# 2 The Basics of Traditional Reconstruction Methods

## 2.1 Line Integrals and Projections

A projection is set of line integrals. An example of a physical phenomenon that generates almost perfect line integrals is the attenuation of X-rays as they propagate through biological tissue. Fig. 1.1 in chapter1, shows a parallel projection. Fig. 2.1 also shows a parallel projection, but with more detailed geometry and coordinate systems. Clearly, *a parallel projection is a collection of parallel line integrals*.

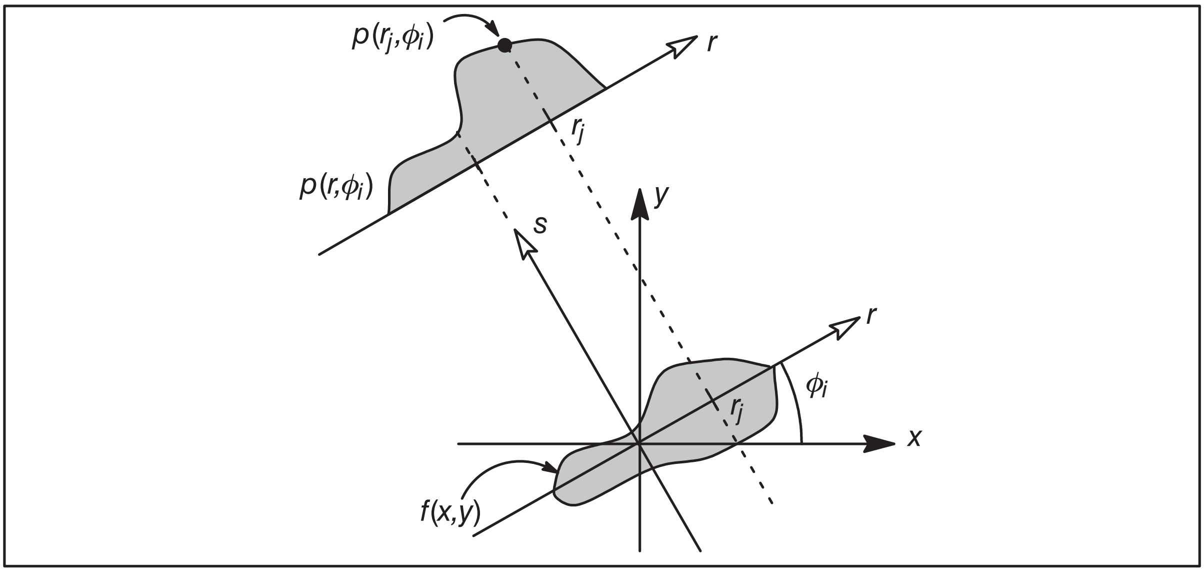


Fig. 2.1 An object $f(x,y)$ and one of its projections $p(r,\theta_i)$.

A parallel projection for a special $\theta=\theta_i$ is denoted $p(r,\theta_i)$. A set of projections, for all angles $\theta$, is denoted $p(r,\theta)$ and is also known as the Radon transform of $f(x,y)$ [Rad17].

The $(r,s)$ coordinate system is a rotated version of the $(x,y)$ coordinate system as expressed by

$$\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.1}$$

or similarly

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix}. \tag{2.2}$$

The set of projections $p(r,\theta)$ in (1.3) can then be formulated as

$$p(r,\theta) = \int_{-\infty}^{\infty} f\,(\underbrace{r\cos\theta - s\sin\theta}_{x},\ \underbrace{r\sin\theta + s\cos\theta}_{y})\,ds \tag{2.3}$$

or

$$p(r,\theta) = \int_{-\infty}^{\infty} f_r(r,s)\,ds \tag{2.4}$$

where $f_r(r,s)$ is a rotated version of $f(x,y)$. By using the delta function we get

$$p(r,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,\delta(x\cos\theta + y\sin\theta - r)\,dxdy. \tag{2.5}$$

