\infty, \infty)$ and obtain

$$G(\boldsymbol{\Phi}, \boldsymbol{\beta}) = \int_{-\infty}^\infty F(s\boldsymbol{\beta})\, |s|\, \mathrm{e}^{2\pi \mathrm{i}\, s(\boldsymbol{\Phi}\cdot\boldsymbol{\beta})} \mathrm{d}s + \int_{-\infty}^\infty F(s\boldsymbol{\beta}) \frac{\mathrm{i}\, s}{\mathrm{i}} \mathrm{e}^{2\pi \mathrm{i}\, s(\boldsymbol{\Phi}\cdot\boldsymbol{\beta})} \mathrm{d}s. \tag{5.4.22}$$

Using 3D Radon transform's central slice theorem $F(s\boldsymbol{\beta})$ is the Fourier transform of the plane integral of the original image f in the direction $\boldsymbol{\beta}$. The factor $|s|$ is the ramp-filter in the Fourier domain, and the factor $(\mathrm{i}\, s)$ corresponds

to the derivative in the spatial domain. Each of the two terms at the right-hand-side of Eq. (5.4.22) is in the form of an inverse Fourier transform, which is, in fact, a convolution. Therefore,

$$G(\boldsymbol{\Phi}, \boldsymbol{\beta}) = p_{\boldsymbol{\beta}}(t) * h(t) - \mathrm{i}\, p_{\boldsymbol{\beta}}(t) * \delta'(t) \tag{5.4.23}$$

where $\mathrm{i} = \sqrt{-1}$, $t = \boldsymbol{\Phi} \cdot \boldsymbol{\beta}$, $h(t)$ is the ramp-filter convolution kernel

$$h(t) = \int_{-\infty}^{\infty} |s| \mathrm{e}^{2\pi \mathrm{i}\, st} \mathrm{d}s, \tag{5.4.24}$$

and the Radon transform of the original image f is defined as

$$p_{\boldsymbol{\beta}}(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\boldsymbol{x}) \delta(\boldsymbol{x} \cdot \boldsymbol{\beta} - t) \mathrm{d}\boldsymbol{x}. \tag{5.4.25}$$

The left-hand-side of Eq. (5.2.23) is related to the cone-beam projections, and the right-hand-side is related to the plane integral of the original image.

Tuy's algorithm is stated as

$$f(\boldsymbol{x}) = \frac{1}{2\pi \mathrm{i}} \iint\limits_{4\pi} \frac{1}{\left| \boldsymbol{\Phi}'(\lambda) \cdot \boldsymbol{\beta} \right|} \frac{\partial G(\boldsymbol{\Phi}(\lambda), \boldsymbol{\beta})}{\partial \lambda} \mathrm{d}\boldsymbol{\beta} \tag{5.4.26}$$

with $\boldsymbol{\Phi}(\lambda) \cdot \boldsymbol{\beta} = \boldsymbol{x} \cdot \boldsymbol{\beta}$. In fact, the factor $(2\pi \mathrm{i}\,)$ in Tuy's algorithm does not make the reconstruction $f(\boldsymbol{x})$ imaginary. Because the combination of the first term in Eq. (5.2.23) above and the factor $\left| \boldsymbol{\Phi}'(\lambda) \cdot \boldsymbol{\beta} \right|$ is odd in $\boldsymbol{\beta}$, the real part in $\iint\limits_{4\pi} \frac{1}{\left| \boldsymbol{\Phi}'(\lambda) \cdot \boldsymbol{\beta} \right|} \frac{\partial G(\boldsymbol{\Phi}(\lambda), \boldsymbol{\beta})}{\partial \lambda} d\boldsymbol{\beta}$ will vanish. Thus, Tuy's algorithm will reconstruct a real image.

5.4.7 Grangeat's Relationship

Grangeat established a relationship between the derivative of Radon data and the derivative of the line-integrals of the data on the flat cone-beam detection plane.

In the following, we only consider a fixed focal point position $\boldsymbol{\Phi}$. We arbitrarily select a straight line on the detector and sum up the cone-beam projections along this line. Let us set up a coordinate system on the detector plane (see Figure 5.23). The u-axis is along the integral line on the detector and the v-axis is orthogonal to the u-axis. We denote the cone-beam projection data on the u-v coordinate system as $g(u, v)$. If we use the object f with the spherical system, then the projection $g(u, v)$ can be expressed as

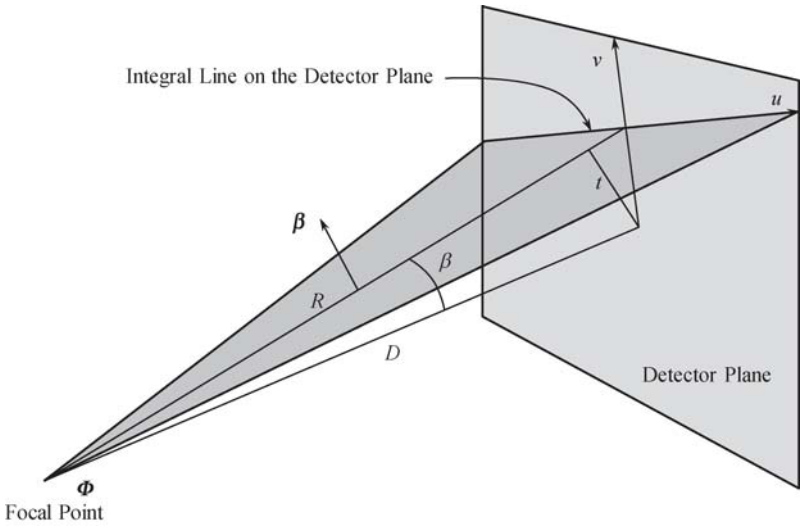


Fig. 5.23. A line is drawn on the detector. The cone-beam data are summed on this line as $s(v)$.

(see Figure 5.24)

$$g(u,v) = \int_{-\infty}^{\infty} f(\boldsymbol{\beta}, \theta, r)\mathrm{d}r, \tag{5.4.27}$$

where u and θ are related as

$$u = \sqrt{D^2 + v^2}\tan\theta. \tag{5.4.28}$$

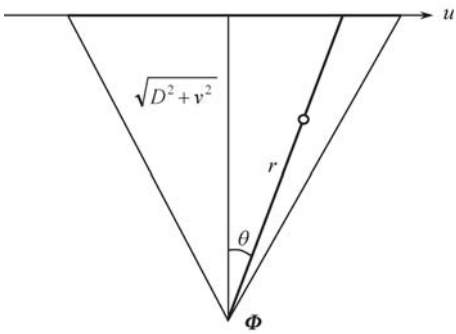


Fig. 5.24. The forming of cone-beam projection.

As in Feldkamp's algorithm, the cone-beam data are pre-scaled by a cosine function $D/\sqrt{D^2+u^2+v^2}$. Thus, the data summation along the u-axis is

actually

$$s(v) = \int_{-\infty}^{\infty} g(u,v)\frac{D}{\sqrt{D^2+u^2+v^2}}\mathrm{d}u$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\boldsymbol{\beta},\theta,r)\frac{D}{\sqrt{D^2+u^2+v^2}}\mathrm{d}r\mathrm{d}u. \tag{5.4.29}$$

We now change the variable u to variable θ, using $u = \sqrt{D^2+v^2}\tan\theta$, $\cos\theta = \dfrac{\sqrt{D^2+v^2}}{\sqrt{D^2+u^2+v^2}}$, and $\dfrac{\mathrm{d}u}{\mathrm{d}\theta} = \dfrac{\sqrt{D^2+v^2}}{\cos^2\theta}$, and we obtain

$$s(v) = \int_{-\infty}^{\infty}\int_{-\pi/2}^{\pi/2} f(\boldsymbol{\beta},\theta,r)\frac{D}{\sqrt{D^2+u^2+v^2}}\frac{\sqrt{D^2+v^2}}{\cos^2\theta}\mathrm{d}r\mathrm{d}\theta$$
$$= \int_{-\infty}^{\infty}\int_{-\pi/2}^{\pi/2} f(\boldsymbol{\beta},\theta,r)\frac{D}{\cos\theta}\mathrm{d}r\mathrm{d}\theta. \tag{5.4.30}$$

We would like to relate this $s(v)$ to Radon transform of f:

$$p_{\boldsymbol{\beta}}(t) = \int_{-\infty}^{\infty}\int_{-\pi/2}^{\pi/2} f(\boldsymbol{\beta},\theta,r)r\mathrm{d}r\mathrm{d}\theta, \tag{5.4.31}$$

where the parameter t is defined in Figure 5.23. We will now make this connection with the idea presented in Section 5.3.2, from which we know that (see Figure 5.25)

$$\frac{\partial f}{\partial \beta} = R\frac{\partial f}{\partial t} = r\cos\theta\frac{\partial f}{\partial t}. \tag{5.4.32}$$

Using this relationship, we have

$$\frac{\partial p_{\boldsymbol{\beta}}(t)}{\partial t} = \int_{-\infty}^{\infty}\int_{-\pi/2}^{\pi/2}\frac{\partial}{\partial t}f(\boldsymbol{\beta},\theta,r)r\mathrm{d}r\mathrm{d}\theta$$
$$= \int_{-\infty}^{\infty}\int_{-\pi/2}^{\pi/2}\frac{1}{r\cos\theta}\frac{\partial}{\partial\beta}f(\boldsymbol{\beta},\theta,r)r\mathrm{d}r\mathrm{d}\theta$$
$$= \frac{1}{D}\int_{-\infty}^{\infty}\int_{-\pi/2}^{\pi/2}\frac{D}{\cos\theta}\frac{\partial}{\partial\beta}f(\boldsymbol{\beta},\theta,r)\mathrm{d}r\mathrm{d}\theta$$
$$= \frac{1}{D}\frac{\partial s(v)}{\partial\beta}. \tag{5.4.33}$$

Finally, we will replace the partial derivative with respect to β with the partial derivative with respect to v. From Figure 5.25, we see that

$$v = D\tan\beta \text{ and } \frac{\mathrm{d}v}{\mathrm{d}\beta} = \frac{D}{\cos^2\beta}. \tag{5.4.34}$$

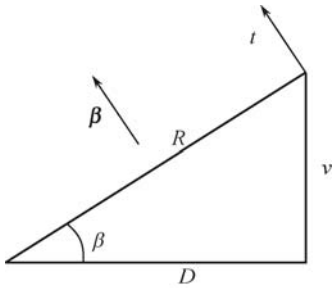


Fig. 5.25. The direction of t is the tangent direction.

The Grangeat's relationship is obtained as

$$\begin{aligned}\frac{\partial p_{\boldsymbol{\beta}}(t)}{\partial t} &= \frac{1}{D}\frac{\partial s(v)}{\partial \beta} \\ &= \frac{1}{D}\frac{\partial s(v)}{\partial v}\frac{\partial v}{\partial \beta} \\ &= \frac{1}{D}\frac{\partial s(v)}{\partial v}\frac{D}{\cos^2\beta} \\ &= \frac{s'(v)}{\cos^2\beta}. \end{aligned} \tag{5.4.35}$$

5.4.8 Katsevich's Algorithm

We denote the helix focal-point by a vector

$$\boldsymbol{a}(s) = \left(R\cos s, R\sin s, \frac{h}{2\pi}s\right), \quad s \in I_\pi(\boldsymbol{x}) \tag{5.4.36}$$

where R is the radius, h is the helix pitch, $\boldsymbol{x} = (x, y, z)$ is the reconstruction point, s is the orbit parameter, and $I_\pi(\boldsymbol{x}) = [s_b, s_t]$, which is determined by the π-segment of the point $\boldsymbol{x}$.

The cone-beam projection is represented in a local coordinate system as $g(\boldsymbol{\theta}, \boldsymbol{a})$, where $\boldsymbol{\theta}$ is a function of the local coordinate system $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ (see Figure 5.26) with a parameter γ and is defined as

$$\boldsymbol{\theta}(\gamma) = (\cos\gamma)\boldsymbol{\alpha} + (\sin\gamma)\boldsymbol{\beta}, \quad -\frac{\pi}{2} < \gamma < \frac{\pi}{2}. \tag{5.4.37}$$

The unit vector $\boldsymbol{\alpha}$ is defined as the direction from the focal point to the reconstruction point. The unit vector $\boldsymbol{\beta}$ is the filtering direction on the detector plane (see Figure 5.27). Generally speaking, filtering can be any kind of processing on the data. In Katsevich's formula, when we say "filtering" we

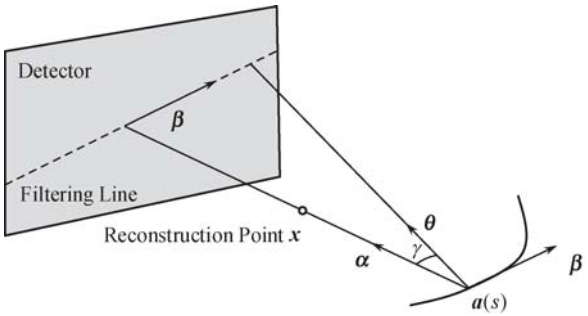


Fig. 5.26. The coordinate system for Katsevich's helical orbit cone-beam algorithm.

specifically mean the Hilbert transform. Vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are orthogonal to each other. Katsevich's formula is given as follows:

$$f(\boldsymbol{x}) = \frac{-1}{2\pi^2} \int_{I_\pi(\boldsymbol{x})} \frac{1}{\|\boldsymbol{x} - \boldsymbol{a}(s)\|} \int_{-\pi/2}^{\pi/2} \frac{\partial g(\boldsymbol{\theta}(\gamma), \boldsymbol{a}(q))}{\partial q}\bigg|_{q=s} \frac{1}{\sin\gamma} \mathrm{d}\gamma \mathrm{d}s, \tag{5.4.38}$$

where the integral over γ is the Hilbert transform and the integral over s is the cone-beam backprojection. Katsevich derived this nice and clean formula because he thought of a trick to assign the weights for redundant plane-integrals by selecting a special filtering direction $\boldsymbol{\beta}$. It can be shown that his assignment of $\boldsymbol{\beta}$ makes the weight $+1$ for a plane that is measured once and makes the weight $+1$ or -1 for a plane that is measured three times (two of them get a $+1$ and one of them gets a -1). The solution of $\boldsymbol{\beta}$ is not unique. Different selections of the filtering direction give different algorithms.

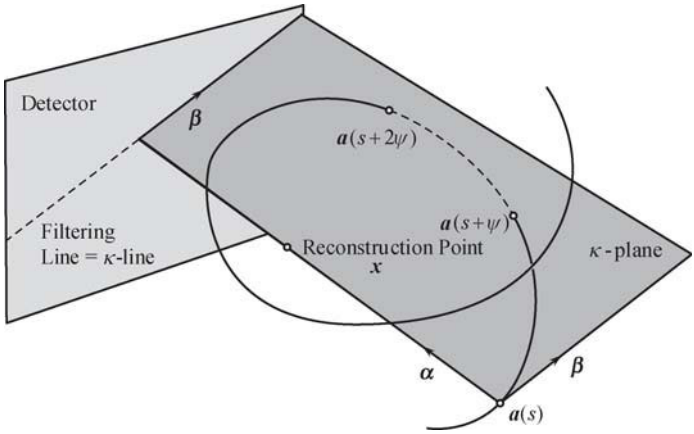


Fig. 5.27. The κ-plane contains the reconstruction point and 3 uniformly spaced points on the helix within the section governed by the π-segment.

Here is one way to find a special filtering direction $\boldsymbol{\beta}$. Let us define a κ-plane as follows. The focal point location is $\boldsymbol{a}(s)$. A point in the field-of-view $\boldsymbol{x}$ is to be reconstructed. We then find an angle ψ in $(-\pi, \pi)$ such that the four points $\boldsymbol{x}$, $\boldsymbol{a}(s)$, $\boldsymbol{a}(s+\psi)$, and $\boldsymbol{a}(s+2\psi)$ are in one plane (see Figure 5.27). This plane exists but is not unique. The angle ψ with the smallest magnitude $|\psi|$ that can construct this plane will be chosen, and the corresponding plane that contains these four points is referred to as a κ-plane $\kappa(s, \psi)$.

The intersection of a κ-plane $\kappa(s, \psi)$ with the detector is called a κ-line. The filtering direction $\boldsymbol{\beta}$ is along the κ-line. If you let the point $\boldsymbol{x}$ vary, you will get a bunch of these κ-planes $\kappa(s, \psi)$, and you will get a bunch of κ-lines on one detector (see Figure 5.28).

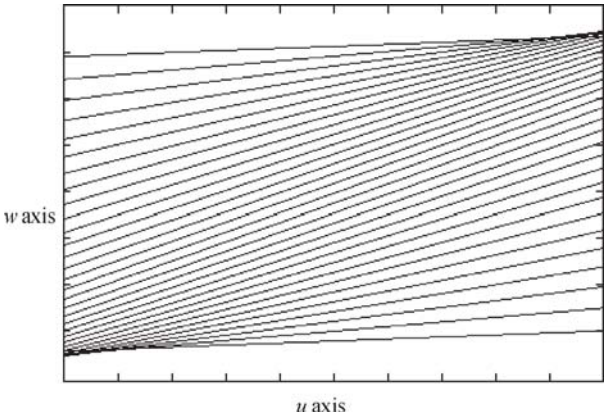


Fig. 5.28. The κ-lines on the flat cone-beam detector. Each line corresponds to a fixed angle ψ. Each reconstruction point associates with one and only one κ-line.

On a flat detector, the κ-lines are straight lines. Let us assign a u-w coordinate system to the detector plane as indicated in Figure 5.28 with the w axis being the helix axis (i.e., the z-axis). The κ-lines can be described by the u-w relation for a fixed ψ:

$$w = \frac{Dh}{2\pi R}\left(\psi + \frac{\psi}{\tan\psi}\frac{u}{D}\right), \tag{5.4.39}$$

where R is the radius of the helix, h is the pitch of the helix, and D is the distance between the detector plane and the focal point. There are many line depicted in Figure 5.28, and each line corresponds to a fixed ψ value.

If a curved detector is used, the κ-lines are no longer straight lines on the detector but are curves. For a fixed ψ, the α-w relationship is given as

$$w = \frac{Dh}{2\pi R}\left(\psi\cos\alpha + \frac{\psi}{\tan\psi}\sin\alpha\right), \tag{5.4.40}$$

where the angle α is defined in Figure 5.29. We can plot the curved κ-lines

using this relationship for a set of ψ values to obtain a counter part of Figure 5.28 for the curved detector (see Figure 5.30).

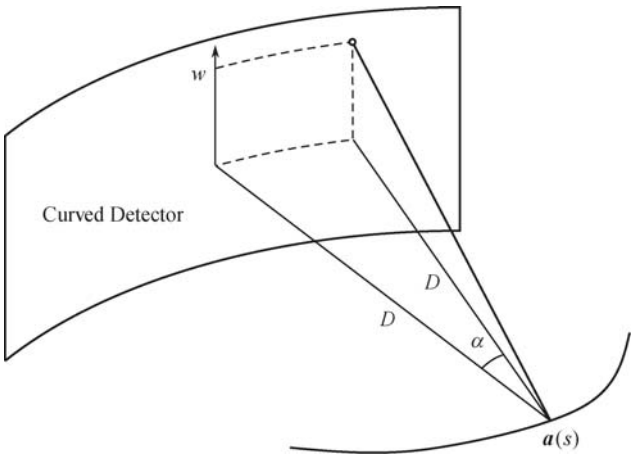


Fig. 5.29. The coordinate system for a curved-detector cone-beam geometry.

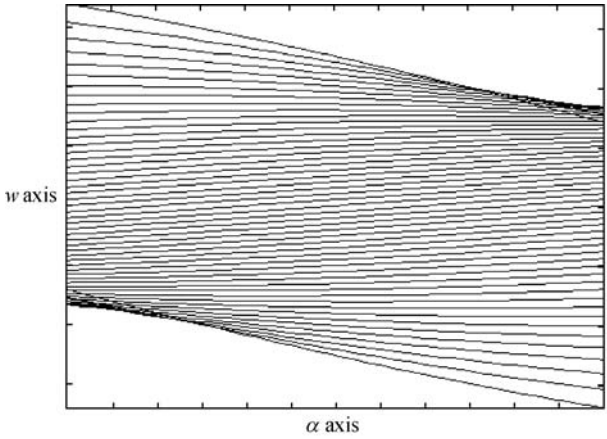


Fig. 5.30. The κ-curves on the curved cone-beam detector. Each curve corresponds to a fixed angle ψ. Each reconstruction point associates with one and only one κ-curve.

Finally, we will explain a little more on the curved-detector implementation of Katsevich's algorithm, which is given again below:

$$f(\boldsymbol{x}) = \frac{-1}{2\pi^2} \int_{I_\pi(\boldsymbol{x})} \frac{1}{\|\boldsymbol{x} - \boldsymbol{a}(s)\|} \int_{-\pi/2}^{\pi/2} \frac{\partial g(\boldsymbol{\theta}(\gamma), \boldsymbol{a}(q))}{\partial q}\bigg|_{q=s} \frac{1}{\sin\gamma} \mathrm{d}\gamma \mathrm{d}s. \tag{5.4.41}$$

Step 1: We take the derivative at a constant direction $\boldsymbol{\theta}$ with respect to the orbit parameter s, that is, to evaluate $\dfrac{\partial g(\boldsymbol{\theta}, \boldsymbol{a}(s))}{\partial s}$.

In practice, the projection data are sampled at discrete focal-point locations with a discrete detector. The derivative will be implemented as a finite difference using pairs of consecutive projections at focal point locations $\boldsymbol{a}(s_k)$ and $\boldsymbol{a}(s_{k+1})$. When evaluating the difference, the pair of projection rays $g(\boldsymbol{\theta}, \boldsymbol{a}(s_k))$ and $g(\boldsymbol{\theta}, \boldsymbol{a}(s_{k+1}))$ must have the same (global) direction $\boldsymbol{\theta}$ and the same w coordinate on the detector, as illustrated in Figure 5.31.

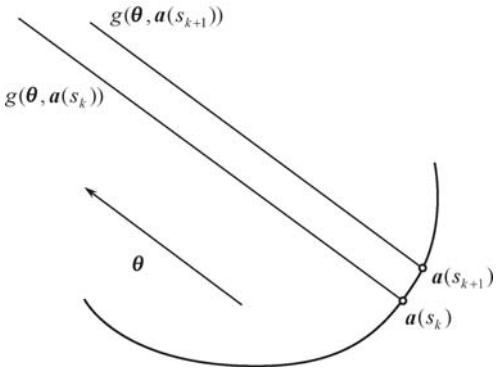


Fig. 5.31. Two neighboring detector views are required to implement the derivative with respect to the orbit parameter s as the difference. When taking the finite difference, the two rays must be parallel and have the same w coordinate.

In the detector α-w coordinates, $g(\boldsymbol{\theta}, \boldsymbol{a}(s_k))$ and $g(\boldsymbol{\theta}, \boldsymbol{a}(s_{k+1}))$ will have different α values. If you let us abuse the notation a little and use the detector coordinate system, we let

$$g\left(\alpha - \frac{\Delta s}{2}, w, \boldsymbol{a}(s_k)\right) = g(\boldsymbol{\theta}, \boldsymbol{a}(s_k)) \tag{5.4.42}$$

and

$$g\left(\alpha + \frac{\Delta s}{2}, w, \boldsymbol{a}(s_{k+1})\right) = g(\boldsymbol{\theta}, \boldsymbol{a}(s_{k+1})), \tag{5.4.43}$$

then we approximate the derivative as

$$\begin{aligned} &\frac{\partial g\left(\alpha, w, \boldsymbol{a}\left(s_{k+\frac{1}{2}}\right)\right)}{\partial s} \\ &\approx \frac{g\left(\alpha + \frac{\Delta s}{2}, w, \boldsymbol{a}\left(s_{k+\frac{1}{2}}\right)\right) - g\left(\alpha - \frac{\Delta s}{2}, w, \boldsymbol{a}\left(s_k\right)\right)}{\Delta s}, \end{aligned} \tag{5.4.44}$$

where $\Delta s = s_{k+1} - s_k$.

Step 2: The result of Step 1 is weighed by the cosine function $\frac{D}{\sqrt{D^2+w^2}}$, obtaining

$$\frac{D}{\sqrt{D^2+w^2}} \times \frac{\partial g\left(\alpha, w, \boldsymbol{a}\left(s_{k+\frac{1}{2}}\right)\right)}{\partial s}. \tag{5.4.45}$$

This scaling is also called the length-correction weighting. This is a point-by-point scaling on the detector.

Step 3: To prepare for Hilbert transform operation, we rebin the κ-curves to horizontal lines, mapping the α-w detector representation to α-ψ representation (see Figure 5.32). In other words, we move each data point on the α-w detector up or down according to $w = \frac{Dh}{2\pi R}\left(\psi\cos\alpha + \frac{\psi}{\tan\psi}\sin\alpha\right)$ so that the κ-curves become horizontal lines.

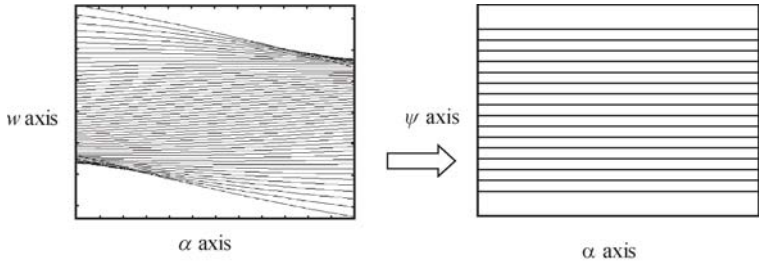


Fig. 5.32. Rebin the κ-curves into horizontally parallel lines.

Step 4: After rebinning, the regular line-by-line 1D Hilbert transform is performed.

Step 5: This step is the inverse of Step 3. It rebins the filtered (i.e., Hilbert transformed) data back to the original detector α-w coordinates with curved κ-lines (see Figure 5.33).

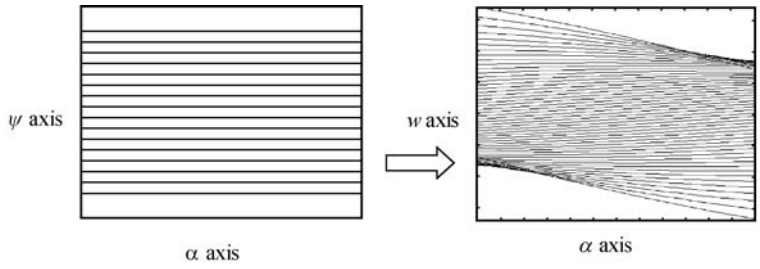


Fig. 5.33. Rebin the parallel lines back to κ-curves.

Step 6: We weigh the data on the curved detector plane by a cosine function $\cos\alpha$, for all different α locations. The angle α is defined in Figure 5.29.

Step 7: This is the final step, and it performs the ray-by-ray 3D cone-beam backprojection with the weighting factor $\frac{1}{\|\boldsymbol{x}-\boldsymbol{a}(s)\|}$ using a helix orbit. This step is similar to the backprojection in the Feldkamp algorithm, where a circular orbit is used.

5.5 Worked Examples

Example 1 A SPECT camera is mounted with a parallel-beam collimator which has a 30° tilt angle as shown in Figure 5.34. The camera rotates around the patient to collect projections. Does this imaging geometry satisfy Orlov's condition?

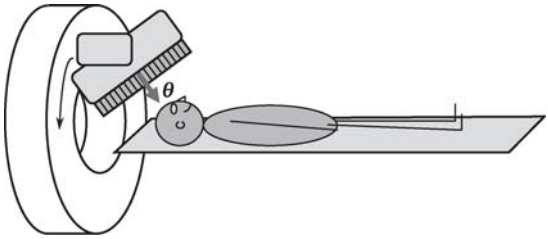


Fig. 5.34. The SPECT detector is tilted.

Solution

No. If we draw the trajectory of the projection direction $\boldsymbol{\theta}$ on a unit sphere, we see that the trajectory is a small circle (see Figure 5.35). This does not satisfy Orlov's condition.

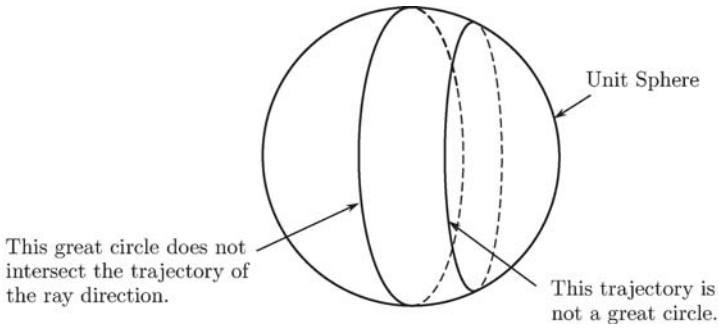


Fig. 5.35. Orlov's condition is not met.

Example 2 State the central slice theorem with a mathematical expression for the 3D line-integral projections and for the 3D plane-integral projections.

Solution

(1) 3D plane integral data

Let $\boldsymbol{\theta} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$; then the central slice theorem for the 3D Radon transform (i.e., plane-integral projections) is

$$P(\omega, \boldsymbol{\theta}) = F(\omega\sin\theta\cos\phi, \omega\sin\theta\sin\phi, \omega\cos\theta), \tag{5.5.1}$$

where $P(\omega, \boldsymbol{\theta})$ is the 1D Fourier transform of the plane integrals $p(s, \boldsymbol{\theta})$ with respect to s, and $F(\omega_x, \omega_y, \omega_z)$ is the 3D Fourier transform of the object $f(x, y, z)$.

(2) 3D line integral data

For the 3D line-integral projections, we need a coordinate system on the detector plane which is perpendicular to $\boldsymbol{\theta}$. Let

$$\boldsymbol{\theta}_u = (-\sin\phi, \cos\phi, 0) \text{ and } \boldsymbol{\theta}_v = (-\cos\theta\cos\phi, -\cos\theta\sin\phi, \sin\theta), \tag{5.5.2}$$

then $\boldsymbol{\theta}$, $\boldsymbol{\theta}_u$, and $\boldsymbol{\theta}_v$ form an orthogonal system, $\boldsymbol{\theta}_u$ is in the direction of the u-axis, and $\boldsymbol{\theta}_v$ is in the direction of the v-axis. Thus, the central slice theorem can be stated as

$$\begin{aligned} & P(\omega_u, \omega_v, \boldsymbol{\theta}) \\ &= F(-\omega_u\sin\phi - \omega_v\cos\theta\cos\phi, \omega_u\cos\phi - \omega_v\cos\theta\sin\phi, \omega_v\sin\theta) \\ &= F(\omega_u\boldsymbol{\theta}_u + \omega_v\boldsymbol{\theta}_v), \end{aligned} \tag{5.5.3}$$

where $P(\omega_u, \omega_v, \boldsymbol{\theta})$ is the 2D Fourier transform of the line integrals $p(u, v, \boldsymbol{\theta})$ with respect to u and v.

Example 3 Run Feldkamp's algorithm and observe the reconstruction artifacts by varying the cone-angle.

Solution

We programmed Feldkamp's algorithm and did a set of computer simulations using a circular focal-point orbit and a Defrise phantom, which consists of five flat uniform ellipsoids (see Figure 5.36). Six different cone angles (2°, 4°, 8°, 16°, 32°, and 64°) were used in the simulations. The central sagittal view for each simulation is displayed in Figure 5.37. It is observed that for large cone angles, Feldkamp's algorithm introduces severe artifacts, especially in the regions away from the central plane (i.e., the orbit plane). The images are quite good for small cone angles.

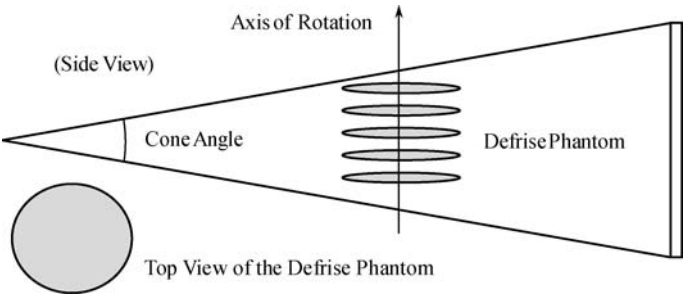


Fig. 5.36. A Defrise phantom is used in computer simulations.

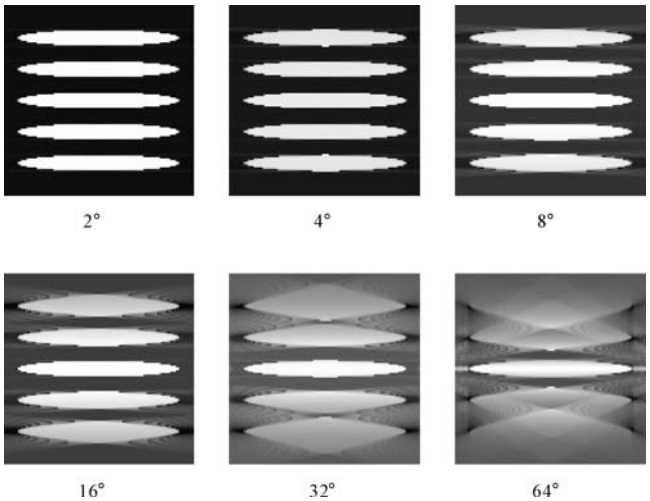


Fig. 5.37. Computer simulation results with Feldkamp's algorithm using different cone angles.

5.6 Summary

- In 3D, the parallel line integrals are referred to as the ray transform, and the parallel plane integrals are referred to as the Radon transform.
- For the Radon projection data, we require that all directions in a 2π solid angle should be measured.
- For the Radon projection data, the image reconstruction algorithm (i.e., the Radon inversion formula) is very simple — a second order derivative followed by Radon backprojection. Of course, one can switch the order of

derivative and backprojection. If the backprojection is performed first, it should be followed by a Laplacian operator, which is a summation of the second order partial derivative with respect to x, y, and z, respectively.

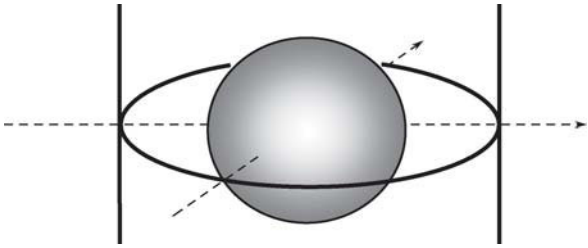
- An efficient way to perform Radon backprojection is to do it in two steps and each step is a series of 2D backprojections.
- For the ray transform data, we require that Orlov's data sufficiency condition be satisfied. The ray directions trace a trajectory on the unit sphere. If every great circle intersects this trajectory, Orlov's condition is met.
- The image reconstruction algorithm for the ray transform depends on the ray direction trajectory. Due to data redundancy, the reconstruction algorithm is not unique. One can either do filtering first or backprojection first.
- Feldkamp et al. developed a simple and robust FBP algorithm for cone-beam circular orbit imaging. Even though this algorithm is a modification of a fan-beam's FBP algorithm and is not exact, it has wide applications in many fields. The reconstruction errors are not significant if the cone-angle is small enough.
- One can use Tuy's condition to verify if the cone-beam imaging geometry is able to provide sufficient projections. Non-planar orbits, such as the helix orbit or the circle-and-line orbit, are required to satisfy Tuy's condition. Tuy developed a relationship between the cone-beam data and the original image; he also developed a cone-beam inversion formula, but it is difficult to use.
- Grangeat's relationship is that the angular derivative of the cone-beam weighted planar integral equals to the derivative of the Radon planar integral. In Grangeat's cone-beam image reconstruction algorithm, the image is reconstructed using the Radon inversion formula. A drawback of Grangeat's cone-beam reconstruction method is the rebinning from the cone-beam data to Radon data. The rebinning step can cause large errors.
- Katsevich's cone-beam image reconstruction algorithm is truly an FBP algorithm with shift-invariant filtering and cone-beam backprojection. One drawback of Katsevich's algorithm is its difficulty in selection of filtering directions. Another drawback is that cone-beam projection data are not fully used.
- The readers are expected to understand the Radon inversion formula and Feldkamp's cone-beam image reconstruction algorithm in this chapter.

Problems

Problem 5.1 Calculate the 3D parallel line-integrals $p(u, v, \boldsymbol{\theta})$ and parallel plane-integrals $p(s, \boldsymbol{\theta})$ of a uniform ball, in which the line density and

area density are both 1. The center of the ball is at the origin of the coordinate system, and the radius of the uniform ball is R.

Problem 5.2 A cone-beam focal point orbit is a circle with two lines as shown. The radius of the circular orbit is R. The object to be imaged is a ball of radius r. Determine the length of the linear orbits so that Tuy's condition can be satisfied.



Problem 5.3 Prove that Feldkamp's algorithm can give an exact reconstruction for the object $f(x, y, z)$ that is constant in the axial direction (i.e., the z direction). In other words, for any given point (x_0, y_0), the function $f(x_0, y_0, z)$ does not vary with variable z.

References

1. Axelsson C, Danielsson PE (1994) Three-dimensional reconstruction from cone-beam data in $O(N^3 \log N)$ time. Phys Med Biol 39: 477 – 491
2. Barrett HH, Swindell W (1981) Radiological Imaging, Academic, New York
3. Chen GH (2003) An alternative derivation of Katsevich's cone-beam reconstruction formula. Med Phys 30: 3217 – 3226
4. Clack R (1992) Towards a complete description of three-dimensional filtered backprojection. Phys Med Biol 37: 645 – 660
5. Clack R, Defrise M (1994) Overview of reconstruction algorithms for exact cone-beam tomography. Proc SPIE 2299: 230 – 241
6. Colsher JG (1980) Fully three-dimensional positron emission tomography. Phys Med Biol 25: 103 – 115
7. Crawford CR (1991) CT filtration aliasing artifacts. IEEE Trans Med Imaging 10: 99 – 102
8. Deans SR (1983) The Radon Transform and Some of Its Applications, John Wiley, New York
9. Defrise M, Clack R (1994) A cone-beam reconstruction algorithm using shift-variant filtering and cone-beam backprojection. IEEE Trans Med Imaging 13: 186 – 195
10. Defrise M (1995) A factorization method for the 3D X-ray transform. Inverse Probl 11: 983 – 994

11. Defrise M, Clack R, Townsend D (1993) Solution to the three-dimensional image reconstruction problem from two-dimensional projections. J Opt Soc A 10: 869 – 877
12. Defrise M, Clack R, Townsend D (1995) Image reconstruction from truncated, two-dimensional parallel projections. Inverse Probl 11: 287 – 313
13. Defrise M, Kinahan PE, Townsend DW et al (1997) Exact and approximate rebinning algorithms for 3-D PET data. IEEE Trans Med Imaging 16: 145 – 158
14. Feldkamp LA, Davis LC, Kress JW (1984) Practical cone beam algorithm. J Opt Soc Am A 1: 612 – 619
15. Grangeat P (1991) Mathematical framework of cone beam 3D reconstruction via the first derivative of the Radon transform. In: Herman G, Luis AK, Natterer F (eds) Mathematical Methods in Tomography, Lecture Notes in Mathematics 1497: 66 – 97
16. Katsevich A (2002) Theoretically exact filtered backprojection-type inversion algorithm for spiral CT. SIAM J Appl Math 62: 2012 – 2026
17. Katsevich A (2002) Analysis of an exact inversion algorithm for spiral cone-beam CT. Phys Med Biol 47: 2583 – 2597
18. Katsevich A (2003) An improved exact filtered backprojection algorithm for spiral computed tomography. Adv Appl Math 32: 681 – 697
19. Kinahan PE, Rogers JG, Harrop R et al (1988) Three-dimensional image reconstruction in object space. IEEE Trans Nucl Sci 35: 635 – 640
20. Kudo H and Saito T (1994) Derivation and implementation of a cone-beam reconstruction algorithm for non-planar orbit. IEEE Trans Med Imaging 13: 196 – 211
21. Natterer F (1986) The mathematics of computerized tomography, John Wiley, New York
22. Noo F, Pack J, Heuscher D (2003) Exact helical reconstruction using native cone-beam geometries. Phys Med Biol 48: 3787 – 3818
23. Orlov SS (1976) Theory of three-dimensional image reconstruction I. Conditions for a complete set of projections. Sov Phs Crystallog 20: 429 – 433
24. Orlov SS (1976) Theory of three-dimensional image reconstruction II. The recovery operator. Sov Phs Crystallog 20: 429 – 433
25. Pack J, Noo F, Clackdoyle (2005) Cone-beam reconstruction using the backprojection of locally filtered projections. IEEE Trans Med Imaging 24: 70 – 85
26. Proksa R, Kohler T, Grass M et al (2000) The n-PI-method for helical cone-beam CT. IEEE Trans Med Imaging 19: 848 – 863
27. Ra JB, Jim CB, Cho ZH et al (1992) A true 3D reconstruction algorithm for the spherical positron tomography. Phys Med Biol 27: 37 – 50
28. Schaller S, Noo F, Sauer F et al (2000) Exact Radon rebinning algorithm for the long object problem in helical cone-beam CT. IEEE Trans Med Imaging 19: 822 – 834
29. Smith BD (1985) Image reconstruction from cone-beam projection: Necessary and sufficient conditions and reconstruction methods. IEEE Trans Med Imaging MI-4: 14 – 25
30. Stazyk M, Rogers J, Harrop R (1992) Analytic image reconstruction in PVI using the 3D Radon transform. Phys Med Biol 37: 689 – 704
31. Stearns CW, Chesler DA, Brownell GL (1990) Accelerated image reconstruction for a cylindrical positron tomography using Fourier domain methods. IEEE Trans Nucl Sci 37: 773 – 777
32. Stearns CW, Crawford CR, Hu H (1994) Oversampled filters for quantitative volumetric PET reconstruction. Phys Med Biol 39: 381 – 388

33. Taguchi K, Aradate H (1998) Algorithm for image reconstruction in multi-slice helical CT. Med Phys 25: 550 – 561
34. Tam KC, Samarasekera S, Sauer F (1998) Exact cone-beam CT with a spiral scan. Phys Med Biol 43: 1015 – 1024
35. Turbell H, Danielsson PE (2000) Helical cone-beam tomography. Int J Imaging Syst Technol 11: 91 – 100
36. Tuy HK (1983) An inverse formula for cone-beam reconstruction. SIAM J Appl Math 43: 546 – 552
37. Wang G, Vannier MW (1993) Helical CT image noise-analytical results. Med Phys 20: 1653 – 1640
38. Wang G, Lin TH, Cheng P et al (1993) A general cone-beam reconstruction algorithm. IEEE Trans Med Imaging 12: 486 – 496
39. Wang G, Ye Y, Yu H, (2007) Approximate and exact cone-beam reconstruction with standard and non-standard spiral scanning. Phys Biol Med 52: R1 – R13
40. Ye Y, Zhao S, Yu H et al (2005) A general exact reconstruction for cone-beam CT via backprojection-filtration. IEEE Trans Med Imaging 24: 1190 – 1198
41. Zeng GL, Gullberg GT (1992) A cone-beam tomography algorithm for orthogonal circle-and-line orbit. Phys Med Biol 37: 563 – 577
42. Zeng GL, Clack R, Gullberg GT (1994) Implementation of Tuy's inversion formula. Phys Med Biol 39: 493 – 507
43. Zhuang TL, Leng S, Nett BE et al (2004) Fan-beam and cone-beam image reconstruction via filtering the backprojection image of differentiated data. Phys Med Biol 49: 5489 – 5503
44. Zou Y, Pan X (2004) An extended data function and its generalized backprojection for image reconstruction in helical cone-beam CT. Phys Med Biol 49: N383 – N387
45. Zou Y, Pan X, Xia D et al (2005) PI-line-based image reconstruction in helical cone-beam computed tomography with a variable pitch. Med Phys 32: 2639 – 2648

6 Iterative Reconstruction

Previous chapters deal with analytic image reconstruction algorithms. This chapter, on the other hand, introduces iterative image reconstruction algorithms. Due to high speed computers, iterative algorithms get more and more attention in medical image reconstruction. This chapter describes the imaging problem as a system of linear equations, and reconstructs an image by minimizing an objective function. Many algorithms are available to solve the system of linear equations or to minimize an objective function. The objective function can be set up by using the likelihood function, and can also include the prior knowledge about the image. The likelihood function models the noise distribution in the projection measurements. The ML-EM algorithm or OS-EM algorithm is the most popular iterative image reconstruction algorithm in emission tomography, and this chapter has devoted significant efforts to it. Many strategies for noise control are discussed. This chapter also presents a recent research hot spot — image reconstruction with highly undersampled data, which is often referred to as compressed sensing and is, in fact, nothing but another application of Bayesian image reconstruction.

6.1 Solving a System of Linear Equations

Instead of using an analytical algorithm to reconstruct an image, image reconstruction can also be obtained by solving a system of linear equations. In doing so, the image is first discretized into pixels or voxels (volumetric pixels) as illustrated in Figure 6.1.

Here, the image pixels $x_j (j = 1, 2, \ldots)$ are labeled in a 1D sequential order, as are all projections $p_i (i = 1, 2, \ldots)$. For the simple example in Figure 6.1, we can relate the image pixels and the projections using a system of linear equations:

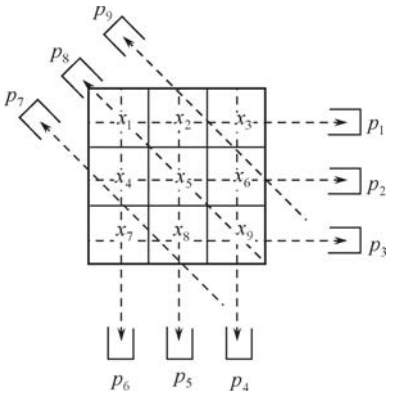


Fig. 6.1. An example with 9 unknowns and 9 measurements.

$$\begin{cases} x_1 + x_2 + x_3 = p_1, \\ x_4 + x_5 + x_6 = p_2, \\ x_7 + x_8 + x_9 = p_3, \\ x_3 + x_6 + x_9 = p_4, \\ x_2 + x_5 + x_8 = p_5, \\ x_1 + x_4 + x_7 = p_6, \\ 2(\sqrt{2}-1)x_4 + (2-\sqrt{2})x_7 + 2(\sqrt{2}-1)x_8 = p_7, \\ \sqrt{2}x_1 + \sqrt{2}x_5 + \sqrt{2}x_9 = p_8, \\ 2(\sqrt{2}-1)x_2 + (2-\sqrt{2})x_3 + 2(\sqrt{2}-1)x_6 = p_9. \end{cases} \tag{6.1.1}$$

This system can be re-written in the matrix form as

$$AX = P, \tag{6.1.2}$$

where $X = [x_1, x_2, \ldots, x_9]^{\mathrm{T}}, P = [p_1, p_2, \ldots, p_9]^{\mathrm{T}}$, and A is the coefficient matrix of the system. The element a_{ij} in A represents the weight of the contribution of the jth pixel x_j to the ith projection p_i. In this example, the contribution is the segment length of the projection ray within the pixel. If the inverse matrix A^{-1} of A exists, the reconstructed image is given by

$$X = A^{-1}P. \tag{6.1.3}$$

Line-length is not the only way to model the "distribution." Some imaging physics (e.g., attenuation and point spread function) can also be included as well.