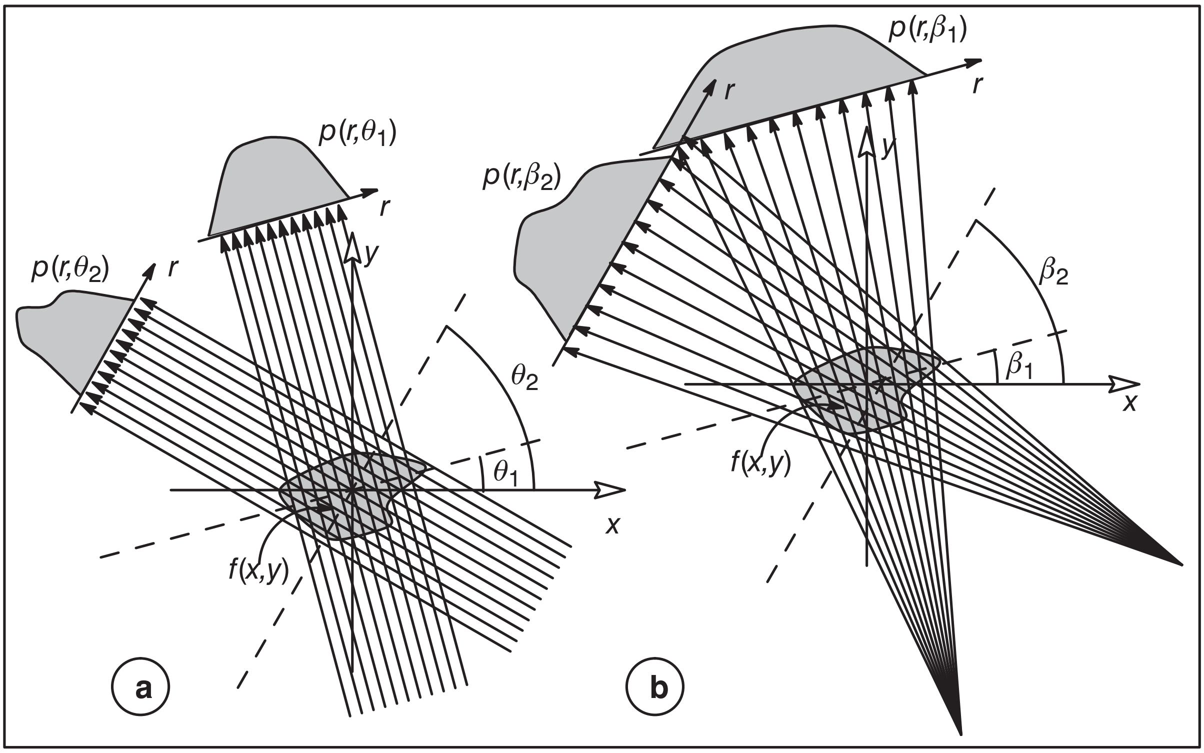


Fig. 2.2 Projection geometries. **(a)** The geometry for parallel projections. **(b)** The geometry for fanbeam projections.

Another type of projection geometry is known as fanbeam projection, see Fig. 2.2b. All modern commercial CT-scanners use fanbeam projection geometry. Part II of this thesis uses simulated special parallel linogram projection geometry for both the linogram reconstruction method and the filtered backprojection method. In part III, where we scrutinize the direct Fourier method, we are likewise employing parallel projection input data to the reconstruction experiments. Finally, in part IV, where everything is brought together, we use fanbeam projection data.

We often use the term **sinogram** in this thesis. A sinogram is the collection of parallel projections of the object-disk taken at equidistant angles $\theta$, see Fig. 2.3. The domain of the sinogram, seen as a 2D function $p(r,\theta)$, is $-R \le r \le R$, $-\pi/2 < \theta \le \pi/2$. The sinogram is a map of the projection data, i.e. of the Radon transform. The sinogram is cyclic in the $\theta$-coordinate. The term was first used by Edholm [Edh75] and is motivated by the fact that a point in the object gives rise to a sinusoid in the sinogram, see Fig. 2.4. The same point gives rise to a circle in the Radon space.