For a practical imaging problem, the matrix A is not square. In this case a generalized inverse of the matrix can be used. For example, we can find a

least-squares solution:

$$X = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}P, \text{ if the system is over-determined;} \tag{6.1.4}$$

$$X = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}P, \text{ if the system is under-determined.} \tag{6.1.5}$$

A generalized inverse can be obtained via a least-squares minimization. In the case that the system is over-determined (i.e., the number of projection rays is greater than the number of image pixels), we let

$$\begin{aligned}\chi^2 &= \|AX - P\|^2 = (AX - P)^{\mathrm{T}}(AX - P) \\ &= X^{\mathrm{T}}A^{\mathrm{T}}AX - 2X^{\mathrm{T}}A^{\mathrm{T}}P - P^{\mathrm{T}}P.\end{aligned} \tag{6.1.6}$$

and set the partial derivatives (i.e., gradient) to zero:

$$\mathbf{0} = \nabla\chi^2 = 2A^{\mathrm{T}}AX - 2A^{\mathrm{T}}P. \tag{6.1.7}$$

Re-arranging the terms, we have

$$A^{\mathrm{T}}AX = A^{\mathrm{T}}P \tag{6.1.8}$$

which is a set of normal equations, because (AX-P) is orthogonal (i.e., normal) to the rows of A : $A^{\mathrm{T}}(AX - P) = \mathbf{0}$. Solving the normal equations immediately yields a generalized solution

$$X = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}P. \tag{6.1.9}$$

On the other hand, in the case that the system is under-determined (i.e., the number of image pixels is greater than the number of projection rays), the system will have infinite number of solutions for $AX = P$, assuming that the system is consistent. In this case we would like to choose the minimum norm solution. Therefore, we use the method of Lagrange multipliers to minimize $\|X\|^2$ subject to $AX = P$. We thus set up a Lagrange function

$$L = \|X\|^2 + \Lambda(AX - P) \tag{6.1.10}$$

with a diagonal matrix Λ=diag $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ containing the Lagrange multipliers $\lambda_1, \lambda_2, \ldots, \lambda_m$ and m being the number of projection rays.

Setting the partial derivatives (i.e., gradient) of the Lagrange function to zero yields

$$\mathbf{0} = 2X + A^{\mathrm{T}}\Lambda \text{ and } AX = P. \tag{6.1.11}$$

Pre-multiplying with matrix A, $\mathbf{0} = 2X + A^T\Lambda$ becomes

$$\mathbf{0} = 2AX + AA^{\mathrm{T}}\Lambda. \tag{6.1.12}$$

Solving for Λ and using $AX = P$, we have

$$\Lambda = -2(AA^{\mathrm{T}})^{-1}AX = -2(AA^{\mathrm{T}})^{-1}P. \tag{6.1.13}$$

Finally, solving for X from $\mathbf{0} = 2X + A^{\mathrm{T}}\Lambda$ gives

$$X = -\frac{1}{2}A^{\mathrm{T}}\Lambda = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}P. \tag{6.1.14}$$

Even if the matrix A is square, its inverse A^{-1} may not exist. When A is not full rank, A^{-1} does not exist. In fact, the matrix A for the example in Figure 6.1 is not full rank. One can easily check that the sum of the rows 1, 2, and 3 is the same as the sum of the rows 4, 5, and 6. If the matrix A is not full rank, we cannot even calculate $(AA^{\mathrm{T}})^{-1}$ or $(A^{\mathrm{T}}A)^{-1}$. In about all applications, the matrix A is not full rank and not square either, and the projections are not consistent due to noise. If the matrix A is rank deficient, you could use the singular value decomposition (SVD) to find a pseudo inverse.

The singular value decomposition (SVD) technique is a powerful and stable method to find a generalized inverse and diagnose the system condition. Now we use SVD to find the least-squares solution for $AX = P$ as follows.

Assume that matrix A has m rows and n columns, and is denoted as $A_{m\times n}$. Using SVD, the matrix $A_{m\times n}$ can be decomposed into

$$A_{m\times n} = U_{m\times m}\Sigma V^{\mathrm{T}}_{n\times n}, \tag{6.1.15}$$

where

$$V^{\mathrm{T}}V = I_{n\times n}, \tag{6.1.16}$$

$$U^{\mathrm{T}}U = I_{m\times m}, \tag{6.1.17}$$

$$\Sigma_{m\times n} = \begin{bmatrix} diag\{\sigma_i\} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6.1.18}$$

with the singular values arranged in the descending order:

$$\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_i \geqslant \ldots \geqslant 0. \tag{6.1.19}$$

A generalized inverse (or, pseudo-inverse) is defined as

$$A^{+} = V\Sigma^{+}U^{\mathrm{T}} \tag{6.1.20}$$

where

$$\Sigma^{+}_{n\times m} = \begin{bmatrix} D_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{6.1.21}$$

and the diagonal matrix D_r with a cut-off index r is defined as

$$D_r = diag\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right\}. \tag{6.1.22}$$

The reconstructed image is given as

$$X = A^{+}P = V\Sigma^{+}U^{\mathrm{T}}P. \tag{6.1.23}$$

In the SVD method, the user selects the cut-off index r. If a very small r is chosen, the resultant reconstructed image only contains low frequency components. If a very large r is chosen, the resultant reconstructed image will contain high frequency components and the image is noisy as well.

More often than not, the matrix A is too large to store in the computer; it can only be generated one row at a time when this row is used in solving the system of equations. Not every method that is able to solve a system of linear equations can be used here. For example, the methods based on diagonalizing the matrix A or transforming matrix A into an upper triangle matrix are not applicable. Any methods that modify the matrix A cannot be used. We can only use methods that use matrix A and its transposed matrix A^{T}. Therefore, iterative methods that only use A and A^{T} (but do not modify them) make sense in finding an approximate solution to our imaging problem.

An analytic reconstruction can be thought of as an "open loop" system, while an iterative algorithm can be thought of as a "closed loop" system. Each loop, referred to as an iteration, usually consists of a projection operation, a comparison of the projected data with the measured data, and a backprojection operation. The backprojection maps the data discrepancies from the projection space to the image space. The backprojected discrepancies will be used to modify the currently estimated image at each iteration (see Figure 6.2).

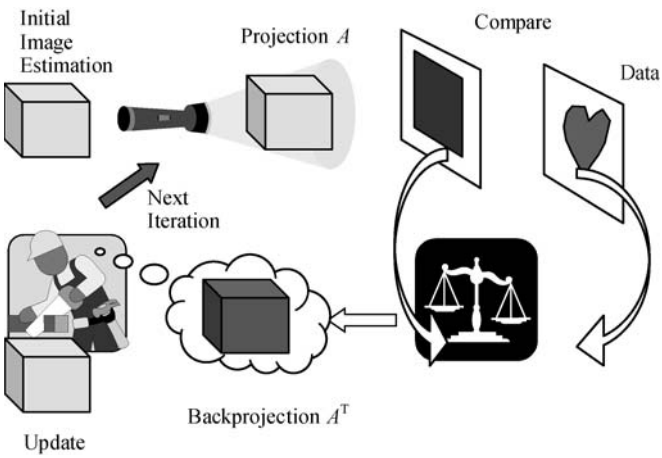


Fig. 6.2. A general procedure of an iterative image reconstruction algorithm.

6.2 Algebraic Reconstruction Technique

The main idea of the ART (algebraic reconstruction technique) algorithm (which is also known as the Kaczmarz method) is to make the estimated image satisfy one equation at a time as illustrated in Figure 6.3, where 3 lines — L_1, L_2, and L_3 — represent 3 equations, and their intersection is the solution. In this example, the image only consists of 2 pixels.

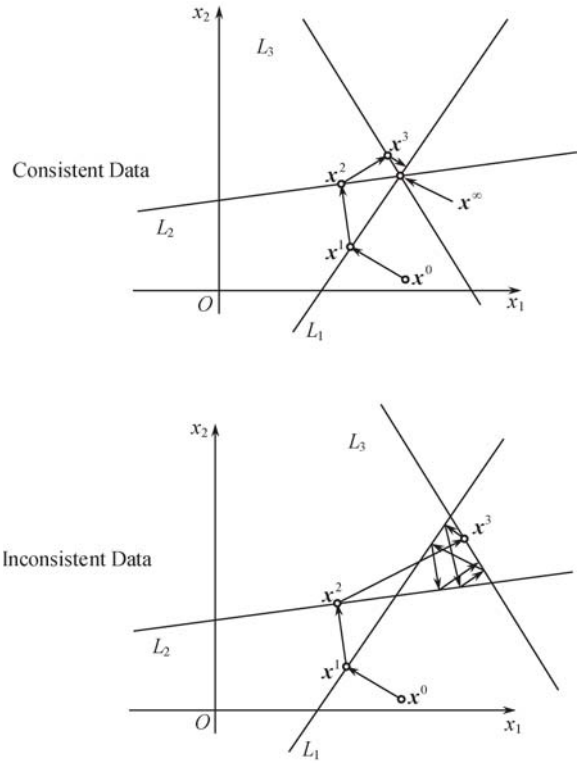


Fig. 6.3. The ART algorithm tries to satisfy each equation at each update.

In Figure 6.3, $\boldsymbol{x}^0$ is the initial guess of the solution. The first step is to project this point $\boldsymbol{x}^0$ perpendicularly onto L_1, obtaining $\boldsymbol{x}^1$. Next, project $\boldsymbol{x}^1$ perpendicularly onto L_2 to obtain $\boldsymbol{x}^2$, and so on, projecting each point onto a line (which is one equation) one at a time. Eventually, the algorithm will converge to the solution of the system of equations (see Figure 6.3 Upper). If the equations are not consistent, the algorithm will bounce around and never

converge (see Figure 6.3 Lower). One iteration is defined as the procedure of going through all the equations once.

The ART algorithm is executed one projection ray at a time, and the image is updated after each ray is considered. Symbolically, the algorithm can be written as

$$\boldsymbol{x}^{next} = \boldsymbol{x}^{current} - Backproject_{ray}\left\{\frac{Project_{ray}(\boldsymbol{x}^{current}) - Measurement_{ray}}{Normalization\ Factor}\right\}. \tag{6.2.1}$$

6.3 Gradient Descent Algorithms

First an objective function χ^2 is formed based on the system of imaging equations:

$$\chi^2 = \|AX - P\|^2, \tag{6.3.1}$$

which is a quadratic function (see Figure 6.4). Due to the noise, the equations are inconsistent. Thus the minimum value of the objective function χ^2 usually has a non-zero, positive value.

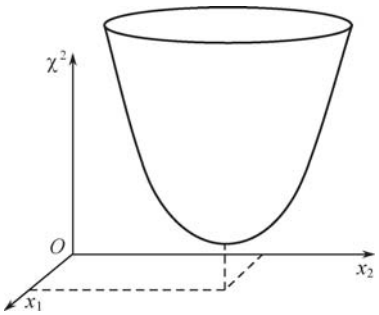


Fig. 6.4. A quadratic objective function.

The strategy of gradient descent algorithms is to evaluate the gradient of the objective function χ^2 and use the gradient information to find the minimum of the objective function. The gradient in 1D is the derivative of the function. A positive gradient means an upward direction, and a negative gradient means a downward direction. The gradient descent algorithms take the opposite direction of the direction that is indicated by the gradient and use a small enough step size so that the algorithms can find the minimum

(see Figure 6.5). The general form of a gradient descent algorithm looks like

$$\boldsymbol{x}^{next} = \boldsymbol{x}^{current} - a_{current}\boldsymbol{\Delta}(\boldsymbol{x}^{current}), \tag{6.3.2}$$

where $\boldsymbol{\Delta}$ is the gradient of the objective function χ^2 at $\boldsymbol{x}^{current}$ and contains

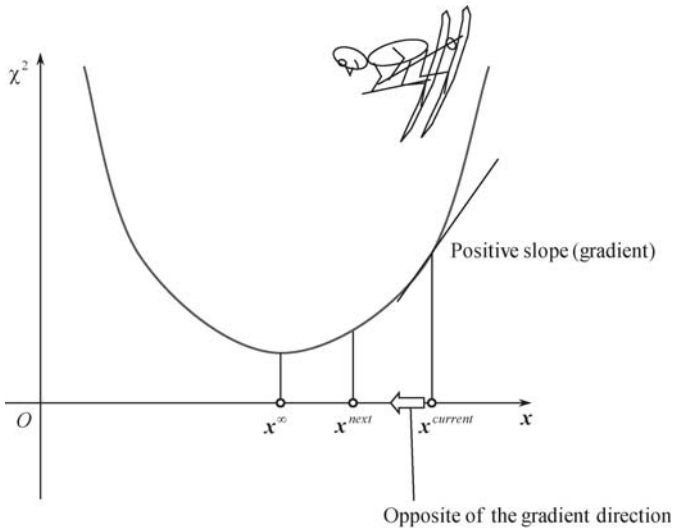


Fig. 6.5. The opposite direction of the gradient is the downhill direction.

both projection and backprojection at all rays. In fact,

$$\boldsymbol{\Delta} = \nabla \|AX - P\|^2 = 2A^{\mathrm{T}}(AX - P), \tag{6.3.3}$$

where ∇ is the notation for the gradient operator, the projection AX is the multiplication of X by matrix A, and the backprojection is multiplication by matrix A^{T}. The data discrepancy is $(AX - P)$. The algorithm converges when $AX = P$ and X does not change any more.

If the system is inconsistent, the algorithm converges when $A^{\mathrm{T}}(AX-P) = \mathbf{0}$. In our notation, X and $\boldsymbol{x}$ are the same thing. If the system is underdetermined, this least-squares problem does not have a unique solution, and the objective function χ^2 has a long valley (see Figure 6.6). The solution will depend upon the initial solution. If the initial solution is zero (that is, $\boldsymbol{x}^0 = \mathbf{0}$), then the algorithm will converge to a minimum norm solution.

Due to noise, we seldom ever have $AX = P$ at convergence. Instead, we get a very noisy image when the iteration number is large. We apply an iterative reconstruction algorithm to noisy data generated with the phantom in Figure 2.10. As shown in Figure 6.7, at the early iterations, the images only contain low frequency components; at higher iterations, high frequency components are recovered and noise comes into effect, too.

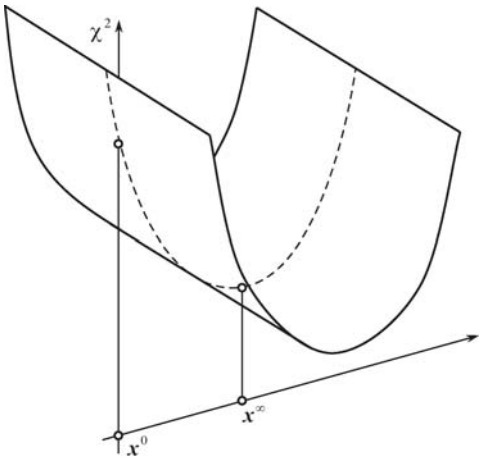


Fig. 6.6. For a degenerated system, the iterative algorithm solution depends on the initial value.

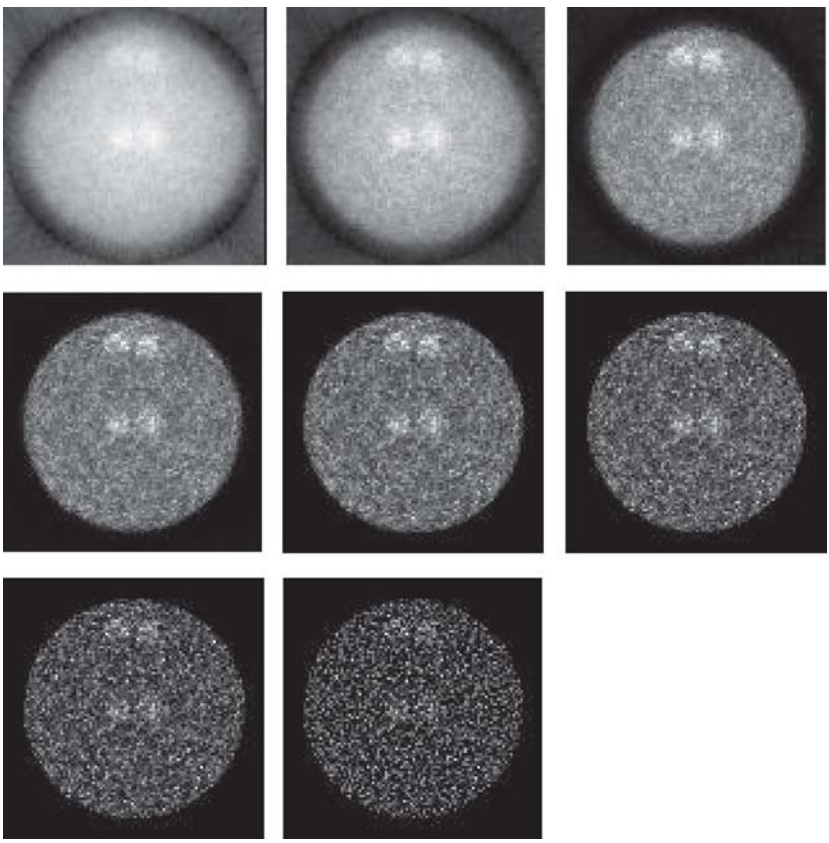


Fig. 6.7. The image gets noisier as the iteration number gets larger.

The gradient direction is easy to compute in a practical imaging problem using $\boldsymbol{\Delta} = \nabla \|AX - P\|^2 = 2A^{\mathrm{T}}(AX - P)$, but the negative gradient direction may not be the optimal direction to use in finding the optimum image. Let us look at the contour lines of a typical objective function χ^2 in Figure 6.8, where the contour lines are ellipses. The gradient direction at any point is perpendicular to the tangent of the ellipse. The searching directions of two consequential steps, $\boldsymbol{u}^{current}$ and $\boldsymbol{u}^{next}$, are orthogonal to each other (that is, $\boldsymbol{u}^{current} \cdot \boldsymbol{u}^{next} = 0$). The searching path is zigzagging and not optimal.

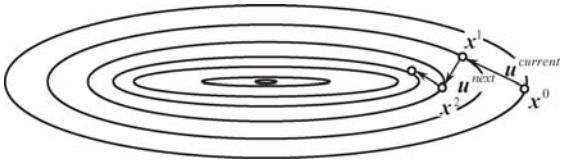


Fig. 6.8. The negative gradient direction may not be the most efficient way to find the function's minimum.

A better searching direction is to use the concept of *conjugate directions*. The conjugate directions are defined by $\boldsymbol{u}^{current} \cdot (A^{\mathrm{T}}A)\boldsymbol{u}^{next} = 0$ (see Figure 6.9). The shape of the objective function χ^2 is characterized by $(A^{\mathrm{T}}A)$. When we use the conjugate directions, we actually first deform the contour lines into circles then find the orthogonal directions. The conjugate directions make the algorithm converge faster.

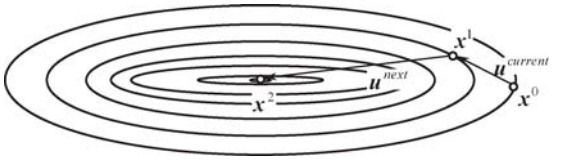


Fig. 6.9. The conjugate gradient direction is more effective than the gradient direction.

6.4 Maximum-Likelihood Expectation-Maximization Algorithms

We do not have to use a least-squares objective function. There are many different ways to set up an objective function. If we use the Poisson noise model or simply use the non-negativity constraint, we will get a special objective function. By minimizing that special objective function, a multiplicative

updating algorithm known as the ML-EM (maximum-likelihood expectation-maximization) algorithm is derived and can be symbolically expressed as

$$\boldsymbol{x}^{next} = \boldsymbol{x}^{current} \frac{Backproject\left\{\dfrac{Measurement}{Project\ (\boldsymbol{x}^{current})}\right\}}{Backproject\ \{\mathbf{1}\}},$$

where $\mathbf{1}$ is a vector with elements of 1's. The size of the vector is that of the projection data vector. In this algorithm, the data discrepancy is calculated as a ratio instead of a difference. The distinguishing feature of this algorithm is its non-negativity. If the initial image $\boldsymbol{x}^0$ does not contain any negative pixels or voxels, the image values will never become negative.

Now let us explain the name of this algorithm: ML-EM. The objective function of this algorithm can be a likelihood function, which is the joint probability density function of Poisson random variables. We are looking for a solution (i.e., the reconstructed image) that can maximize this likelihood function. Therefore, this is a Maximum Likelihood (ML) algorithm.

When we try to maximize or minimize a function (e.g., an objective function or a likelihood function), we usually take the partial derivatives with respect to all of its unknowns (i.e., the pixel or voxel values), set these derivatives to zero, and solve for the unknowns. It turns out that our Poisson likelihood function is too complicated for us to optimize. We take the expectation value (or the statistical mean value) of the likelihood function. This is the "E" step, and it simplifies the problem significantly. We then find the maximum of the expected likelihood function. This is the "M" step. Therefore, we have the name "EM," the Expectation-Maximization.

This ML-EM algorithm is also called the Richardson-Lucy algorithm or Lucy-Richardson algorithm, because Richardson and Lucy developed this algorithm for image deblurring applications in 1972 and 1974. There are many EM algorithms in different fields of research. The usual ML-EM algorithm is derived and used particularly for the emission data reconstruction. We also have transmission-data ML-EM algorithms, too, but they are not as popular.

6.5 Ordered-Subset Expectation-Maximization Algorithm

In ART, the image is updated after each projection ray is considered. On the other hand, in gradient descent methods and in the ML-EM algorithm, the image is updated only when all projection rays are considered. One way to speed up the convergence rate of an iterative algorithm is to make more frequent image updates.

In an OS-EM (ordered-subset expectation-maximization) algorithm, the projection views are grouped in different sets (called subsets), the algorithm goes through the subsets in a specified order, and the image is updated after each subset is considered. Figure 6.10 shows an example of how the projection views are divided into subsets. There are many strategies for dividing the views into subsets.

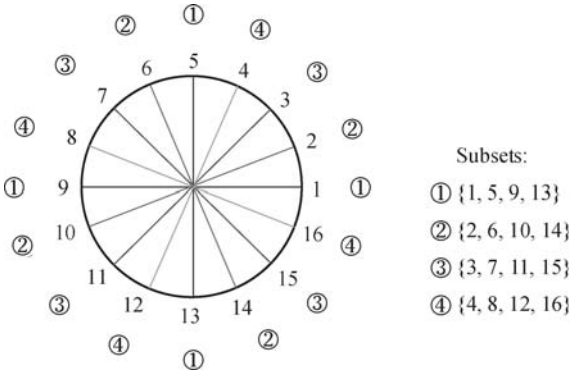


Fig. 6.10. The total 16 projection views are divided in to 4 subsets.

Increasing the number of subsets accelerates the convergence rate but may increase the noise as well. Roughly speaking, if you have N subsets, you may accelerate the ML-EM algorithm about N times. Modest acceleration of approximately 10 times is possible with very little increase in noise.

6.6 Noise Handling

Nowadays, many people choose an iterative image reconstruction algorithm over an analytical algorithm simply because the iterative algorithm can provide images containing less noise with the same or better resolution. We will investigate how the analytical and iterative algorithms handle the noise in this section.

6.6.1 Analytical Methods — Windowing

In an analytical algorithm, noise regulation is achieved via the application of a window function when the projection data are filtered. The filter in an image reconstruction algorithm is always a high-pass filter (e.g., the ramp

filter) in which the high-frequency components are enhanced more than the low-frequency components. In order to suppress the high-frequency noise, a window function is always applied to the ramp filter (see Figure 6.11). Basically, the noise regulation strategy in an analytical algorithm is to control the bandwidth. Thus, both high frequency noise and high frequency signal are discarded.

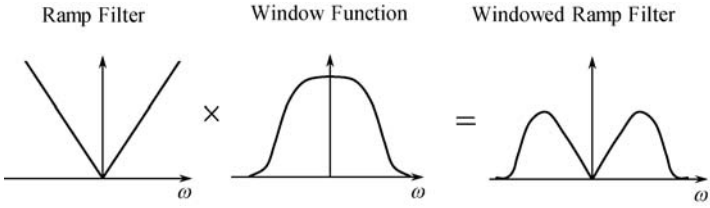


Fig. 6.11. Application of a window function to the ramp filter.

6.6.2 Iterative Methods — Stopping Early

There are many ways to control the noise in an iterative algorithm. We can first consider a rough noise propagation model of a linear iterative algorithm:

$$Error_{image} = \lambda_n(\omega) \times Error_{data}, \tag{6.6.1}$$

where $Error_{image}$ is the error magnitude in the reconstructed image, $Error_{data}$ is the error magnitude in the projections, and $\lambda_n(\omega)$ is the algorithm transfer function which depends on the frequency ω and the iteration number n.

We can compare an iterative reconstruction algorithm with an SVD matrix pseudo-inverse solution. You may imagine that $\lambda_n(\omega)$ contains the information of both the singular values and singular vectors of the imaging matrix A. The frequency components are in the singular vectors. As the iteration number n increases, more singular vectors join $\lambda_n(\omega)$. The iteration number is somehow related to the cut-off index in an SVD pseudo inversion expression. With a larger iteration number n, $\lambda_n(\omega)$ contains components with higher frequencies. In some linear algorithms, this relationship can be simplified to

$$Error_{image} = \kappa \times Error_{data}, \tag{6.6.2}$$

where κ is similar to the condition number of matrix A, and κ is defined as the ratio of the largest singular value σ_1 over the cut-off singular value σ_n. This simplification is reasonable because the "worst" noise influence comes from the frequency components (i.e., the singular vector) corresponding to the current smallest singular value σ_n. In the SVD pseudo inverse method,

the reconstructed image is a sum of many terms. Each term is a product of a component (i.e., the singular vector) and the reciprocal of its corresponding singular value $1/\sigma$. The largest gain comes from $1/\sigma_n$, which corresponds to a singular vector with very high frequencies.

In your mind, you can imagine that $\lambda_n(\omega)$ is a windowed ramp filter (see Figure 6.11). The width of the window increases as the iteration number increases. We immediately see that one way to control noise is to control the iteration number.

This analogy between an iterative reconstruction and an SVD pseudo inverse is not mathematically correct, but it is a way of showing the similarity of these two approaches. This analogy can give us some insight of an iterative algorithm.

Stopping early is the simplest way to regulate the noise. However, iterative algorithms do not have a uniform convergence rate throughout the image. After an iterative algorithm is stopped, the resultant image will have non-uniform resolution. If you would like your reconstructed image to have uniform resolution, one remedy is to over-iterate (i.e., not to stop early) and then apply a post filter to suppress the noise.

6.6.3 Iterative Methods — Choosing Pixels

The second way to reduce image noise is to reduce the errors in the data. This approach of regularization is a unique feature for the iterative algorithm. The errors between the projections P and the model AX, $Error_{data}$, consist of two parts: deterministic errors and random errors. The deterministic errors are generated from the non-ideal system model AX. First of all, discretizing the continuous object into pixels (or voxels) may cause errors. One must consider the trade-offs when deciding pixel size. Smaller pixels give a more accurate model but increase the number of unknowns to be solved. Larger pixels make the image model less accurate, but fewer unknowns can make the inverse problem more stable.

Using non-overlapping uniform pixels or voxels to model an image is not an ideal approach because this image model contains a lot of discontinuity in the image and introduces too many artificial high-frequency components into the image. Some people have tried to use overlapping non-uniform pixels (or voxels) such as blobs (see Figure 6.12), which results in improved image quality. This gives a more realistic band-limited image model.

One drawback of using blobs as image voxels is the increased computational complexity. An alternative approach has been investigated to achieve the same effect but with better computational efficiency. This strategy uses the traditional non-overlapping voxels in the image, but a low-pass filter is applied to the backprojected image. The kernel function of the low-pass filter is chosen as the 3D "profile" of the blobs. In other words, the backprojected

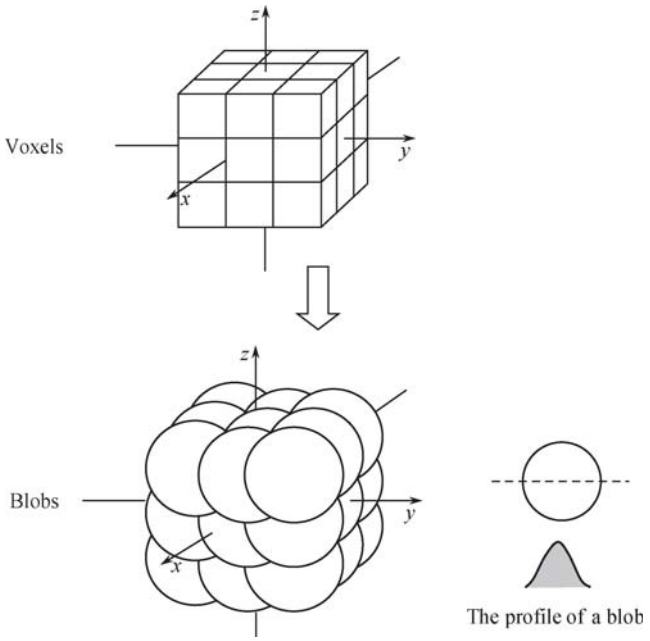


Fig. 6.12. Using overlapping blobs to replace the traditional voxels can better model the image.

image is three-dimensionally convolved with the blob (see Figure 6.13).

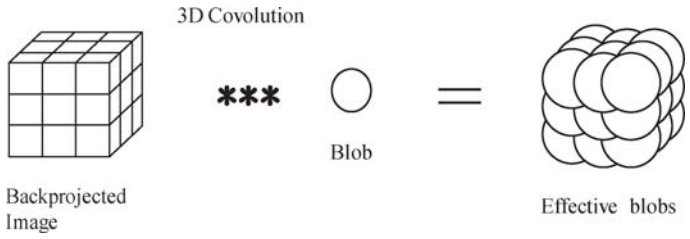


Fig. 6.13. An alternative approach to get the blob effect.

To make the inverse problem more stable, as a rule of thumb, we would select the pixel size larger than the detector bin size; this makes the number of image pixels smaller than the number of detector bins (see Figure 6.14). In practice, it is advantageous to choose a large array size (with a small bin dimension) on the detector during data acquisition. This makes a big difference in noise control in an iterative algorithm, especially when the system resolution is modeled in the projector/backprojector. A balanced selection of the detector bin size is half the size of the image pixel. If the image size is $256 \times 256 \times 256$ and there is no option on the scanner to acquire 512×512

projections, you can acquire data using the 256 × 256 mode and interpolate the data into 512 × 512 arrays during iterative image reconstruction.

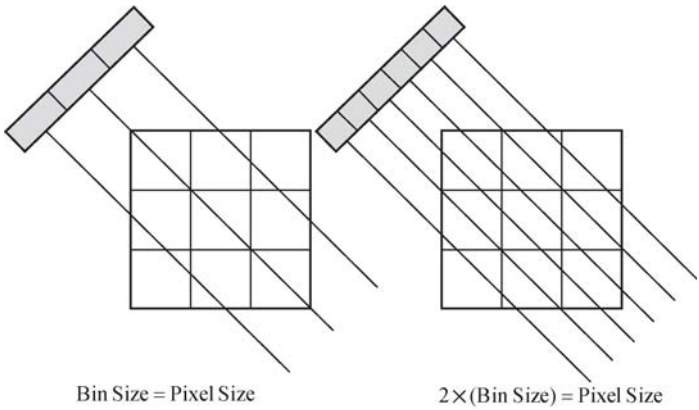


Fig. 6.14. It is advantageous to use a detector bin size that is smaller than the image pixel size.

6.6.4 Iterative Methods — Accurate Modeling

Modeling the system's point spread function (see Figure 6.15) and patient induced attenuation and scattering in the matrix A will significantly reduce the deterministic errors between the projections P and the model AX. If you are not ready to model the imaging physics in matrix A, you still have a choice of line-length weighting or area weighting of the image path within the pixel to be used in calculating the elements in matrix A (see Figure 6.16). The freedom to model the imaging system with various geometries and physics effects is the main advantage of using an iterative reconstruction algorithm. We are able to control the errors between the projection data and the model to some degree; we can at least control the deterministic part of them with good system modeling. Smaller data modeling errors result in smaller errors in the image. With reduced data modeling error, we can increase the iteration number to get better image resolution with the same or reduced image noise.

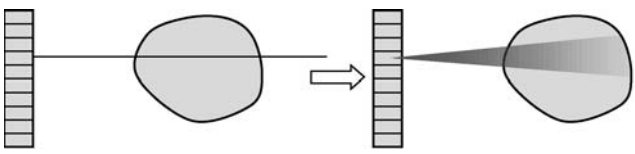


Fig. 6.15. Modeling the system distance-dependent resolution and sensitivity in the projector.

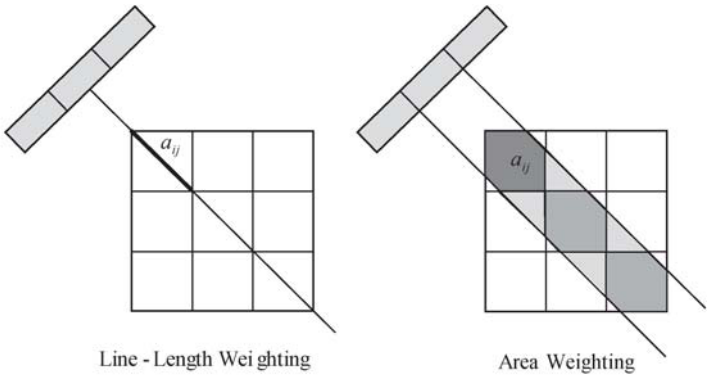


Fig. 6.16. Area weighting is a better model than the line-length weighting but has a higher computation cost.

There is a fourth way to control the noise in an iterative algorithm: the random part. We can model the noise distribution in the objective function. Section 6.7 will be dedicated to this topic.

The fifth way is to use the prior knowledge about the image that we are looking for in addition to using the projection data alone. This topic will be covered in Section 6.8.

6.7 Noise Modeling as a Likelihood Function

In order for the noise model to work, we must have redundant measurements, otherwise the noise model has no effect on the solution. In Figure 6.17, there are two lines, L_1 and L_2, representing two independent measurements described by two independent linear equations. These measurements can be noisy or noiseless. The solution is the intersection of these two lines, regardless of the presence of noise or if you trust L_1 more or L_2 more. Due to noise, this intersection may not be the true solution at all. There is nothing we can do to improve upon the solution if we only have two measurements.

What if we have three measurements L_1, L_2, and L_3 (see Figure 6.18)? Because of noise influence the three lines do not intersect at one point. How should we pick a reasonable solution? A wise decision would depend on the noisiness of each measurement. We should trust the measurement with less noise more, and trust measurement with more noise less.

If we use the variance σ^2 to characterize the noisiness of a measurement, we can assign a weighting factor $1/\sigma^2$ to that measurement. Thus we can use a variance-based weighting scheme to select a solution. Our old least-squares

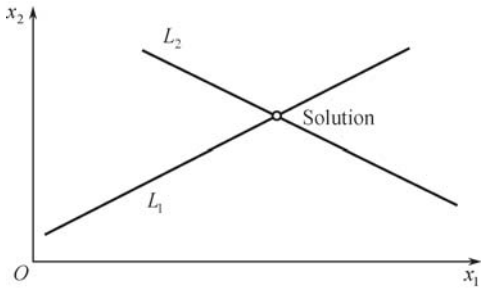


Fig. 6.17. Two lines intersect at one point, which is the solution of the corresponding two equations.

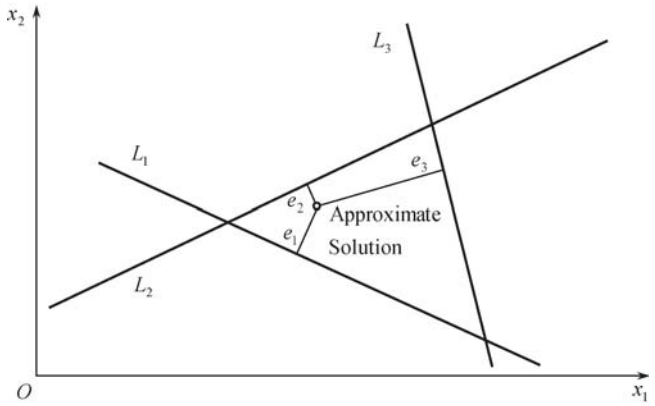


Fig. 6.18. If the system has redundant measurements and is not consistent, one can use the noisiness to weight each measurement and find an approximate solution. In this example, we assume that L_3 is noisier than L_1, and L_2 is less noisy than L_1.

objective function

$$\chi^2 = \|AX - P\|^2 = (AX - P)^{\mathrm{T}}(AX - P) \tag{6.7.1}$$

becomes a weighted least-squares objective function

$$\chi^2_W = (AX - P)^{\mathrm{T}} W (AX - P), \tag{6.7.2}$$

where W is a diagonal matrix $W = diag\left\{\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \ldots, \frac{1}{\sigma_N^2}\right\}$, and N is the number of projections.

This weighted, least-squares objective function can also be obtained through the likelihood function by assuming Gaussian noise in the projections. The ith projection measurement p_i is a Gaussian random variable whose mean value is $\mu_i = \sum_j a_{ij} x_j = A_i X$ and variance is σ_i^2. Here, A_i is

the ith row of the matrix A. The Gaussian distribution density function gives

$$Prob(p_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(A_i X - p_i)^2}{2\sigma_i^2}\right). \tag{6.7.3}$$

We assume that all projections are statistically independent. The likelihood function is the joint probability density function by considering all projections together:

$$Prob(P) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(A_i X - p_i)^2}{2\sigma_i^2}\right). \tag{6.7.4}$$

Our goal is to find an image X that maximizes the above likelihood function, hence the term "maximum likelihood solution." Taking the logarithm of the likelihood function, we have

$$\ln(Prob(P)) = -\frac{1}{2}\sum_i \frac{(A_i X - p_i)^2}{\sigma_i^2} + \sum_i \ln\left(\frac{1}{\sqrt{2\pi}\sigma_i}\right). \tag{6.7.5}$$

The second term in the above equation is a constant. Therefore, maximizing the likelihood function is equivalent to minimizing the following weighted least-squares objective function

$$\chi^2_W = \sum_i \frac{(A_i X - p_i)^2}{\sigma_i^2} = (AX - P)^{\mathrm{T}} W (AX - P). \tag{6.7.6}$$

If the data noise is not Gaussian, the above approach of setting up a likelihood function and an objective function still applies. However, the resultant objective function is different.

6.8 Including Prior Knowledge

We sometimes know more about the image that we are looking for — in addition to the measurements. We can enforce this prior knowledge into the image by adding an extra term to the objective function. This is normally called a Bayesian method.

For example, if we know in advance that the image X is very smooth, we can add a penalty term $\|\nabla X\|^2$ to suppress sharp jumps and encourage the smoothness:

$$\text{New Objective Function}(X) = \text{Old Objective Function}(X) + \beta \|\nabla X\|^2, \tag{6.8.1}$$

where β is a user specified controlling parameter. Using the squared norm of the gradient $\|\nabla X\|^2$ as a penalty term can be generalized as using an "energy" function $U(X)$ as a penalty term. The energy function $U(X)$ is defined as:

$$U(X) = \sum_{i,j} w_{ij} V(x_i - x_j), \tag{6.8.2}$$

where the summation is over a neighborhood (clique), and V is a convex function, which may or may not be quadratic (see Figure 6.19). If V is a quadratic function, this energy function encourages smoothness and penalizes jumps. If the function V increases more slowly than a quadratic function (say, V increases linearly), then it can preserve edges and smooth out the noise. How does the algorithm know which is the edge that you want to keep and which is noise that you want to smooth out? Let us answer this question by comparing a linear function $|x|$ and a quadratic function x^2. We assume that the noise consists of small jumps, while the edges separate the large jumps. When $|x|$ is small, we have $|x| > x^2$ and the linear function gives relatively heavier penalty. Thus the noise is suppressed. When $|x|$ is large, we have $|x| < x^2$ and the linear function gives a relatively much lighter penalty. Thus the edges are preserved.

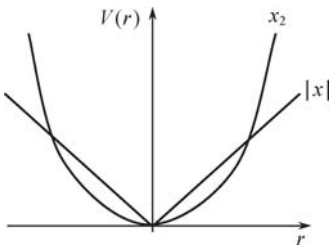


Fig. 6.19. The penalty function can have different function forms, depending on the application.

Algorithms that include the prior information carry many names, such as Bayesian methods or MAP (maximum *a posteriori*) algorithms.

The new objective function is, in fact, the conditional probability $Prob(X|P)$. Bayes' law states

$$Prob(X\big|P) = \frac{Prob(P\big|X)Prob(X)}{Prob(P)}. \tag{6.8.3}$$

Taking the logarithm yields

$$\ln(Prob(X\big|P)) = \ln(Prob(P\big|X)) + \ln(Prob(X)) - \ln(Prob(P)) \tag{6.8.4}$$

where the third term has nothing to do with our unknowns X and can be eliminated from the objective function. The Bayesian objective function then

becomes

$$L(X) = \ln(Prob(P\big|X)) + \ln(Prob(X)), \tag{6.8.5}$$

or symbolically,

$$\text{(Posterior Function)} = \text{(Likelihood Function)} + \beta\text{(Prior Function)}. \tag{6.8.6}$$

The first term on the right-hand-side is the old objective function for a maximum likelihood algorithm, and the second term contains the prior information term about the image X. This justifies our expression of the new objective function at the beginning of this section.

*6.9 Mathematical Expressions

This section gives the mathematical expressions for the ART algorithm, the ML-EM algorithm, the OS-EM algorithm, Green's one-step late algorithm, and ML-TV algorithm. Computer implementation steps are given for the conjugate gradient algorithm. The derivation of the ML-EM algorithm is also presented.

6.9.1 ART

The ART algorithm is a row-action algorithm. It considers one ray-sum at a time and can be expressed as

$$X^{next} = X^{current} - \frac{A_i X^{current} - p_i}{\|A_i\|^2} A_i^{\mathrm{T}}, \tag{6.9.1}$$

where $A_i X$ performs the forward projection along the ith projection ray, p_i is the measured projection from the ith projection bin, $\|A_i\|^2 = \sum_j a_{ij}^2$ is the sum of the squared "contribution factors" along the ith ray, and cA_i^T backprojects the value c along the ith ray. If we re-write the above algorithm in the following form

$$X^{next} = X^{current} - \left(\frac{A_i X^{current}}{\|A_i\|} - \frac{p_i}{\|A_i\|} \right) \frac{A_i^{\mathrm{T}}}{\|A_i\|}, \tag{6.9.2}$$

the geometric meaning of this algorithm can be easily explained as in Figure 6.20.

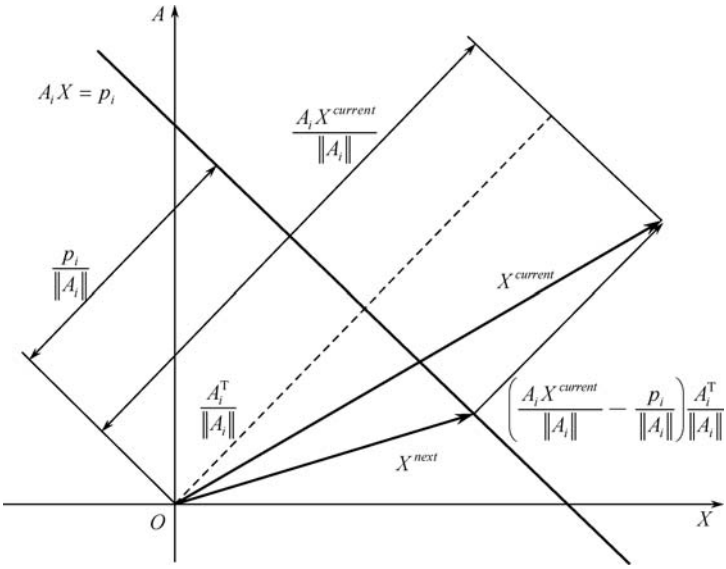


Fig. 6.20. An illustration of the ART algorithm.

There are many versions of the ART algorithms. The SIRT (Simultaneous Iterative Reconstruction Technique) does not update the image ray by ray, but update the image once per iteration. Another version is to update the image angle by angle. Another version uses a relaxation (or damping) factor to reduce the step-size and stabilize the algorithm. Still another version is called the MART algorithm, in which updating is multiplicative, instead of additive. One advantage of the MART algorithm is that the resultant image is always non-negative provided the initial guess of the image is non-negative.

6.9.2 Conjugate Gradient Algorithm

This conjugate gradient (CG) algorithm solves the normal equations

$$A^{\mathrm{T}} A X = A^{\mathrm{T}} P. \tag{6.9.3}$$

Let $M = A^{\mathrm{T}} A$ and $B = A^{\mathrm{T}} P$. The normal equations become

$$MX = B. \tag{6.9.4}$$

Matrix M is real, symmetric and positive definite (or semidefinite). The implementation steps of the CG algorithm are given as follows.

Set up initial conditions

$$X^{(0)} = \mathbf{0}, \quad R_0 = B, \quad \text{and} \quad \Delta_0 = R_0. \tag{6.9.5}$$

These three image domain arrays are represented as three 1D column vectors.

Iterations (for n = 1, 2, 3, ... do the following)

Update the step-size (a scalar) α_n as a ratio of two scalars

$$\alpha_n = \left(R_{n-1}^{\mathrm{T}} R_{n-1}\right) / \left(\Delta_{n-1}^{\mathrm{T}} M \Delta_{n-1}\right). \tag{6.9.6}$$

Update the image with

$$X^{(n)} = X^{(n-1)} + \alpha_n \Delta_{n-1}. \tag{6.9.7}$$

Calculate the residual image

$$R_n = R_{n-1} - \alpha_n M \Delta_{n-1}. \tag{6.9.8}$$

Calculate the factor (a scalar) β_n used to find the searching direction

$$\beta_n = \left(R_n^{\mathrm{T}} R_n\right) / \left(R_{n-1}^{\mathrm{T}} R_{n-1}\right). \tag{6.9.9}$$

Calculate the new searching direction for the next iteration

$$\Delta_n = R_n + \beta_n \Delta_{n-1}. \tag{6.9.10}$$

endfor

This CG algorithm has some properties:

(1) A Krylov subspace, κ_n, is expanding as the iteration number increases

$$\begin{aligned} \kappa_n &= \left\langle B, MB, M^2B, \ldots, M^{n-1}B \right\rangle \\ &= \left\langle X^{(1)}, X^{(2)}, X^{(3)}, \ldots, X^{(n)} \right\rangle \\ &= \langle R_0, R_1, R_2, \ldots, R_{n-1} \rangle \\ &= \langle \Delta_0, \Delta_1, \Delta_2, \ldots, \Delta_{n-1} \rangle \end{aligned} \tag{6.9.11}$$

where $\langle R_0, R_1, R_2, \ldots, R_{n-1} \rangle$ denotes a space spanned by $R_0, R_1, R_2, \ldots, R_{n-1}$.

(2) The residuals are orthogonal to each other, that is,

$$R_n^{\mathrm{T}} R_j = 0, \quad j < n. \tag{6.9.12}$$

(3) The search directions are M-conjugate with each other, that is,

$$\Delta_n^{\mathrm{T}} M \Delta_j = 0, \quad j < n. \tag{6.9.13}$$

(4) At each iteration, $X^{(n)}$ minimizes the objective function over the Krylov subspace κ_n. Therefore, if M is $m \times m$, the algorithm converse in at most m iterations.

What makes the iterative CG algorithm remarkable is the choice of the search direction Δ_{n-1}, which has the special property that minimizing the objective function over $X^{(n)} + \langle \Delta_{n-1} \rangle$ actually minimizes it over all of κ_n.

6.9.3 ML-EM

The emission data ML-EM algorithm is the most popular iterative algorithm in emission tomography and is expressed as

$$x_j^{next} = \frac{x_j^{current}}{\sum_i a_{ij}} \sum_i a_{ij} \frac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{current}}, \tag{6.9.14}$$

where the summation over $\hat{j}$ is the projector identical to $A_i X$ in Section 6.9.1, and the summations over i are the backprojectors. This algorithm compares the measured projection p_i with the forward projection of the current estimate $A_i X$ as a ratio. This ratio is backprojected to the image domain. The summation $\sum_i a_{ij}$ is the backprojection of constant 1 to the image domain. The ratio of these two backprojected images determines a modification factor to update the current estimate of the image.