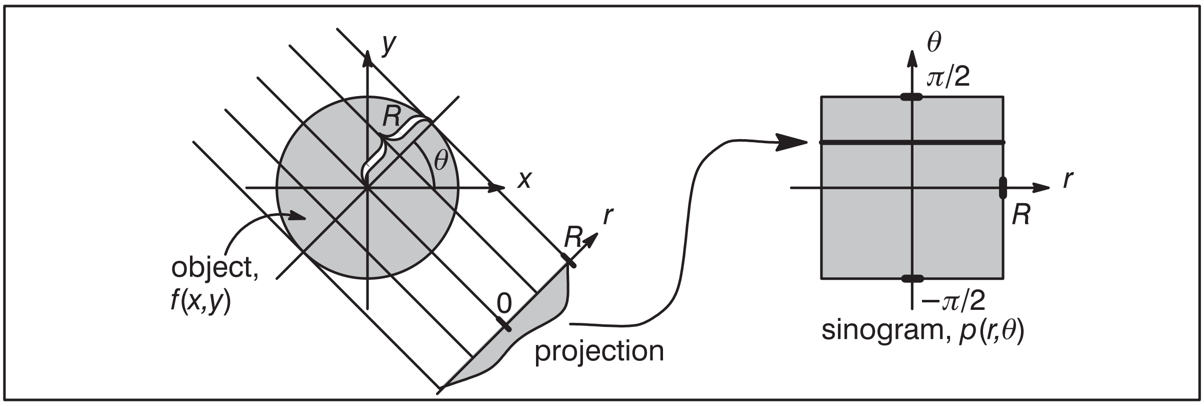


Fig. 2.3 Collection of projections of an object-disk in a sinogram.

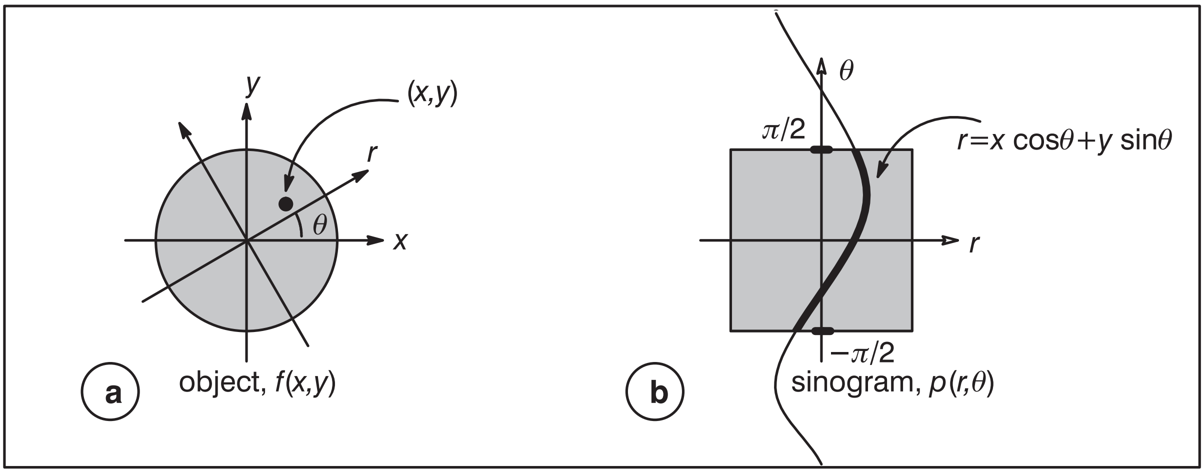


Fig. 2.4 A point in the object-disk gives rise to a sinusoid in the sinogram.

We summarize the above observations in proposition 1.

**Proposition 1** In theory, a sinogram $p(r,\theta)$ can be generated from $f(x,y)$ in the following two ways.

**A)** For each point $(r,\theta)$, compute or measure the line integral for the line $r = x\cos\theta + y\sin\theta$ in $f(x,y)$ and store as $p(r,\theta)$. This is Fig. 2.3.

**B)** For each point $(x,y)$, store $f(x,y)$ along the sinusoid $r = x\cos\theta + y\sin\theta$ in $p(r,\theta)$. This is Fig. 2.4.