The following is the derivation of the emission-projection ML-EM algorithm. If p is a random variable with the Poisson distribution, then its probability mass function is given as

$$\text{Prob}(p|\lambda) = \mathrm{e}^{-\lambda} \frac{\lambda^p}{p!} \tag{6.9.15}$$

where λ is the expected value of this random variable. For an imaging problem $AX = P$, the number of photons emitted from each image pixel is a Poisson random variable, and each measurement p_i can be treated as the summation of these Poisson variables. We write

$$p_i = \sum_j c_{ij}, \tag{6.9.16}$$

where c_{ij} is a Poisson random variable and

$$\lambda_{ij} = E(c_{ij}) = a_{ij} x_j. \tag{6.9.17}$$

Note that X is not random.

We can set up the likelihood function as the joint probability mass function of all Poisson distributed random variables c_{ij}:

$$\text{Prob} = \prod_{i,j} \mathrm{e}^{-\lambda_{ij}} \frac{\lambda_{ij}^{c_{ij}}}{c_{ij}!} = \prod_{i,j} \mathrm{e}^{-a_{ij} x_j} \frac{(a_{ij} x_j)^{c_{ij}}}{c_{ij}!}. \tag{6.9.18}$$

Taking the logarithm of this likelihood function yields

$$\ln(\text{Prob}) = \sum_{i,j} (c_{ij} \ln(a_{ij} x_j) - a_{ij} x_j) - \sum_{i,j} \ln(c_{ij}!). \tag{6.9.19}$$

The second summation term $\sum_{i,j} \ln(c_{ij}!)$ does not contain the parameters x_j to be estimated; therefore, it can be discarded without changing the maximum-likelihood problem. To find the maximum-likelihood solution of x_j, we will maximize the following objective function:

$$L = \sum_{i,j} \Big(c_{ij} \ln(a_{ij} x_j) - a_{ij} x_j \Big). \tag{6.9.20}$$

The "E" Step

The above objective function contains random variables c_{ij}. The "E" (expectation) step is to replace it by expected value using the measurement p_i and the current estimate of the parameters x_j. That is, c_{ij} is replaced by

$$E(c_{ij} \big| p_i, X^{current}) = \frac{a_{ij} x_j^{current}}{\sum_k a_{ik} x_k^{current}} p_i. \tag{6.9.21}$$

Thus, after the "E" step, the objective function becomes

$$E(L \big| P, X^{current}) = \sum_{i,j} \left(\frac{a_{ij} x_j^{current}}{\sum_k a_{ik} x_k^{current}} p_i \ln(a_{ij} x_j) - a_{ij} x_j \right). \tag{6.9.22}$$

The "M" Step

To maximize the new objective function $E(L \big| P, X^{current})$, we will take the derivative of it with respect to estimation parameters x_j and set the derivatives to zero, that is,

$$\begin{aligned} \frac{\partial E(L|P, X^{current})}{\partial x_j} &= \sum_i \left(\frac{a_{ij} x_j^{current}}{\sum_k a_{ik} x_k^{current}} p_i \frac{a_{ij}}{a_{ij} x_j} - a_{ij} \right) \\ &= \frac{1}{x_j} \sum_i \frac{a_{ij} x_j^{current}}{\sum_k a_{ik} x_k^{current}} p_i - \sum_i a_{ij} \\ &= 0. \end{aligned} \tag{6.9.23}$$

Solving for x_j, we finally have the ML-EM algorithm:

$$x_j^{next} = \frac{x_j^{current}}{\sum_i a_{ij}} \sum_i a_{ij} \frac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{current}}. \tag{6.9.24}$$

In fact, this multiplicative ML-EM algorithm can also be written in an additive form so that it appears like a gradient descent algorithm as

$$X^{next} = X^{current} + S^{current} A^T \Lambda^{current} (P - AX^{current}) \tag{6.9.25}$$

where the step size is

$$S^{current} = diag \left\{ \frac{x_j^{current}}{\sum_i a_{ij}} \right\}, \tag{6.9.26}$$

and the noise variance weighting is

$$\Lambda^{current} = diag \left\{ 1 / \sum_k a_{ik} x_k^{current} \right\} \approx diag \left\{ \frac{1}{p_i} \right\}. \tag{6.9.27}$$

In reality, the measurement noise may not be exactly Gaussian distributed or Poisson distributed. The author personally believes that it is not very critical what the noise distribution is, while it is very important to know the variance of the measurement noise, because it is the variance that is used in measurement weighting. For example, in an imaging system, the measurement p_i is noisy and the noise variance is related to p_i as

$$(\text{Variance of } p_i) = s_i \bar{p}_i, \tag{6.9.28}$$

where s_i can be a system scaling factor, p_i may not be Poisson distributed, and $\bar{p}_i$ is the expected value of p_i. Realizing that the imaging model is

$$\sum_k a_{ik} x_k = p_i, \tag{6.9.29}$$

we can modify the ML-EM algorithm and obtain a general image reconstruction algorithm as

$$x_j^{next} = \frac{x_j^{current}}{\sum_i a_{ij} \frac{1}{s_i}} \sum_i a_{ij} \frac{p_i}{s_i \sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{current}}. \tag{6.9.30}$$

If we re-write this modified algorithm in the additive form, the noise variance weighting is

$$\Lambda^{current} = diag \left\{ \frac{1}{\left(s_i \sum_k a_{ik} x_k^{current} \right)} \right\} \approx diag \left\{ \frac{1}{(s_i p_i)} \right\}. \tag{6.9.31}$$

6.9.4 OS-EM

With minor changes to the ML-EM algorithm, the ordered subset version of it can be readily obtained as the OS-EM algorithm:

$$x_j^{next} = \frac{x_j^{current}}{\sum_{i \in S_k} a_{ij}} \sum_{i \in S_k} a_{ij} \frac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{current}}, \tag{6.9.32}$$

where S_k represents the kth subset of the projections.

6.9.5 Green's One-Step Late Algorithm

The ML-EM algorithm can be changed into a MAP (green's one-step late algorithm, i.e., Bayesian) algorithm by adding a penalty term in the denominator:

$$x_j^{next} = \frac{x_j^{current}}{\sum_i a_{ij} + \beta \frac{\partial U(X^{current})}{\partial x_j^{current}}} \sum_i a_{ij} \frac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{current}}, \tag{6.9.33}$$

where $U(X)$ is the energy function defined in Section 6.8, and β is a control parameter. This is not a true MAP algorithm because the energy function U is supposed to be evaluated using the next estimate X^{next}, which is not yet available, hence the term "one step late."

6.9.6 Matched and Unmatched Projector/Backprojector Pairs

This topic is quite controversial. It is almost like a religion in which people have their own opinions and carry out their practice accordingly. By projector/backprojector pair, we normally mean matrices A and A^T as described earlier in this chapter. If the backprojection matrix is the transpose of the projection matrix, then this pair is called matched. Otherwise, the pair is called unmatched.

In an analytic reconstruction algorithm, the image pixel is a point. The image is not discretized. The projector, even not used in the reconstruction algorithm, is an integral of the continuous image. Its matched backprojector treats an image pixel as a point, too. Sometimes this backprojector is referred

to as the pixel-driven backprojector. When implementing a pixel-driven backprojector, we start with an image pixel location, and find the location on the detector to get the backprojection value.

In an iterative reconstruction algorithm, the image pixel is no longer a point, but an area. The projector draws a ray from each detector bin, determines the contribution of each pixel according to overlap of this ray with the pixel of interest. This projector is sometimes referred to as the ray-driven projector. Its matched backprojector is the ray-driven backprojector.

We all agree that one should make the projector (i.e., the matrix A) to model the imaging geometry and imaging physics as accurately as possible, significantly reducing the deterministic modeling errors. The question is whether one is allowed to use unmatched projector/backprojector pairs. We have seen many cases from the practical implementations. In some cases, the unmatched pair gives better results than the matched pair, in terms of artifact removal and speedup in computation time. In other cases, the unmatched pair gives more artifacts than the matched pair. A common reason one would use a backprojector other than A^T is to save computation time. We need to be cautioned that the solution can be different if an unmatched pair is used to replace the matched pair in an iterative reconstruction algorithm.

First of all, one cannot just arbitrarily pick up a backprojector and use it in an iterative algorithm. For example, one cannot use a fan-beam backprojector to reconstruct a parallel-beam image. The minimum requirement of choosing a backprojector is that the projection-then-backprojection operator can only blur the image; it cannot cause image distortion in shape and other motions such as rotation and translation. If a projector/backprojector pair is applied to a point source, the result can only be a blurred point source at the same location.

Second of all, we will investigate how a solution can be changed if a different backprojector is used to solve the system $AX{=}P$. Let us consider a modified Landweber iterative algorithm with a backprojector B^{T}, where B^{T} may not be the same as A^{T}, as

$$X^{next} = X^{current} + SB^{\mathrm{T}}(P - AX^{current}), \tag{6.9.34}$$

where $S = diag\{s_1, s_2, \ldots, s_n\}$ with positive diagonal elements controlling the step size of each iteration for each unknown pixel x_j. Let matrix T be

$$T = I - SB^{\mathrm{T}}A, \tag{6.9.35}$$

with I being the identity matrix. Then this modified Landweber algorithm has a general expression for each iteration as

$$X^{(k+1)} = T^{k+1}X^{(0)} + (T^k + \ldots + T + I)SB^{\mathrm{T}}P, \tag{6.9.36}$$

where k is the iteration index, $X^{(k+1)}$ means X^{next}, $X^{(k)}$ means $X^{current}$, $X^{(0)}$ means the initial condition, and so on. This algorithm converges if and

only if $\max_i\{|\lambda_i|\} < 1, i = 1, 2, \ldots, n$, where λ_i is the eigenvalue of the square matrix T. If the eigenvalues of the matrix $B^{\mathrm{T}}A$ are all positive then these convergence conditions can be met by choosing small enough step sizes. After the algorithm converges, the final solution is given by

$$X^{(\infty)} = (B^{\mathrm{T}}A)^{-1}B^{\mathrm{T}}P, \tag{6.9.37}$$

which has a strong dependency on the choice of the backprojector B^{T}. Only in a special case when A^{-1} and B^{-1} exist, the solution

$$X^{(\infty)} = A^{-1}P \tag{6.9.38}$$

is backprojector B^{T} independent. If any one of the eigenvalues of the matrix $B^{\mathrm{T}}A$ is negative, the corresponding eigenvalue of matrix T will be greater than 1 and the algorithm will diverge.

In general, when an unmatched projector/backprojector pair is used in an iterative algorithm, the final solution and the intermediate solutions are backprojector dependent. The matched pair solves the system

$$A^{\mathrm{T}}AX = A^{\mathrm{T}}P, \tag{6.9.39}$$

while the unmatched pair solves a different system

$$B^{\mathrm{T}}AX = B^{\mathrm{T}}P. \tag{6.9.40}$$

In order to speed up the convergence rate of the algorithm, some people include the ramp filter in the backprojector. They treat the combined ramp filter and the backprojector as a new backprojector. The new backprojector first performs ramp filtering then backprojection.

The unmatched pair enforces a different weighting scheme and has a different noise effect than the original problem. The unmatched pair also has a different sampling and data interpolation properties than the original problem. How much these differences can influence the reconstructed image is problem dependent and the users should exercise their judgment to choose a backprojector in their particular problem.

*6.10 Reconstruction Using Highly Undersampled Data with l_0 Minimization

It has been reported that it is possible to exactly reconstruct a 256×256 image with only 10 views. It seems magic to us, because more than 100 views are normally required in such an imaging problem. Let us explore how they do it. The setup is quite simple:

$$\min_X \|\psi X\|_0 \quad \text{subject to} \quad AX = P \tag{6.10.1}$$

where $\|\bullet\|_0$ is the zero quasi-norm and ψ is a sparsifying transform. The zero quasi-norm $\|\bullet\|_0$ is easy to understand; it simply counts the non-zero elements in a vector. For example, $\boldsymbol{v} = [3, 0, 0, 1, 7]$, in which there are 3 non-zero elements; therefore, $\|\boldsymbol{v}\|_0 = 3$.

The trick of this method is to design the sparsifying transform ψ. This transform can be anything as long as it transforms your regular image into a sparse image in which most (say, 97%) pixels are zero.

If your regular image is a piecewise-constant image, you can use the $\|\nabla\|$ (i.e., the magnitude of the gradient) as your sparsifying transform ψ (see Figure 6.21).

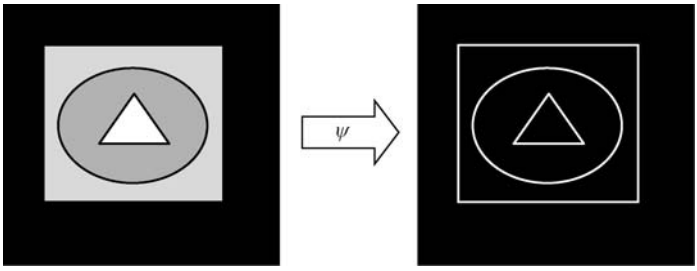


Fig. 6.21. A sparsifying transform extracts some essential information from the original image to produce a spars image.

A practical medical image is not a piecewise-constant image. If you take the gradient, the resultant image is not very sparse. Thus, it is tricky to find the sparsifying transform ψ for your application.

Another problem is that the l_0-minimization is difficult to do. To be honest, we do not have an effective way to find its minimum. We normally set up an objective function using the l_2-norm, which basically gives you a user-friendly least-squares function such as $\sum_i v_i^2$, and we can perform many fun things with it (e.g., taking a derivative). However, the l_2-norm does not work well when data are highly undersampled.

The l_1-norm $\left\{\sum_i |v_i|\right\}$ is a popular alternative. It does not perform as well as the l_0 quasi norm in selecting the optimum image, but it is better than the l_2-norm. On the other hand, it is not as user-friendly as the l_2-norm but is much better than the l_0 quasi norm.

Let us illustrate the differences among the l_2-norm, l_1-norm, and l_0-quasinorm. A norm is a sort of measure of the distance or length. Let us consider a point in the 2D x-y coordinate system with coordinates of (3, 4). We want to find its "distance" from the origin (0, 0). If we use the l_2-norm, the l_2-distance is given as $\sqrt{3^2 + 4^2} = 5$. If we use the l_1-norm, the

l_1-distance is given as $|3|+|4| = 7$. If we use the l_0-quasinorm, the l_0-distance is given as 1+1=2.

Let us consider a different point in the 2D x-y coordinate system which has coordinates of (0, 4). We want to find its "distance" from the origin (0, 0). If we use the l_2-norm, the l_2-distance is given as $\sqrt{0^2+4^2} = 4$. If we use the l_1-norm, the l_1-distance is given as $|0| + |4| = 4$. If we use the l_0-quasinorm, the l_0-distance is given as 0+1=1.

Figure 6.22 shows the unit circles using the l_2-norm, l_1-norm, and l_0-quasinorm, respectively. Except for the l_2-norm unit circle, the other two unit circles do not look like circles at all. For a "measure" to be qualified as a norm, it needs to satisfy a set of axioms. A quasi-norm is almost a norm except that it does not satisfy an axiom called the triangle inequality.

In reality, the data are inconsistent due to random noise and deterministic modeling errors. Thus, it is improper to use $AX = P$ as a constraint in an optimization problem. When using $AX = P$ as a constraint does not make any sense, it is still all right to use the Bayesian method to set up an objective function as discussed in Section 6.8. All we have to do here is to replace the energy function $U(X)$ in Section 6.8 by a norm of a sparsifying transform of the image X. If the derivative of this norm of the sparsifying transform exists, we can use the one-step late algorithm introduced in Section 6.9.5 to reconstruct the image.

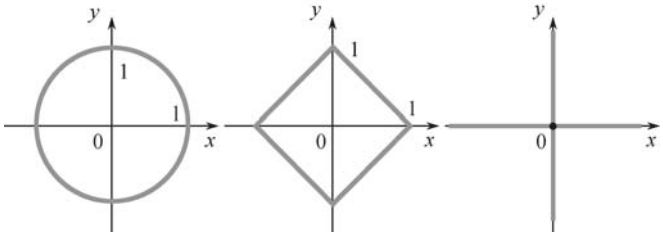


Fig. 6.22. A unit circle is a trajectory of the points that have a distance 1 from the origin. Left: l_2-norm's unit circle. Middle: l_1-norm's unit circle. Right: l_0-quasinorm's unit circle.

You may have also heard of the total variation minimization. The TV (total variation) norm of an image $v(x, y)$ is the integral of the l_1-norm of the gradient $\nabla v(x, y)$, that is,

$$TV(v) = \iint \|\nabla v(x,y)\|_1 \, \mathrm{d}x\mathrm{d}y = \iint \sqrt{\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2} \, \mathrm{d}x\mathrm{d}y. \quad (6.10.2)$$

Therefore, TV norm minimization is the same as the l_1-norm minimization of the gradient image, and it enforces a flat image with the gradient being zero in most places. The resultant image tends to be piecewise constant. In

practice, a small number ε is introduced to calculate the TV-norm as

$$TV(v) = \iint \|\nabla v(x,y)\|_1 \,\mathrm{d}x\mathrm{d}y = \iint \sqrt{\left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \varepsilon^2}\,\mathrm{d}x\mathrm{d}y. \tag{6.10.3}$$

so that the TV norm is differentiable. Using Green's one-step-late method presented in Section 6.9.5, a total variation regulated EM (TV-EM) algorithm can be obtained as

$$x_j^{next} = \frac{x_j^{current}}{\sum_i a_{ij} + \beta \dfrac{\partial TV(X^{current})}{\partial x_j^{current}}} \sum_i a_{ij} \frac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{current}}. \tag{6.10.4}$$

For a 2D image X, if we express each pixel with double indices as $x_{k,l}$, then the partial derivative in the above TV-EM algorithm can be given as

$$\begin{aligned}\frac{\partial TV(X)}{\partial x_{k,l}} = &\frac{x_{k,l} - x_{k-1,l}}{\sqrt{(x_{k,l} - x_{k-1,l})^2 + (x_{k-1,l+1} - x_{k-1,l})^2 + \varepsilon^2}} \\ &+ \frac{x_{k,l} - x_{k,l-1}}{\sqrt{(x_{k+1,l-1} - x_{k,l-1})^2 + (x_{k,l} - x_{k,l-1})^2 + \varepsilon^2}} \\ &- \frac{x_{k+1,l} + x_{k,l+1} - 2x_{k,l}}{\sqrt{(x_{k+1,l} - x_{k,l})^2 + (x_{k,l+1} - x_{k,l})^2 + \varepsilon^2}}.\end{aligned} \tag{6.10.5}$$

In theory, the l_0 quasi-norm is the best measure to promote the sparsity of ψX and should be used. However, the optimization procedure is not tractable for any l_p norm with $0 \leqslant p < 1$, because its associated objective function is not convex any more. The closest norm that produces a convex objective function is the l_1 norm. Fortunately, the l_1 norm minimization is nearly optimal, in the sense that its solution is not far from the solution with the much more complicated l_0 quasi-norm minimization.

Reconstruction with highly-undersampled data is a Bayesian reconstruction problem. Performing the sparsifying transform ψX is one way to extract the prior knowledge from the image X. If other prior information about the image X is available, it should be included in the objective function as well.

6.11 Worked Examples

Example 1 For an arbitrary matrix A, its generalized inverse matrix A^+ must satisfy the following four properties:

$$AA^+A = A, \tag{6.11.1}$$

$$A^+AA^+ = A^+, \tag{6.11.2}$$

$$(A^+A)^* = A^+A, \tag{6.11.3}$$

$$(AA^+)^* = AA^+. \tag{6.11.4}$$

Here M^* is the Hermitian transpose (also called conjugate transpose) of a matrix M. For a real matrix, it is simply the transpose. In the following, we assume that matrix A is real.

Please verify that if $(A^{\mathrm{T}}A)^{-1}$ exits, then $A^+ = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ is a generalized inverse matrix of A. Please also verify that if $(AA^{\mathrm{T}})^{-1}$ exits, then $A^+ = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}$ is a generalized inverse matrix of A.

Proof

Case 1: If $(A^{\mathrm{T}}A)^{-1}$ exits, $A^+ = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$.
Property 1:

$$\text{Left} = AA^+A = A\left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}A = A\left(A^{\mathrm{T}}A\right)^{-1}\left(A^{\mathrm{T}}A\right) = A = \text{Right} \tag{6.11.5}$$

Property 2:

$$\begin{aligned}\text{Left} = A^+AA^+ &= \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}A\left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}} \\ &= \left(A^{\mathrm{T}}A\right)^{-1}\left(A^{\mathrm{T}}A\right)\left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}} \\ &= \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}} = A^+ = \text{Right}\end{aligned} \tag{6.11.6}$$

Property 3:

$$\text{Left} = \left(A^+A\right)^{\mathrm{T}} = \left(\left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}A\right)^{\mathrm{T}} = \left(\left(A^{\mathrm{T}}A\right)^{-1}\left(A^{\mathrm{T}}A\right)\right)^{\mathrm{T}} = I \tag{6.11.7}$$

$$\text{Right} = A^+A = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}A = \left(A^{\mathrm{T}}A\right)^{-1}\left(A^{\mathrm{T}}A\right) = I \tag{6.11.8}$$

Property 4:

$$\begin{aligned}\text{Left} = \left(AA^+\right)^{\mathrm{T}} &= \left(A\left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\right)^{\mathrm{T}} = A\left(\left(A^{\mathrm{T}}A\right)^{-1}\right)^{\mathrm{T}}A^{\mathrm{T}} \\ &= A\left(\left(A^{\mathrm{T}}A\right)^{\mathrm{T}}\right)^{-1}A^{\mathrm{T}} = A\left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}} = AA^+ = \text{Right}\end{aligned} \tag{6.11.9}$$

Case 2: If $(AA^{\mathrm{T}})^{-1}$ exits, $A^+ = A^{\mathrm{T}}(AA^{\mathrm{T}})^{-1}$.
Property 1:

$$\text{Left} = AA^+A = AA^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}A = \left(AA^{\mathrm{T}}\right)\left(AA^{\mathrm{T}}\right)^{-1}A = A = \text{Right} \tag{6.11.10}$$

Property 2:

$$\begin{aligned}\text{Left} = A^+AA^+ &= A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}AA^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1} \\ &= A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}\left(AA^{\mathrm{T}}\right)\left(AA^{\mathrm{T}}\right)^{-1} \\ &= A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1} = A^+ = \text{Right}\end{aligned} \tag{6.11.11}$$

Property 3:

$$\begin{aligned}\text{Left} &= \left(A^{+}A\right)^{\mathrm{T}} = \left(A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}A\right)^{\mathrm{T}} = A^{\mathrm{T}}\left(\left(AA^{\mathrm{T}}\right)^{-1}\right)^{\mathrm{T}}A \\ &= A^{\mathrm{T}}\left(\left(AA^{\mathrm{T}}\right)^{\mathrm{T}}\right)^{-1}A = A^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}A = A^{+}A = \text{Right} \quad (6.11.12)\end{aligned}$$

Property 4:

$$\begin{aligned}\text{Left} &= \left(AA^{+}\right)^{\mathrm{T}} = \left(AA^{\mathrm{T}}\left(AA^{\mathrm{T}}\right)^{-1}\right)^{\mathrm{T}} \\ &= \left(\left(AA^{\mathrm{T}}\right)\left(AA^{\mathrm{T}}\right)^{-1}\right)^{\mathrm{T}} = I \quad (6.11.13)\end{aligned}$$

$$\text{Right} = AA^{+} = AA^{\mathrm{T}}\left({}^{\mathrm{T}}A\right)^{-1} = \left(AA^{\mathrm{T}}\right)\left(A^{\mathrm{T}}A\right)^{-1} = I \quad (6.11.14)$$

Example 2 In Section 6.1, we used SVD to decompose matrix $A_{m\times n}$ into $A_{m\times n} = U_{m\times m}\Sigma V^{\mathrm{T}}_{n\times n}$ and defined $A^{+} = V\Sigma^{+}U^{\mathrm{T}}$ (assuming $\sigma_r > 0$) with $\Sigma^{+}_{n\times m} = \begin{bmatrix} D_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and $D_r = diag\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right\}$. Is $A^{+} = V\Sigma^{+}U^{\mathrm{T}}$ a generalized inverse of matrix A according the four properties stated in Example 1? Find the condition under which $A^{+} = V\Sigma^{+}U^{\mathrm{T}}$ is a generalized inverse of matrix A.