Note that the line integrals in proposition 1A is what we get from a real CT-scanner, whereas the procedure in proposition 1B only can be done only if $f(x,y)$ is already known.

## 2.2 The Projection Slice Theorem

The projection slice theorem is stated as follows.

**Theorem 2** The Fourier transform $P(R,\theta_i)$ of a parallel projection $p(r,\theta_i)$ of an image $f(x,y)$ taken at angle $\theta_i$ is found in the two-dimensional transform $\boldsymbol{F}(X,Y)$ on a line with the angle $\theta_i$ against the $X$-axis.

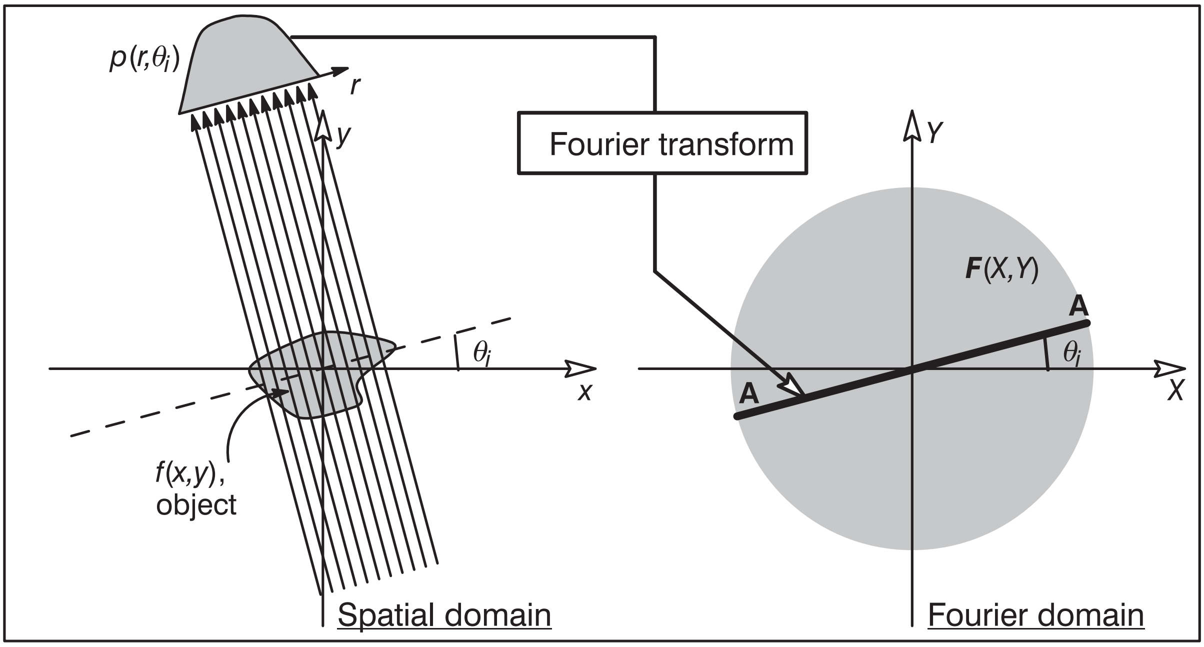


Fig. 2.5 The projection slice theorem illustrated.

The theorem states that the Fourier transform of $p(r,\theta_i)$ gives the values of $\boldsymbol{F}(X,Y)$ along line AA in Fig. 2.5. An intuitive understanding of the projection slice theorem may be given by the following, see Fig. 2.6. Consider the values along the $X$-axis in the Fourier domain, $\boldsymbol{F}(X,0)$. Here we find the zero frequencies in the y-direction from $f(x,y)$. But the zero frequencies in the $y$-direction can also be calculated by integrating in this direction along projection rays orthogonal to the $x$-axis and then taking the 1D Fourier transform in the $x$-direction. And this is exactly what the projection slice theorem states for a projection with the angle $\theta_i = 0$. For symmetry reasons this explanation can be applied to all lines through the origin in $\boldsymbol{F}(X,Y)$ and the projection slice theorem should hold for all $\theta$.