Solution

Let us apply the first property to $A^{+} = V\Sigma^{+}U^{\mathrm{T}}$ and $A_{m\times n} = U_{m\times m}\Sigma V^{\mathrm{T}}_{n\times n}$:

$$\begin{aligned}AA^{+}A &= U\Sigma V^{\mathrm{T}}V\Sigma^{+}U^{\mathrm{T}}U\Sigma V^{\mathrm{T}} = U\Sigma\left(V^{\mathrm{T}}V\right)\Sigma^{+}\left(U^{\mathrm{T}}U\right)\Sigma V^{\mathrm{T}} \\ &= U\Sigma\Sigma^{+}\Sigma V^{\mathrm{T}}. \quad (6.11.15)\end{aligned}$$

In order to have $AA^{+}A = A^{+}$, we must have

$$\Sigma\Sigma^{+}\Sigma = \Sigma, \quad (6.11.16)$$

which is equivalent to

$$\begin{aligned}&diag\{\sigma_1, \sigma_2, \ldots, \sigma_r, \ldots, \sigma_m\} diag\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right\} \\ &\quad \times diag\{\sigma_1, \sigma_2, \ldots, \sigma_r, \ldots, \sigma_m\} \\ &= diag\{\sigma_1, \sigma_2, \ldots, \sigma_r, \ldots, \sigma_m\} \quad (6.11.17)\end{aligned}$$

or

$$\sigma_{r+1} = \ldots = \sigma_m = 0. \quad (6.11.18)$$

Since

$$\sigma_1 \geqslant \sigma_2 \geqslant \ldots \geqslant \sigma_i \geqslant \ldots \geqslant 0, \quad (6.11.19)$$

Property 1 is equivalent to

$$\sigma_{r+1} = 0 \text{ and } \sigma_r > 0. \tag{6.11.20}$$

Now let us look at Property 2:

$$A^+AA^+ = A^+ \tag{6.11.21}$$

which is equivalent to

$$diag\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right\} = diag\left\{\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \ldots, \frac{1}{\sigma_r}, 0, \ldots, 0\right\}. \tag{6.11.22}$$

This property is always satisfied.

Property 3 $(A^+A)^* = A^+A$ and Property 4 $(AA^+)^* = AA^+$both imply that

$$diag\{1, 1, \ldots, 1, 0, \ldots, 0\} = diag\{1, 1, \ldots, 1, 0, \ldots, 0\}, \tag{6.11.23}$$

which is always satisfied.

To summarize the above discussion, the condition under which $A^+ = V\Sigma^+U^{\mathrm{T}}$ is a generalized inverse of matrix A is

$$\sigma_{r+1} = 0 \text{ and } \sigma_r > 0. \tag{6.11.24}$$

However, in practice, a cut-off index much smaller than this r is used to obtain a stable solution.

Example 3 None of the following images match the measured projections. Which one is the best solution among the three solutions? (Compare the χ^2)

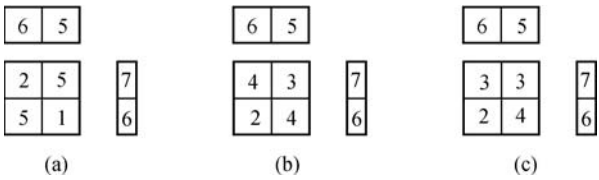


Fig. 6.23. Three images try to match the projections.

Solution

(1) $\chi^2 = 0^2 + 0^2 + 1^2 + 1^2 = 2, (\leftarrow Best)$
(2) $\chi^2 = 0^2 + 0^2 + 2^2 + 0^2 = 4,$
(3) $\chi^2 = 1^2 + 0^2 + 1^2 + 2^2 = 6.$

Example 4 Find a least-squares reconstruction to the imaging problem shown in Figure 6.24.

Solution

The imaging matrix A for this problem $AX = P$ is given as

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{6.11.25}$$

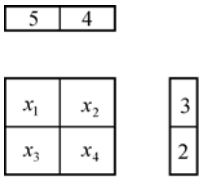


Fig. 6.24. A 2×2 image reconstruction problem.

The rank of A is 3. The system $AX = P$ is also not consistent. Therefore, there is no solution for this problem. We can use a singular value decomposition (SVD) based method to find a pseudo solution. Using Matlab with X=*pinv*$(A) * P$, we get $x_1 = 2.25, x_2 = 1.75, x_3 = 1.75$, and $x_4 = 1.25$.

Example 5 Let $A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$. Find the conjugate direction $\boldsymbol{u}_1$ of $\boldsymbol{u}_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

Solution

The conjugate direction is defined by this relationship: $\boldsymbol{u}_0 \cdot (A^{\mathrm{T}}A)\boldsymbol{u}_1 = 0$, where

$$A^{\mathrm{T}}A = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}. \tag{6.11.26}$$

We also know

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 0. \tag{6.11.27}$$

Thus $\begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, which leads to

$$\boldsymbol{u}_1 = \begin{bmatrix} 1/4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1/4 \\ -1 \end{bmatrix}. \tag{6.11.28}$$

(If you want, you could normalize $\boldsymbol{u}_1$ and make it a unit vector.)

The matrix $A^{\mathrm{T}}A$ defines a quadratic form, which is an ellipse in our case. The point of this example is as follows: For any initial direction $\boldsymbol{u}_0$, draw an ellipse specified by the quadratic form $A^{\mathrm{T}}A$ such that $\boldsymbol{u}_0$ is a tangential direction. If you travel along the conjugate direction $\boldsymbol{u}_1$, you will reach the center of the ellipse (see Figure 6.25).

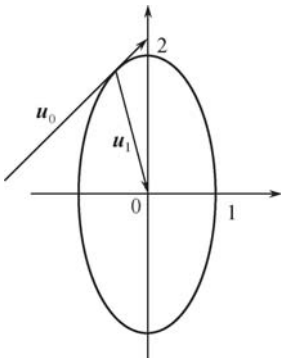


Fig. 6.25. Conjugate directions.

Example 6 Compute one iteration of the ML-EM algorithm with the initial image and projections given below:

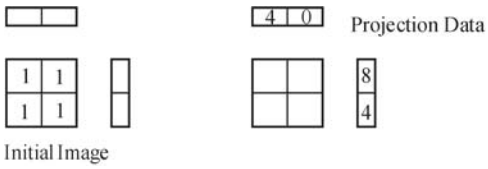
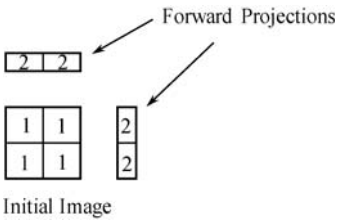


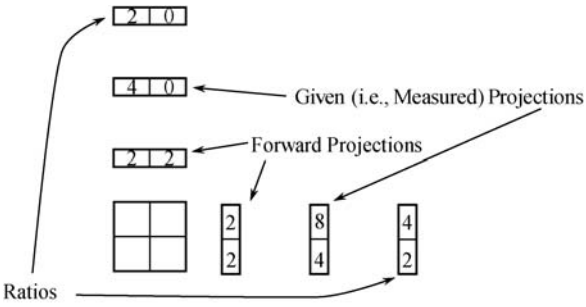
Fig. 6.26. Initial image and projection data for Example 6.

Solution

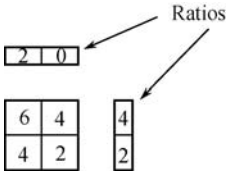
Step 1: find the forward projections.



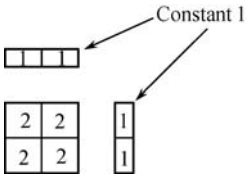
Step 2: find the ratios of given (i.e., measured) projections and the forward projections of the current image.



Step 3: backproject the ratios.



Step 4: backproject a constant 1 from all rays.



Step 5: find pixel-by-pixel ratio of the backprojected image obtained from Step 3 and the backprojected image from Step 4.

$$\frac{\begin{array}{|c|c|}\hline 6 & 4 \\ \hline 4 & 2 \\ \hline \end{array}}{\begin{array}{|c|c|}\hline 2 & 2 \\ \hline 2 & 2 \\ \hline \end{array}} = \begin{array}{|c|c|}\hline 3 & 2 \\ \hline 2 & 1 \\ \hline \end{array}$$

Step 6: pixel-by-pixel update the current image (using point-by-point multiplication, not matrix multiplication).

$$\begin{array}{|c|c|}\hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} \times \begin{array}{|c|c|}\hline 3 & 2 \\ \hline 2 & 1 \\ \hline \end{array} = \begin{array}{|c|c|}\hline 3 & 2 \\ \hline 2 & 1 \\ \hline \end{array}$$

Example 7 Use a Bayesian method to find a stable solution of the

system

$$\begin{cases} x_1 + 0.01x_2 = 1.2, \\ x_1 + 0.001x_2 = 1. \end{cases} \tag{6.11.29}$$

Solution

If we consider the potential measurement errors, the system of equations can be written as

$$\begin{cases} x_1 + 0.01x_2 = 1.2 + \delta_1, \\ x_1 + 0.001x_2 = 1 + \delta_2. \end{cases} \tag{6.11.30}$$

The solution of this modified system is given by

$$\begin{cases} x_1 = 0.978 - 0.111\delta_1 + 1.11\delta_2, \\ x_2 = 22.22 + 111.1\delta_1 - 111.1\delta_2. \end{cases} \tag{6.11.31}$$

It is seen that x_2 is sensitive to measurement noise. Let us assume *a priori* that "x_1 and x_2 are close" and solve this problem using the Bayesian method. Here, a prior is a probability distribution representing knowledge or belief about an unknown quantity *a priori*, that is, before any data have been observed. First, we set up an objective function and use the prior knowledge as a penalty term:

$$\begin{aligned} F(x_1, x_2) = &(x_1 + 0.01x_2 - 1.2 - \delta_1)^2 + (x_1 + 0.001x_2 - 1 - \delta_2)^2 \\ &+\beta(x_1 - x_2)^2. \end{aligned} \tag{6.11.32}$$

Note that β in this expression is not a Lagrange multiplier, but a pre-assigned constant. To minimize $F(x_1, x_2)$, we set $\partial F/\partial x_1 = 0$ and $\partial F/\partial x_2 = 0$. This results in a *different* problem:

$$\begin{aligned} &\begin{bmatrix} 2+\beta & 0.011-\beta \\ 0.011-\beta & 0.01^2 + 0.001^2 + \beta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &\quad = \begin{bmatrix} (1.2+\delta_1) + (1+\delta_2) \\ (0.01)(1.2+\delta_1) + (0.001)(1+\delta_2) \end{bmatrix}. \end{aligned} \tag{6.11.33}$$

The solution of this system depends on the value of β, as well as the values of δ_1 and δ_2. Some MAP solutions are listed in the following two tables:

Case 1: Noiseless ($\delta_1 = \delta_2 = 0$)

β	0	0.01	0.1	1	10	100
Condition #	50000	200	22	5.7	22	200
x_1	0.978	1.094	1.094	1.094	1.094	1.094
x_2	22.222	1.178	1.103	1.095	1.094	1.094

Case 2: Noisy ($\delta_1 = -\delta_2 = 0.2$)

β	0	0.01	0.1	1	10	100
Condition #	50000	200	22	5.7	22	200
x_1	0.733	1.094	1.095	1.095	1.095	1.095
x_2	66.667	1.357	1.122	1.098	1.096	1.095

The system becomes more stable with an additional prior term in the objective function. The stability is reflected by much smaller condition numbers. This example also tells us that using a prior changes the original problem. Even for noiseless data, the MAP solutions may be different from the true solution. When you plan to use the Bayesian method, be careful and be sure that your prior knowledge is reasonable.

Example 8 Show that for the ML-EM algorithm at each iteration, the total number of the counts of the forward projections is the same as the total number of the counts of the original projection data:

$$\sum_i \left(\sum_j a_{ij} x_j \right) = \sum_i p_i. \tag{6.11.34}$$

Proof

$$\begin{aligned}
&\sum_i \sum_j a_{ij} x_j \\
&= \sum_i \sum_j a_{ij} \left(\frac{x_j^{old}}{\sum_n a_{nj}} \sum_n a_{nj} \frac{p_n}{\sum_k a_{nk} x_k^{old}} \right) \\
&= \sum_j x_j^{old} \frac{\sum_j a_{ij}}{\sum_n a_{nj}} \sum_n a_{nj} \frac{p_n}{\sum_k a_{nk} x_k^{old}} \quad \text{[Change the order of summations.]} \\
&= \sum_j x_j^{old} \sum_n a_{nj} \frac{p_n}{\sum_k a_{nk} x_k^{old}} \quad \because \left[\sum_j a_{ij} = \sum_n a_{nj} \right] \\
&= \sum_n p_n \frac{\sum_j a_{nj} x_j^{old}}{\sum_k a_{nk} x_k^{old}} \quad \text{[Change the order of summations.]} \\
&= \sum_n p_n. \qquad (6.11.35)
\end{aligned}$$

Example 9 Use a computer simulation to demonstrate that the resolution recover rate is location dependent in the ML-EM algorithm.

Solution

A 2D computer-generated phantom shown in Figure 6.27 is used to generate projection data with and without Poisson noise. In data generation, attenuation, system blurring, and photon scattering are not included. That is, the projections are ideal line integrals of the object. The image size is 256×256, and 256 views uniformly distributed over 360° are used for projection data generation.

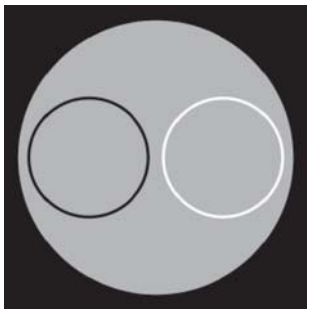


Fig. 6.27. The true image (a mathematical phantom).

The iterative ML-EM algorithm is used for image reconstruction. Two reconstructed images are shown in Figure 6.28, one of which is obtained after 25 iterations, and the other is obtained after 250 iterations.

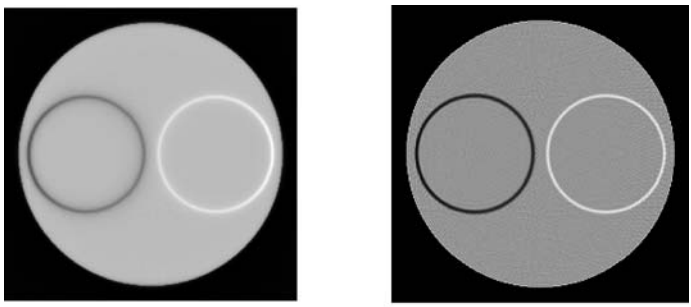


Fig. 6.28. Iterative ML-EM reconstructions.

After 250 iterations, the algorithm has almost converged, and the resolution throughout the image is uniform. When the algorithm is stopped early at the 25^{th} iteration, the resolution is not uniformly recovered. Higher resolution can be observed at the edge of the object. The resolution recovery rate

is slower at the center. The motivation of early stopping is to regulate the noise. This is demonstrated with the noisy data reconstructions as shown in Figure 6.29, where two images are displayed. One of the images is obtained after 25 iterations and the other is obtained after 250 iterations.

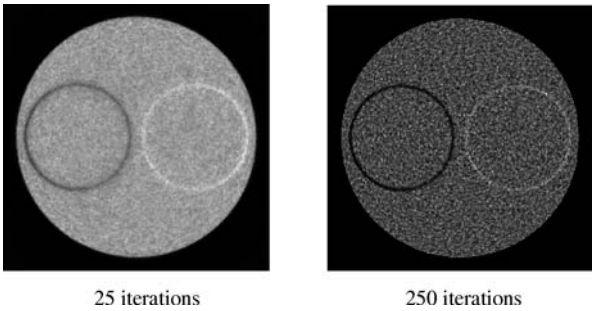


Fig. 6.29. Iterative ML-EM reconstructions with the same noisy projection data.

The image with 250 iterations is noisier than that with 25 iterations. If we apply a low-pass filter to the noisy image obtained with 250 iterations, the image becomes less noisy and still maintains the uniform resolution throughout the image (see Figure 6.30). Therefore, it is a good strategy to over-iterate and then to apply a low-pass filter to control noise.

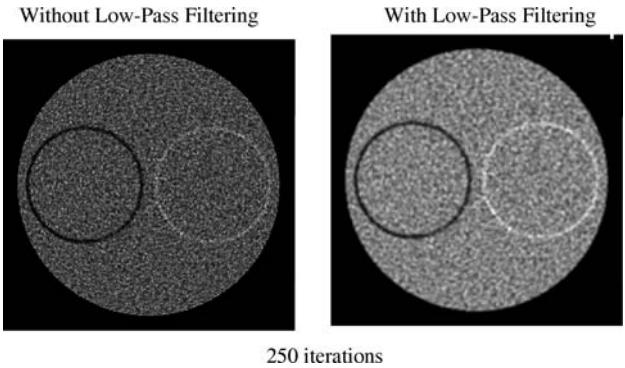


Fig. 6.30. Applying a low-pass filter to the image reconstructed with 250 iterations of ML-EM results in uniform resolution and reduced noise.

On the other hand, if the projections are ideal line-integrals, the 2D parallel-beam FPB algorithm can provide images with uniform resolution (see Figure 6.31). However, if the imaging system has a spatially variant resolution and sensitivity, the current FBP algorithms are unable to model the system accurately and cannot provide uniform resolution in their reconstructions.

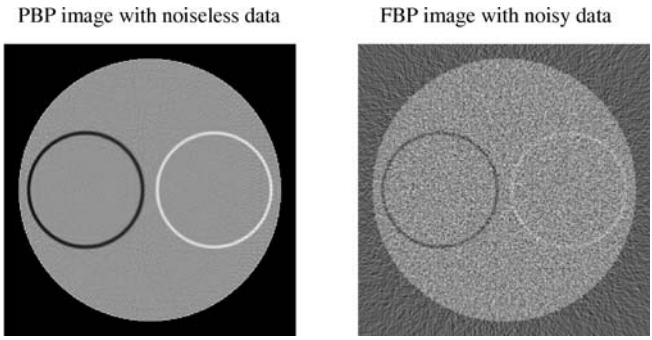


Fig. 6.31. Filtered backprojection reconstructions with noise-less and noisy data.

6.12 Summary

- The main difference between an analytic image reconstruction algorithm and an iterative image reconstruction algorithm is in image modeling. In an analytic algorithm, the image is assumed to be continuous, and each image pixel is a point. The set of discrete pixels is for display purpose. We can make those display points any way we want. However, in an iterative algorithm, a pixel is an area, which is used to form the projections of the current estimate of the image. The pixel model can significantly affect the quality of the reconstructed image.
- Another difference between an analytic image reconstruction algorithm and an iterative image reconstruction algorithm is that the analytic algorithm tries to solve an integral equation, while the iterative algorithm tries to solve a system of linear equations.
- A system of linear equations is easier to solve than an integral equation. This allows the linear equations to model more realistic and more complex imaging geometry and imaging physics. In other words, the iterative algorithm can solve a more realistic imaging problem than an analytic algorithm. As a result, the iterative algorithm usually provides a more accurate reconstruction.
- Iterative algorithms are used to minimize an objective function. This objective function can effectively incorporate the noise in the measurement. Currently, analytic algorithms cannot model noise, and its noise control is achieved by frequency windowing.
- The iterative ML-EM and OS-EM algorithms are most popular in emission tomography image reconstruction. They assume Poisson noise statistics.
- Even though noise is modeled in the objective function, the reconstructed

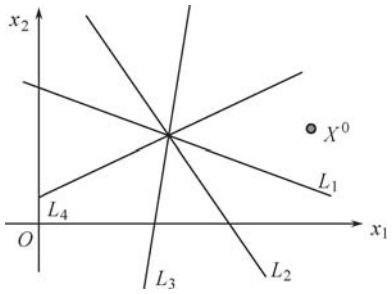
image is still noisy. There are five methods to control noise.

- The first method is to stop the iterative algorithm early, before it converges. This simple method has a draw back that it may result in an image with non-uniform resolution. One remedy is to iterate till convergence and apply a post lowpass filter to suppress the noise.
- The second method is to replace the flat, non-overlapping pixels by smooth, overlapping pixels to represent the image. The method can remove the artificially introduced high frequency components by the flat, non-overlapping pixels in the image. A drawback of using smooth, overlapping pixels is the increased computational complexity. One remedy is to use the traditional flat, non-overlapping pixels and apply a lowpass filter to backprojected images.
- The third method is to model more accurate imaging geometry and physics in the projector/backprojector pair. The aim of this method is to reduce the deterministic discrepancy between the projection model and the measured data.
- The fourth method is to use the correct noise model to set up an objective function. The author's personal belief is that the noise model that is based on the joint probability density function is not very critical. You can have a wrong noise model (i.e., a wrong joint probability density function), but you must have the accurate variance. The important part is to incorporate a correct measurement noise variance to weigh the data. It is not as important to worry about whether the noise is strictly Gaussian, strictly Poisson, and so on.
- The fifth method is the use of prior knowledge. If the projection data do not carry enough information about the object, due partially to insufficient measurements and partially to noise, prior knowledge about the object can supplement more information about the object and make the iterative algorithm more stable. Be careful that if the prior knowledge is not really true, it can mislead the algorithm to converge to a wrong image.
- The readers are expected to understand how the system of imaging equations is set up and how the iterative ML-EM algorithm works in this chapter.

Problems

Problem 6.1 Some iterative algorithms, for example, the ART and OS-EM algorithms, update the image very frequently. For those algorithms, the processing order of the data subsets is important. In this problem, we use the ART algorithm to graphically solve a system of linear equations

$\{L_1, L_2, L_3, L_4\}$ with two variables as shown in the figure below. The initial estimated solution is X^0. Solve the system with two different orders: (a) $L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow L_4$ and (b) $L_1 \rightarrow L_3 \rightarrow L_2 \rightarrow L_4$, respectively. Compare their performance in terms of convergence rate.



Problem 6.2 The iterative ML-EM algorithm

$$x_j^{(k+1)} = x_j^{(k)} \frac{\sum_i a_{ij} \dfrac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{(k)}}}{\sum_i a_{ij}},$$

or the iterative OS-EM algorithm, has many modified versions. One of the versions introduces a new parameter h:

$$x_j^{(k+1)} = x_j^{(k)} \left(\frac{\sum_i a_{ij} \dfrac{p_i}{\sum_{\hat{j}} a_{i\hat{j}} x_{\hat{j}}^{(k)}}}{\sum_i a_{ij}} \right)^h.$$

This parameter h usually takes a real value in the interval between 1 and 5. The purpose of using this parameter h is to increase the iteration step size and make the algorithm converge faster. If the parameter h is chosen in the interval between 0 and 1, it reduces the iteration step size. Does this new algorithm satisfy the property of total count conservation as studied in Worked Problem 8 in this chapter? If that property is not satisfied any more, you can always scale the newly updated image with a factor. Find this scaling factor so that the total count is conserved for each iteration.

Problem 6.3 The modified algorithm discussed in Problem 6.2 above can only be used in emission imaging applications, because the linear equations are weighted by Poisson variance of the emission measurements.

One can develop a similar algorithm for the transmission measurements to find the attenuation map of the object as:

$$\mu_j^{(k+1)} = \mu_j^{(k)} \left(\frac{\sum_i a_{ij} \frac{N_0 e^{-\sum_{\hat{j}} a_{i\hat{j}} \mu_{\hat{j}}^{(k)}}}{N_i}}{\sum_i a_{ij}} \right)^h ,$$

where the measurements are modeled as $N_i = N_0 e^{-\sum_j a_{ij}\mu_j}$. Here, we do not take the logarithm and convert these non-linear equations into linear equations. Instead, we go ahead and solve this system of non-linear equations $N_i = N_0 e^{-\sum_j a_{ij}\mu_j}$ directly. Use the Taylor expansion to convert this algorithm to its corresponding the additive updating form. Discuss how the parameter h controls the iteration step size, how the non-linear equations are weighted, and what the weighting quantity is.

Problem 6.4 We have learned that in an iterative image reconstruction algorithm the projector should model the imaging system as accurately as possible. For a set of practical data acquired from an actual imaging system, there is a simple way to verify whethter the modeling in the projector is accurate enough. This method is described as follows. Run an iterative algorithm to reconstruct the image. After the algorithm is converged, calculate and display the data discrepancy images at every view. A discrepancy image is the difference between the projection of the reconstructed image and the measured projection data, or is the ratio of the projection of the reconstructed image to the measured projection data. If the projector models the imaging system accurately, you do not see the shadow of the object in the discrepancy images and you only see the random noise in the discrepancy images. Verify this method with a computer simulation.

References

1. Bai C, Zeng GL, Gullberg GT et al (1998) Slab-by-slab blurring model for geometric point response and attenuation correction using iterative reconstruction algorithms. IEEE Trans Nucl Sci 45: 2168 – 2173
2. Barrett HH, Wilson DW, Tsui BMW (1994) Noise properties of the EM algorithm, I. Theory. Phys Med Biol 39: 833 – 846
3. Beekman F, Kamphuis C, Viergever M (1996) Improved SPECT quantitation using fully three-dimensional iterative spatially variable scatter response compensation. IEEE Trans Med Imaging 15: 491 – 499

4. Byrne CL (1996) block-iterative methods for image reconstruction from projections. IEEE Trans Imaging Processing 5: 792 – 794
5. Candès E, Romberg J, Tao T (2006) Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans Inf Theory 52: 489 – 509
6. Censor Y, Eggermont PPB, Fordon D (1983) Strong underrelaxation in Kaczmarz's method for inconsistent system. Numer Math 41: 83 – 92
7. Chang L (1979) Attenuation and incomplete projection in SPECT. IEEE Trans Nucl Sci 26: 2780 – 2789
8. Chen GH, Tang J, Hsieh (2009) Temporal resolution improvement using PICCS in MDCT cardiac imaging. Med Phys 36: 2130 – 2135
9. Chiao PC, Rogers WL, Fessler JA et al (1994) Model-based estimation with boundary side information or boundary regularization. IEEE Trans Med Imaging 13: 227 – 234
10. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J Royal Stat Soc B 39: 1 – 38
11. DiBella E, Barclay A, Eisner R et al (1996) A comparison of rotation-based methods for iterative reconstruction algorithms. IEEE Trans Nucl Sci 43: 3370 – 3376
12. Donoho D (2006) Compressed sensing. IEEE Trans Inf Theory 52: 1289 – 1306
13. Fessler J (1996) Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography. IEEE Trans Image Processing 5: 493 – 506
14. Fessler J, Hero A (1994) Space-alternating generalized expectation maximization algorithm. IEEE Trans Sig Processing 42: 2664 – 2677
15. Fessler JA, Rogers WL (1996) Spatial resolution properties of penalized-likelihood image reconstruction: Space-invariant tomography. IEEE Trans Imaging Processing 5: 1346 – 1358
16. Frey EC, Ju ZW, Tsui BMW (1993) A fast projector-backprojector pair for modeling the asymmetric spatially-varying scatter response in SPECT imaging. IEEE Trans Nucl Sci 40: 1192 – 1197
17. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. IEEE Trans Patt Anal Mach Intell 6: 721 – 741
18. Gilbert P (1972) Iterative methods for the reconstruction of three dimensional objects from their projections. J Theor Biol 36:105-117
19. Gindi G, Lee M, Rangarajan A et al (1991) Bayesian reconstruction of functional images using registered anatomical images as priors. In: Colchester ACF, Hawkes DJ (eds) Information Processing in Medical Imaging, Springer-Verlag, New York, 121 – 131
20. Gordon R, Bender R, Herman GT (1970) Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. J Theor Biol 29: 471 – 481
21. Hebert TJ, Leahy R (1992) Statistic-based MAP image reconstruction from Poisson data using Gibbs priors. IEEE Trans Sig Processing 40: 2290 – 2303
22. Herman GT (1980) Image Reconstruction from Projections: The Fundamentals of Computerized Tomography, Academic, New York
23. Hsieh YL, Zeng GL, Gullberg GT (1998) Projection space image reconstruction using strip functions to calculate pixels more "natural" for modeling the geometric response of the SPECT collimator. IEEE Trans Med Imaging 17: 24 – 44

24. Hudson HM, Larkin RS (1994) Accelerated image reconstruction using ordered subsets of projection data. IEEE Trans Med Imaging 13: 601 – 609
25. Hwang DS, Zeng GL (2005) A new simple iterative reconstruction algorithm for SPECT transmission measurement. Med Phys 32: 2312 – 2319
26. Hwang DS, Zeng GL (2005) Reduction of noise amplification in SPECT using smaller detector bin size. IEEE Trans Nucl Sci 52: 1417 – 1427
27. Hwang DS, Zeng GL (2006) Convergence study of an accelerated ML-EM algorithm using bigger step size, Phys Med Biol 51: 237 – 252
28. Qi J (2003) A unified noise analysis for iterative image estimation. Phys Med Biol 48: 3505 – 3519
29. Kacmarz S (1937) Angenaherte Aufosung von Systemen linearer Gleichungen. Bull Acad Polon Sci A35: 355 – 357
30. King, Miller TR (1985) Use of a nonstationary temporal Wiener filter in nuclear medicine. Eur J Nucl Med 10: 458 – 461
31. Lalush DS, Tsui BMW (1993) A generalized Gibbs prior for maximum a posteriori reconstruction in SPECT. Phys Med Biol 38: 729 – 741
32. Lalush DS, Tsui BMW (1994) Improving the convergence of iterative filtered backprojection algorithms. Med Phys 21: 1283 – 1286
33. Lange K (1990) Convergence of EM image reconstruction algorithms with Gibbs smoothing. IEEE Trans Med Imaging 9: 439 – 446
34. Lange K, Carson R (1984) EM reconstruction algorithms for emission and transmission tomography. J Comput Assist Tomogr 8: 306 – 316
35. Leng S, Tang J, Zambelli J et al (2008) High temporal resolution and streak-free four-dimensional cone-beam computed tomography. Phys Med Biol 53: 5653 – 5673
36. Lewitt R (1992) Alternatives to voxels for image representation in iterative reconstruction algorithms. Phys Med Biol 37: 705 – 716
37. Lewitt RM, Muehllehner G (1986) Accelerated iterative reconstruction for positron emission tomography based on the EM algorithm for maximum likelihood estimation. IEEE Trans Med Imaging 5: 16 – 22
38. Liang Z (1993) Compensation for attenuation, scatter, and detector response in SPECT reconstruction via iterative FBP methods. Med Phys 20: 1097 – 1106
39. Liang Z, Jazczak R, Greer K (1989) On Bayesian image reconstruction from projections: uniform and nonuniform *a priori* source information. IEEE Trans Med Imaging 8: 227 – 235
40. Liow J, Strother SC (1993) The convergence of object dependent resolution in maximum likelihood based tomographic image reconstruction. Phys Med Biol 38: 55 – 70
41. Lucy LB (1974) An iterative technique for the rectification of observed distribution. Astrophys J 79: 745 – 754
42. Naraynan VM, King MA, Soare E et al (1999) Application pf the Karhunen-Loeve transform to 4D reconstruction of gated cardiac SPECT images. IEEE Trans Nucl Sci 46: 1001 – 1008
43. Nuyts J, Michel C, Dupont P (2001) Maximum-likelihood expectation-maximization reconstruction of sinograms with arbitrary noise distribution using NEC-transformations. IEEE Trans Med Imaging 20: 365 – 375
44. Pan TS, Yagle AE (1991) Numerical study of multigrid implementations of some iterative image reconstruction algorithms. IEEE Trans Med Imaging 10: 572 – 588

45. Pan X, Wong WH, Chen CT et al (1993) Correction for photon attenuation in SPECT: Analytical framework, average attenuation factors, and a new hybrid approach. Phys. Med. Biol. 38: 1219 – 1234
46. Panin VY, Zeng GL, Gullberg GT (1999) Total variation regulated EM algorithm. IEEE Trans Nucl Sci 46,: 2202 – 2210
47. Qi J, Leahy RM (1999) Fast computation of the covariance of MAP reconstructions of PET images. Proc. SPIE 3661: 344 – 355
48. Rucgardson WH (1972) Bayesian-based iterative method of image restoration. J Opt Soc Am A 62: 55 – 59
49. Shepp LA, Vardi Y (1982) Maximum likelihood estimation for emission tomography. IEEE Trans Med Imaging 1: 113 – 121
50. Snyder DL, Miller MI (1985) The use of sieves of stabilize images produced with the EM algorithm for emission tomography. IEEE Trans Nucl Sci 32: 3864 – 3872
51. Tanaka E (1987) A fast reconstruction algorithm for stationary positron emission tomography based on a modified EM algorithm. IEEE Trans Med Imaging 6: 98 – 105
52. Trzasko J, Manduca A (2009) Highly undersampled magnetic resonance image reconstruction via homotopic l_0-minimization. IEEE Trans Med Imaging 28: 106 – 121
53. Wallis JW, Miller TR (1993) Rapidly converging iterative reconstruction algorithms in single-photon emission computed tomography. J Nucl Med 34: 1793 – 1800
54. Wilson DW, Tsui BMW, Barrett HH (1994) Noise properties of the EM algorithm, II. Monte Carlo Simulations. Phys Med Biol 39: 847 – 871
55. Yu H, Wang G (2009) Compressed sensing based interior tomography. Phys Med Biol 54: 2791 – 2805
56. Zeng GL, Gullberg Tsui BMW et al (1991) Three-dimensional iterative reconstruction algorithms with attenuation and geometric point response correction. IEEE Trans Nucl Sci 38: 693 – 702
57. Zeng GL, Hsieh YL, Gullberg GT (1994) A rotating and warping projector-backprojector for fan-beam and cone-beam iterative algorithm. IEEE Trans Nucl Sci 41: 2807 – 2811
58. Zeng GL, Gullberg GT (2000) Unmatched projector/backprojector pairs in an iterative reconstruction algorithm. IEEE Trans Med Imaging 19: 548 – 555
59. Zhang B, Zeng GL (2006) An immediate after-backprojection filtering method with blob-shaped window functions for voxel-based iterative reconstruction. Phys Med Biol 51: 5825 – 5842

7 MRI Reconstruction

The MRI imaging physics is quite different from that of transmission or emission imaging as we discussed in the previous chapters. This chapter first introduces the imaging physics of MRI, showing how the MRI signals are formed. We will see that the MRI signals are in the Fourier domain and the image reconstruction is achieved via the inverse Fourier transform.

7.1 The "M"

MRI stands for magnetic resonance imaging. Its working principle is quite different from that of emission and transmission tomography. MRI is an image of "proton density" in a cross-section of the patient. The data in MRI can be simplified as weighted plane integrals of the proton density function in that plane with "frequency" dependent weighting functions.

In this chapter, we present a watered-down version of the MRI principle. The goal of regular MRI is to get a picture of the distribution of hydrogen atoms (H^+) within the patient body.

The hydrogen atom is simply a proton that carries a positive charge and is continuously spinning. A spinning charge generates a magnetic field around it, as if it were a tiny magnet (see Figure 7.1). We call this tiny magnet a "$\boldsymbol{\mu}$

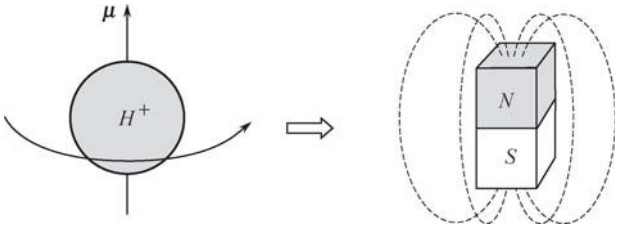


Fig. 7.1. A spinning proton acts like a tiny magnet.

moment," which is a vector. The orientation is important for a vector.

In the absence of a strong external magnetic field, the proton magnetic moments $\boldsymbol{\mu}$ are randomly oriented inside the body. Thus, the *net* magnetic moment is zero $\mathbf{0}$ (see Figure 7.2).

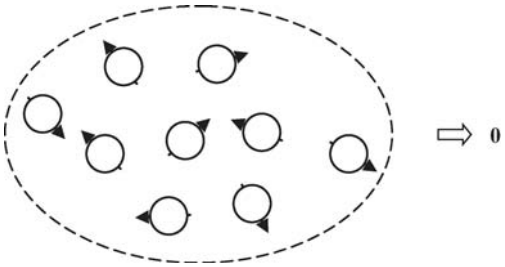


Fig. 7.2. The net magnetic moment is zero without a strong external magnetic field.

When a strong external magnetic field is applied, the net magnetic moment is no longer zero. We denote this net magnetic moment as a vector $\boldsymbol{M}$ (see Figure 7.3). We must point out that even in this case, not all $\boldsymbol{\mu}$ are pointing in the same direction. About half of them point in direction of the magnetic field, and the other half point in the opposite direction. Therefore, the magnitude of $\boldsymbol{M}$ is very small and is proportional to

$$\frac{\gamma \hbar B_0}{2KT} \tag{7.1.1}$$

where T is the absolute temperature of the patient (in °K), K is the Boltzman constant (8.62×10^{-11} MeV/°K), $\hbar$ is the Plank constant ($6.625\ 2\times10^{-27}$ ergs), B_0 is the magnetic field strength (in Tesla), and γ is an atom-dependent constant (42.58 MHz/Tesla for H^+) called the gyromagnetic ratio. We can see that a stronger magnetic field helps generate a stronger signal.

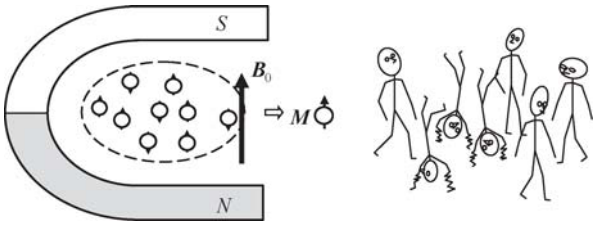


Fig. 7.3. In the strong external magnetic field a small non-zero net magnetization can be observed.

Another concept that we need to explain MRI physics is precession. Let us take a look at a spinning toy top (or a gyroscope). Besides spinning, there

is another motion: the spin axis rotates around the direction of gravity (an external force). This motion of the spin axis rotating about the direction of gravity is referred to as precession (see Figure 7.4). If the top does not spin, precession will not happen; it just falls.

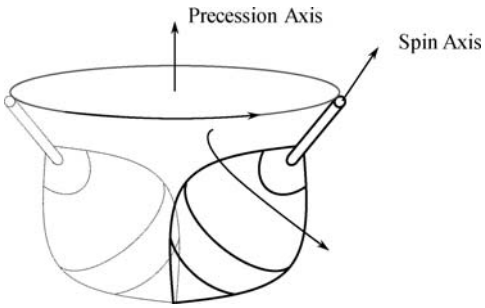


Fig. 7.4. A spinning toy top precesses about a vertical axis due to the gravity.

7.2 The "R"

We have a vector $\boldsymbol{M}$, called net magnetic moment, which is spinning by itself. Normally, the vector $\boldsymbol{M}$ points at the same direction of the external magnetic field $\boldsymbol{B}_0$. If we somehow knock the vector $\boldsymbol{M}$ off balance, and $\boldsymbol{M}$ is not in the direction of $\boldsymbol{M}$ anymore, then the vector $\boldsymbol{M}$ will precess about the direction of $\boldsymbol{B}_0$ just as the toy top precesses about the direction of gravity (see Figure 7.5). The precession frequency is called the Larmor frequency and is given as

$$\omega_0 = \gamma B_0 \tag{7.2.1}$$

where B_0 is the external magnetic field strength, and γ is the gyromagnetic ratio. For a proton, $\gamma = 42.58$ MHz/Tesla. If the MRI machine has a magnetic

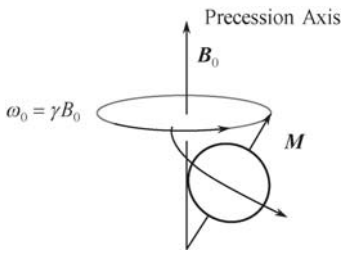


Fig. 7.5. Protons precess at the Larmor frequency ω_0.

field of 1.5 Tesla, then the Larmor frequency is approximately 64 MHz, which is close to the frequency range of an FM radio.

The MRI signal is nothing but this 64 MHz radio frequency electromagnetic wave sent out from the patient body after the net magnetic moment $\boldsymbol{M}$ is somehow knocked off balance. Thus, the MRI signal is also called the RF (radio frequency) signal. The strength of the signal is proportional to the proton density inside the patient.

To knock the net magnetic moment $\boldsymbol{M}$ off balance is not an easy task. In order to move the vector $\boldsymbol{M}$, we need to create a virtual situation so that the vector $\boldsymbol{M}$ does not feel the existence of the external $\boldsymbol{B}_0$ field.

We will first put the vector $\boldsymbol{M}$ on a virtual rotating merry-go-round or a rotating platform (see Figure 7.6), which is rotating at the Larmor frequency. We will later find a way to create precession while the vector $\boldsymbol{M}$ is on that rotating platform. The vector $\boldsymbol{M}$ now is currently standing on the rotating platform upright and is spinning on its own axis; there is no motion relative to the platform. Even if the vector $\boldsymbol{M}$ is not standing upright and it has a non-zero angle with the vertical line, vector $\boldsymbol{M}$ will stay with that angle and will not have any motion relative to the platform.

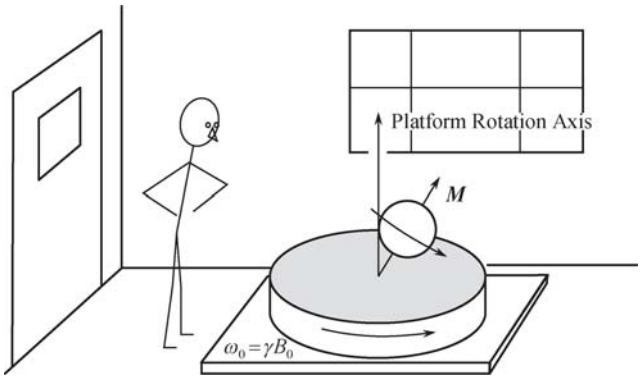


Fig. 7.6. The magnetization vector does not precess relative to the rotating platform.

Next, we step on the rotating platform and apply a new magnetic field $\boldsymbol{B}_1$, orthogonal to the main magnetic field $\boldsymbol{B}_0$. On the rotating platform, the vector $\boldsymbol{M}$ does not feel the existence of the main field $\boldsymbol{B}_0$; it only feels the push from the new field $\boldsymbol{B}_1$. Since $\boldsymbol{M}$ is not aligned with $\boldsymbol{B}_1$, it will precess about the direction of $\boldsymbol{B}_1$ (see Figure 7.7) at a precession frequency $\omega_1 = \gamma B_1$. Once $\boldsymbol{M}$ reaches the platform floor, we turn the new field $\boldsymbol{B}_1$ off. Thus, the mission of knocking off $\boldsymbol{M}$ is accomplished. The $\boldsymbol{B}_1$ field is only turned on very briefly; it is called a 90° RF pulse if it is turned off as $\boldsymbol{M}$

soon as touches the platform floor.

What is this $\boldsymbol{B}_1$ field anyway? The magnetic field $\boldsymbol{B}_1$ is applied on a virtual rotating platform, which rotates at the Larmor frequency, say 64 MHz for a proton in a 1.5 Tesla MRI machine. Therefore, $\boldsymbol{B}_1$ is an alternating electromagnetic RF field with the *same* frequency as the Larmor frequency. Another term for the same frequency is the *resonance* frequency. The $\boldsymbol{B}_1$ RF signal is sent to the patient through an RF coil, which is basically an antenna. The procedure of turning the $\boldsymbol{B}_1$ RF field on and knocking $\boldsymbol{M}$ off balance is called RF excitation.

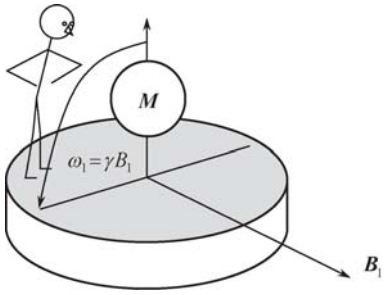


Fig. 7.7. On the rotating platform, the effect of the main field can be ignored. Only the new B_1 field is effective.

After RF excitation, we turn off $\boldsymbol{B}_1$. Now the net magnetic moment $\boldsymbol{M}$ is not in the equilibrium state, but in the excited state. The vector $\boldsymbol{M}$ is precessing about the main field direction $\boldsymbol{B}_0$, and RF signals that contain the patient proton density information are emitted.

The excited state is unstable. After excitation, the vector $\boldsymbol{M}$ then goes through the relaxation period and eventually returns to the original equilibrium state, where the vector $\boldsymbol{M}$ points to the $\boldsymbol{B}_0$ direction (see Figure 7.8). In a Cartesian system, the $\boldsymbol{B}_0$ direction is the z-direction, and the "platform floor" is the x-y plane.

The x and y components of the vector $\boldsymbol{M}$ make up the MRI signal. After $\boldsymbol{M}$ relaxes back to its equilibrium position, both its x and y components are zero; hence, no more signals can be detected. Another RF pulse excitation is needed for further data acquisition. This procedure is repeated over and over again until enough data are acquired for imaging.

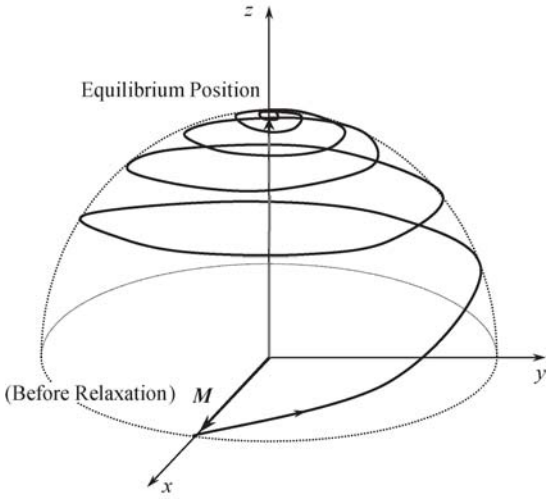


Fig. 7.8. Relaxation of the magnetization vector.

7.3 The "I"

The RF signal emitted by the patient as described in Section 7.2 cannot be used to form an image because it is a combined signal from everywhere. We need a way to code the location information. This is achieved by the gradient coils. In the MRI machine, there are many coils. The main large superconducting coil immersed in liquid helium is used to generate the strong static $\boldsymbol{B}_0$ field. There are RF coils, which are used to emit RF pulses for excitation and to receive RF imaging signals. There are other coils in the machine to create gradients; they are x-gradient coils, y-gradient coils, and z-gradient coils.

7.3.1 To Obtain z-Information — Slice Selection

The z-gradient coils are shown in Figure 7.9. The currents in the two coils are running in the opposite directions and generate the local magnetic fields to enhance and reduce the main field $\boldsymbol{B}_0$, respectively. The resultant magnetic field is still pointing in the z-direction; however, the field strength is stronger at locations with larger z values and is weaker at locations with smaller z

values. These coils create a gradient of the magnetic field strength in the z direction.

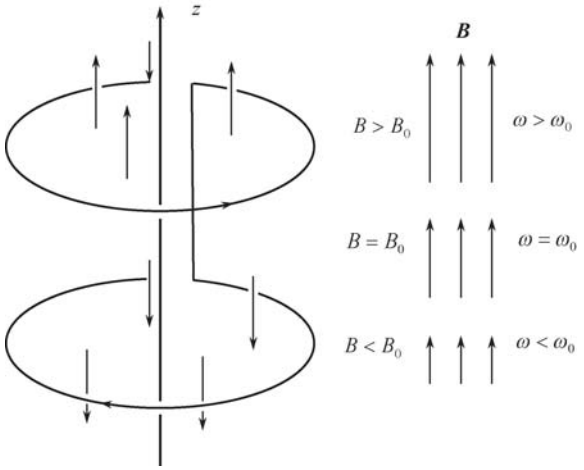


Fig. 7.9. The z-gradient coils create a non-uniform field in the z-direction.

This z-gradient makes a non-uniform magnetic field, which results in different Larmor frequencies ω for a different z position. The Larmor frequency is ω_0 only at one z-slice. We turn on and off the z-gradient and the $\boldsymbol{B}_1$ RF pulse at the same time. The $\boldsymbol{B}_1$ RF pulse is at the frequency of ω_0. According to the resonance frequency principle, only one z-slice of the patient body is affected by the RF pulse. That is, only the protons in this particular z-slice get excited and send out the RF signal (see Figure 7.10).

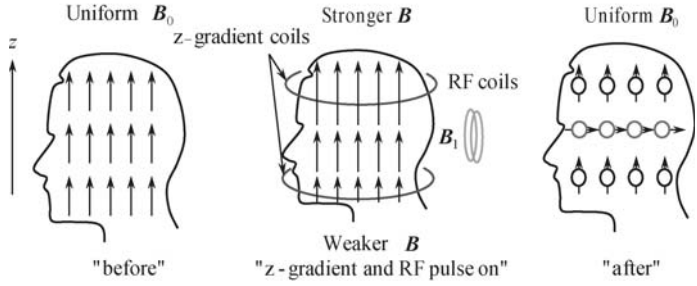


Fig. 7.10. Slice selection is done by the z-gradient and the B_1 field.

7.3.2 To Obtain x-Information — Frequency Encoding

We use the x-gradient to provide the x-position information. When we are ready to receive the RF signal from the patient, we turn on the x-gradient. The x-gradient is generated by the x-gradient coils, and it makes the strength of the main magnetic field vary in the x-direction. The principle of using x-gradient to code x-position is illustrated in Figure 7.11. Using the relation $\omega = \gamma B$, stronger magnetic field B gives the higher frequency. Thus, the x-location can be determined by the received frequency.

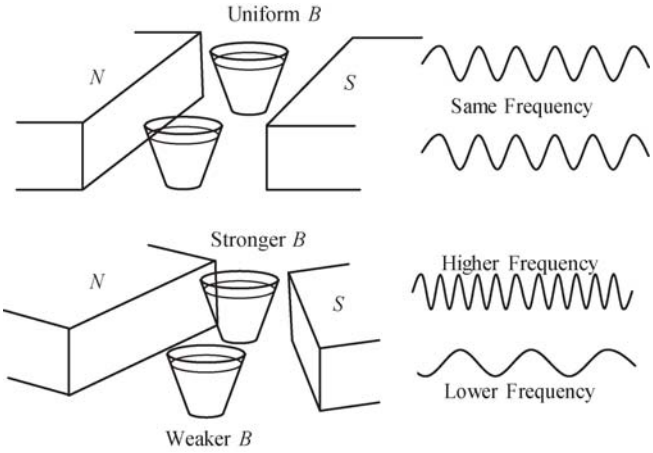


Fig. 7.11. Stronger field produces higher frequency.

The x-gradient coils are depicted in Figure 7.12. We assume that the x-direction is the direction from the patient's right ear to left ear. The x-gradient is turned on only when the RF signal is received, and this gradient is also called the read-out gradient. Do not turn it on during slice selection. When you turn two gradients on at the same time, they will combine and form a gradient in the third direction.

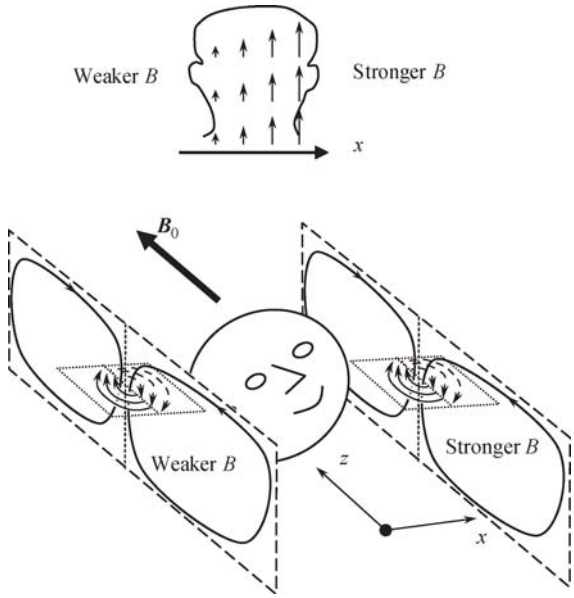


Fig. 7.12. The x-gradient coils generate a non-uniform magnetic field in the read-out direction.

7.3.3 To Obtain y-Information — Phase Encoding

After slice selection and before RF signal read-out, we turn on the y-gradient for a short time. Before a gradient is turned on, $\boldsymbol{M}$ at all locations precess at the same frequency (i.e., same speed). However, when the y-gradient is turned on, the field strength at a different y-position is different. As a result, $\boldsymbol{M}$ at a different y-position precesses at a different Larmor frequency (e.g., precession at a faster rate for a position with a larger y value, see Figure 7.13). After the y-gradient is on for a short while, it is turned off. At this

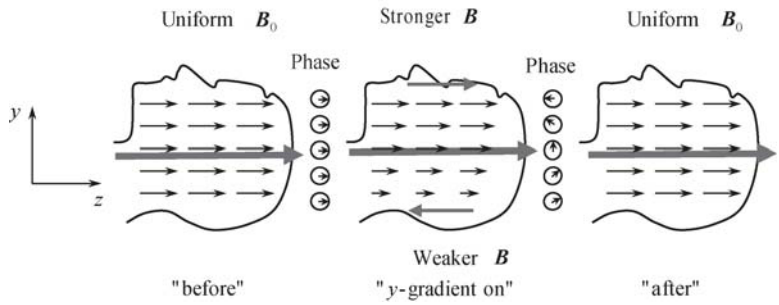


Fig. 7.13. The y-gradient causes the phase displacement as a function of y location.

moment, $\boldsymbol{M}$ at a different y position will arrive at a different phase (i.e., angle). This phase carries the information of the y position.

The y-gradient coils look exactly like the x-gradient coils, except for a 90° rotation. The effective gradient is confined to the gap between two pairs of the coils (see Figure 7.14).

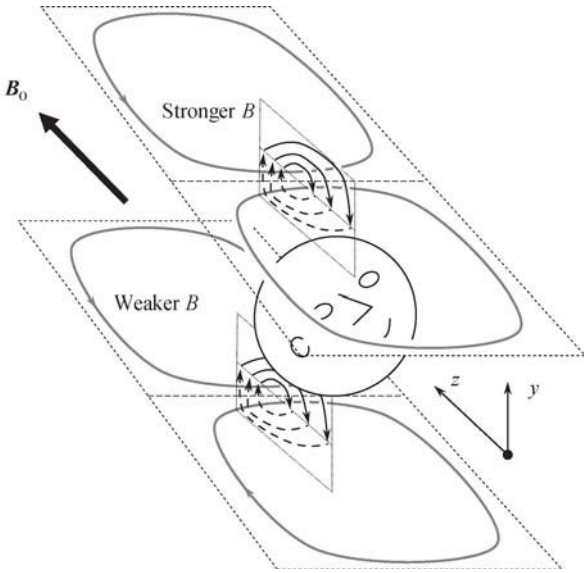


Fig. 7.14. The y-gradient coils make the magnetic field non-uniform in the y direction.

The time diagram shown in Figure 7.15 summaries this basic MRI data acquisition procedure. This procedure is repeated many times. At each time, a different value of the y-gradient is used. The next section will show that the acquired data is nothing but the 2D Fourier transform of the image $f(x, y)$, which is closely related to the proton density distribution inside the patient body. A 2D inverse Fourier transform is used to reconstruct the image. A typical MRI image is shown in Figure 7.16. There is a small negative pulse

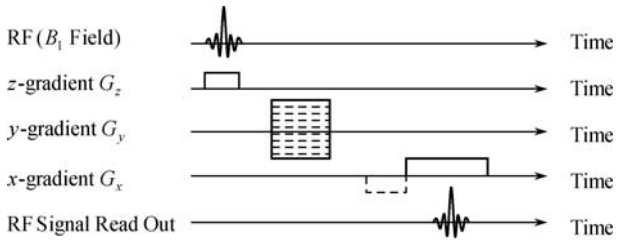


Fig. 7.15. The timing diagram for an MRI pulse sequence.

in front of the x-gradient readout pulse. The purpose of it is to create an "echo" to make the strongest signal at the center.

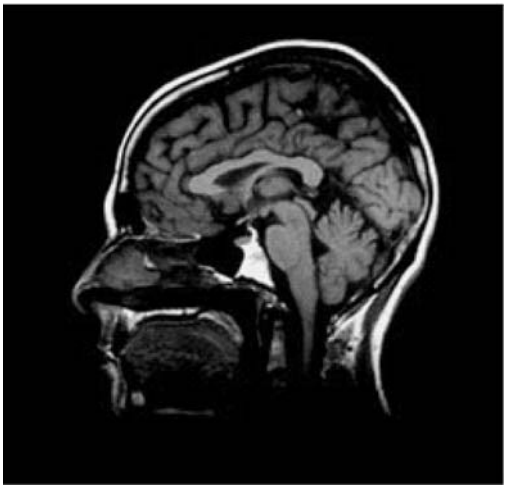


Fig. 7.16. An MRI image of the head.

*7.4 Mathematical Expressions

In this section, we assume that a slice selection has been done and $\boldsymbol{M}(x, y)$ is a function of x and y. The vector $\boldsymbol{M}(x, y)$ can be decomposed into the x-component $M_x(x, y)$, the y-component $M_y(x, y)$, and the z-component $M_z(x, y)$. We define a complex function $f(x,\, y)$ as

$$f(x, y) = M_x(x, y) + \mathrm{i}\, M_y(x, y). \tag{7.4.1}$$

The goal of MRI is to obtain this function $f(x,\, y)$ and display its magnitude $|f(x,\, y)|$ as the final output for the radiologists to read.

Let us first consider the effect of the read-out (x) gradient. When the x-gradient is turned on, the magnetic field strength is a function of x as

$$B(x) = B_0 + xG_x \tag{7.4.2}$$

and the associated Larmor frequency is calculated as

$$\omega(x) = \gamma(B_0 + xG_x). \tag{7.4.3}$$

At read-out, the function $f(x, y)$ will be encoded as

$$f(x, y)\cos(2\pi\omega(x)t) = f(x, y)\cos(2\pi\gamma(B_0 + xG_x)t), \tag{7.4.4}$$

where γB_0 is the carrier frequency and has no contribution for image reconstruction. In the MRI receiver there is a demodulator that can remove this carrier frequency. After the removal of the carrier frequency, the leftover baseband signal is given as

$$f(x, y) \cos(2\pi\gamma x G_x t). \tag{7.4.5}$$

Since the signal comes from the entire x-y plane, the received baseband signal is the summation of the signals from each location (x, y):

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \cos(2\pi\gamma x G_x t) \mathrm{d}x \mathrm{d}y. \tag{7.4.6}$$

Second, let us consider the effect of the phase-encoding (y) gradient. When the y-gradient is turned on for a period of time T, the magnetic field strength is a function of y as

$$B(y) = B_0 + yG_y, \tag{7.4.7}$$

and the associated Larmor frequency is calculated as

$$\omega(y) = \gamma(B_0 + yG_y). \tag{7.4.8}$$

After the period of T, the phase change is a function of y as

$$\phi(y) = T\omega(y) = \gamma\, B_0 + \gamma\, yG_y T. \tag{7.4.9}$$

Recall that the function $f(x, y)$ is complex with a magnitude and a phase and can be expressed as

$$f(x, y) = |f(x, y)|\, \mathrm{e}^{\mathrm{i}\,\varphi(x,y)}. \tag{7.4.10}$$

After a phase change of $\phi(y)$, $f(x, y)$ becomes

$$|f(x, y)|\, \mathrm{e}^{\mathrm{i}\,\varphi(x,y)} \mathrm{e}^{-\mathrm{i}\,\phi(y)} = f(x, y) \mathrm{e}^{-\mathrm{i}\,(\gamma B_0 T + \gamma y G_y T)}. \tag{7.4.11}$$

We can ignore the first term $\gamma B_0 T$ in the exponential because it introduces the same phase change to all y-positions and carries no information of the image.

The function $f(x, y)$ is now encoded by the phase changing factor as

$$f(x, y) \mathrm{e}^{-\mathrm{i}\,2\pi\gamma y G_y T}, \tag{7.4.12}$$

which is the signal that we try to read out. Therefore, the read-out signal is

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [f(x, y) \mathrm{e}^{-\mathrm{i}\,2\pi\gamma y G_y T}] \cos(2\pi\gamma x G_x t) \mathrm{d}x \mathrm{d}y. \tag{7.4.13}$$

The MRI machines use quadrature data acquisition, which has two outputs 90° out of phase. One output gives

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [f(x, y) \mathrm{e}^{-\mathrm{i}\,2\pi\gamma y G_y T}] \cos(2\pi\gamma x G_x t) \mathrm{d}x \mathrm{d}y, \tag{7.4.14}$$

and the other gives

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}[f(x,y)\mathrm{e}^{-\mathrm{i}\,2\pi\gamma yG_yT}]\sin(2\pi\gamma xG_xt)\mathrm{d}x\mathrm{d}y. \tag{7.4.15}$$

We can combine them into a complex signal, with one output as the real part and the other output as the imaginary part,

$$signal(t)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{i}\,2\pi\gamma yG_yT}\mathrm{e}^{-\mathrm{i}\,2\pi\gamma xG_xt}\mathrm{d}x\mathrm{d}y. \tag{7.4.16}$$

If we re-write it as

$$signal_{G_y}(t)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{i}\,2\pi[x(\gamma G_xt)+y(\gamma G_yT)]}\mathrm{d}x\mathrm{d}y, \tag{7.4.17}$$

we immediately recognize it as the 2D Fourier transform of $f(x,y)$:

$$F(k_x,k_y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{i}\,2\pi[x(\gamma G_xt)+y(\gamma G_yT)]}\mathrm{d}x\mathrm{d}y, \tag{7.4.18}$$

with $k_x=\gamma\ G_xt$ and $k_y=\gamma\ G_yT$. When we sample the time signal over time t, we get the samples of the x-direction frequencies, k_x. When we repeat the scan with a different value of G_y, we get the samples of the y-direction frequencies, k_y. For this reason, people often call the MRI signal space the k-space (see Figure 7.17), which is the Fourier space. During data acquisition, the k-space is filled out one line at a time according to $k_x=\gamma\ G_xt$ and $k_y=\gamma\ G_yT$.

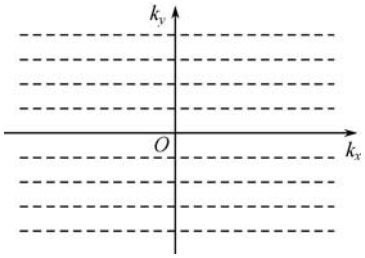


Fig. 7.17. The k-space.

Finally we will consider a polar k-space scanning strategy in which the x-gradient and the y-gradient are turned on and off simultaneously (see Figure 7.18). In this case, we do not have the phase encoding step; we only have the read-out gradient, which is determined by both the x and y gradients.

The signal read-out is given as

$$signal_{G_x,G_y}(t)=F(k_x,k_y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x,y)\mathrm{e}^{-\mathrm{i}\,2\pi[x(\gamma G_xt)+y(\gamma G_yt)]}\mathrm{d}x\mathrm{d}y, \tag{7.4.19}$$

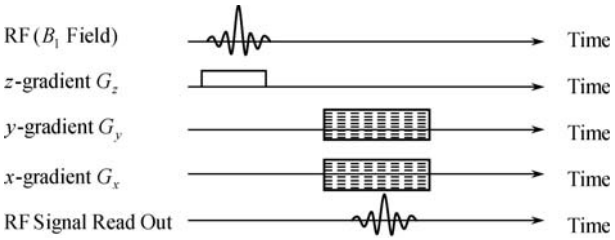


Fig. 7.18. The timing diagram for polar scanning.

with $k_x = \gamma\, G_x t$ and $k_y = \gamma\, G_y t$. The ratio $k_y/k_x = G_y/G_x$ tell us that each RF excitation cycle measures a line in the k-space with a slope of G_y/G_x (see Figure 7.19). For a different RF excitation, a new set of G_x and G_y is used, and a new line in the k-space is obtained. Figure 7.19 reminds us of the central slice theorem. We therefore can use the filtered backprojection algorithm (see Section 2.3) to reconstruct this MRI image.

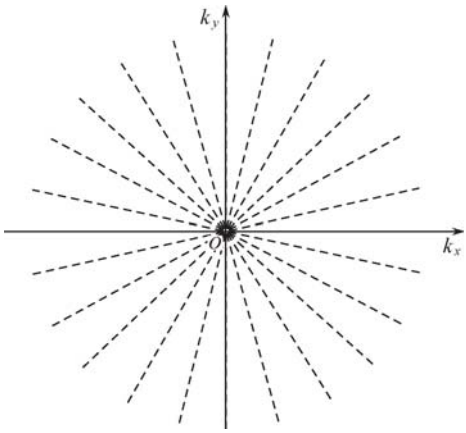


Fig. 7.19. The k-space sampling for the polar scan.

7.5 Worked Examples

Example 1 The vector $\boldsymbol{M}$ has three components: M_x, M_y, and M_z. Does the magnitude $\sqrt{M_x^2 + M_y^2 + M_z^2}$ always remain constant?

Answer

No. During relaxation, M_x and M_y relax to zero faster than M_z relaxes back to its equilibrium maximum value. This makes the magnitude $\sqrt{M_x^2 + M_y^2 + M_z^2}$ time varying.

Example 2 The received MRI signal is converted to a discrete signal via an analog-to-digital converter (ADC). Does the sampling rate of the ADC determine the image resolution?

Answer

No. The image resolution in MRI is determined by how far out the k-space is sampled. The distance from the farthest sample in the k-space to the origin (i.e., the DC point) gives the highest resolution in the image. The sampling time-interval in the ADC determines the image field of view. If the sampling rate of the ADC is not high enough, you will see image aliasing artifacts (for example, the nose appears at the back of the head).

Example 3* Design a pulse sequence that gives a spiral k-space trajectory.

Solution

In this case, the x- and y-gradients must be turned on simultaneously. The generic expression for k_x and k_y are $k_x(t) = \gamma \int_0^t G_x(\tau)\mathrm{d}t, k_y(t) = \gamma \int_0^t G_y(\tau)\mathrm{d}t$, respectively. On the other hand, a k-space spiral can be expressed as

$$\begin{cases} k_x = \alpha(t)\cos(\beta(t)), \\ k_y = \alpha(t)\sin(\beta(t)), \end{cases} \tag{7.5.1}$$

respectively, for some parameters α and β. Therefore, we can choose

$$\begin{cases} G_x = \dfrac{1}{\gamma}\dfrac{\mathrm{d}[\alpha(t)\cos(\beta(t))]}{\mathrm{d}t}, \\ G_y = \dfrac{1}{\gamma}\dfrac{\mathrm{d}[\alpha(t)\sin(\beta(t))]}{\mathrm{d}t}. \end{cases} \tag{7.5.2}$$

The corresponding time diagram and the k-space trajectory are shown in Figure 7.20. Its image reconstruction is normally performed by first regridding the k-space samples on the spiral trajectory into regularly spaced Cartesian coordinates then taking the 2D inverse Fourier transform to obtain the final image.

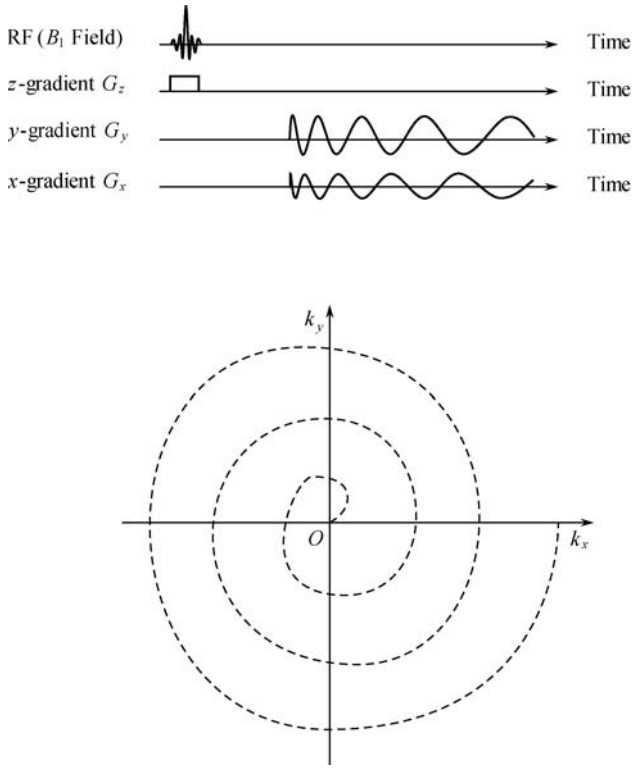


Fig. 7.20. The timing diagram and the k-space representation of a spiral scan.

7.6 Summary

- The working principle of MRI is quite different from that of transmission and emission tomography. The MRI signal is in the form of radio waves and is received by antennas (called coils).
- The "M" part: The patient must be positioned in a strong magnetic field, so that the magnetic moments created by the proton spins have a chance to line up.
- The "R" part: A resonant radio frequency signal is required to be emitted towards the patient, so that the net magnetic moments can be tipped over and do not align with the main magnetic field. When the net magnetic moments precess about the direction of the main magnetic field, radio frequency signals are sent out from the patient body.
- The "I" part: Gradient coils are turned on and off to encode the out-

coming radio frequency signals, so that the signals can carry to position information.

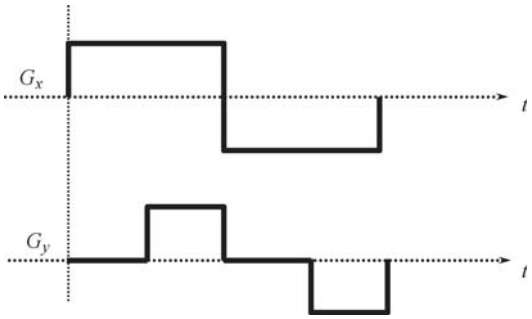
- The received MRI signal by the RF coils is in the Fourier domain (or spatial frequency space, or k-space). The image is reconstructed by performing a 2D inverse Fourier transform.
- The readers are expected to understand how the MRI signal is encoded to carry position information and why the MRI signal in the k-space is the 2D Fourier transform of the object.

Problems

Problem 7.1 During slice selection in the z direction, the slice thickness is not zero, but is a positive value Δz. Therefore, the RF pulse that generates the alternative magnetic field B_1 should have a proper bandwidth. How is this bandwidth determined by the slice thickness Δz?

Problem 7.2 If you plan to reconstruct an MRI image with an iterative algorithm, how do you handle the complex data? You may want to process the real-part of the data and the imaginary part of the data separately. How do you set up an objective function?

Problem 7.3 According to $k_x(t) = \gamma \int_0^t G_x(\tau)\mathrm{d}\tau$, $k_y(t) = \gamma \int_0^t G_y(\tau)\mathrm{d}\tau$ and the gradient waveforms given in the figure below, sketch the corresponding k-space trajectory.



References

1. Abragam A (1961) The principle of nuclear magnetic resonance. Oxford University Press, Oxford
2. Brown MA, Semelka RC (2003) MRI basic principles and applications. Wiley, Hoboken, New Jersey
3. Fukushima E, Raeder SBW (1981) Experimental pulse NMR, a nuts and bolts approach. Addison-Wesley, Reading, Massachusetts
4. Gerald LW, Carol P (1984) MRI a primer for medical imaging. Slack, Thorofare, New Jersey
5. Haacje EM, Brown RW, Thompson MR et al (1999) Magnetic resonance imaging: physical principles and sequence design. Wiley, New York
6. Liang ZP, Lauterbur PC (2000) Principles of magnetic resonance imaging: a signal processing perspective. IEEE Press, Piscataway, New Jersey
7. NessAiver M (1997) All you really need to know about MRI physics. Simply Physics, Baltimore, Maryland
8. Macovski A (1996) Noise in MRI. Magnetic Resonance Medicine 36: 494 – 497
9. Stark D, Bradley W (1992) Magnetic resonance imaging, Mosby. St. Louis. Missouri

Index

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S