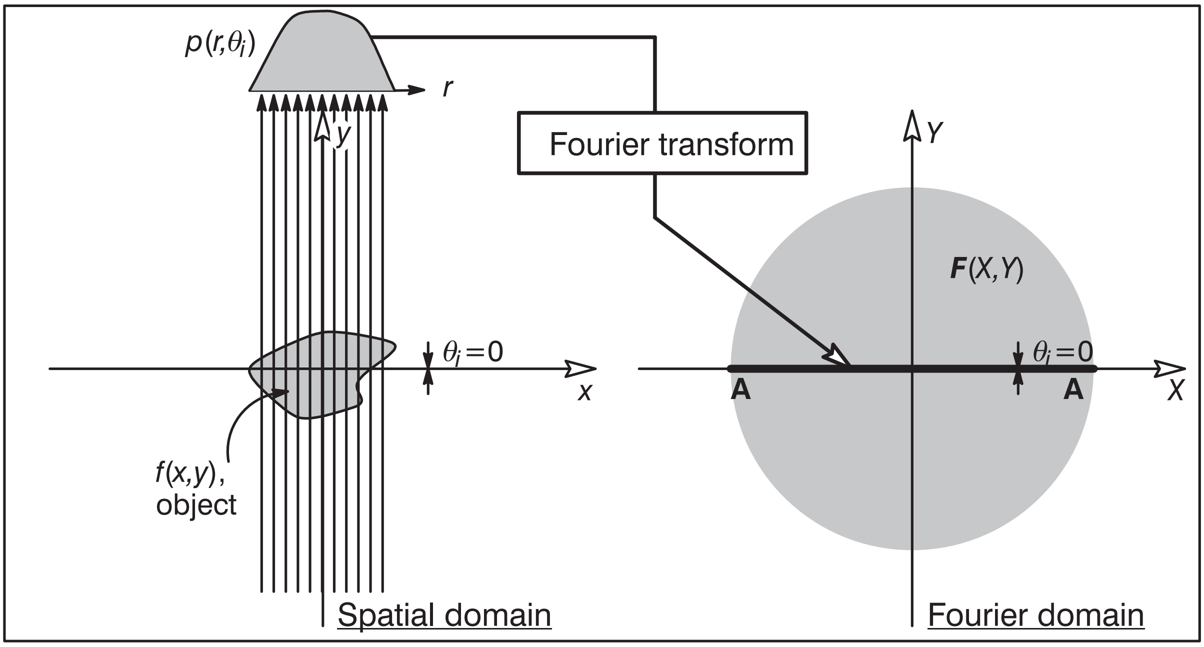


Fig. 2.6 A special case of the projection slice theorem.

Mathematically, the Fourier transform of $p(r,\theta)$ in $r$ is given by

$$P(R,\theta) = \int_{-\infty}^{\infty} p(r,\theta)\, e^{-j2\pi Rr} dr \tag{2.6}$$

or, when inserting (2.3),

$$P(R,\theta) = \int_{-\infty}^{\infty} \left[ \int_{-\infty}^{\infty} f(r\cos\theta - s\sin\theta,\ r\sin\theta + s\cos\theta)\, ds \right] e^{-j2\pi Rr} dr. \tag{2.7}$$

We change to the $(x,y)$ coordinate system using $r = x\cdot\cos\theta + y\cdot\sin\theta$, $x = r\cdot\cos\theta - s\cdot\sin\theta$, $y = r\cdot\sin\theta + s\cdot\cos\theta$, and by replacing the area differential $drds$ with $dxdy$. The result is

$$P(R,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi R(x\cos\theta + y\sin\theta)}\, dxdy \tag{2.8}$$

or

$$P(R,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-j2\pi\left[\overbrace{(R\cos\theta)}^{X}x + \overbrace{(R\sin\theta)}^{Y}y\right]} dxdy. \tag{2.9}$$

For any fixed value of $\theta$, the right hand side of this equation now represents values along a line through the origin of the two-dimensional Fourier transform $\boldsymbol{F}(X,Y)$ and forming the angle $\theta$ with the $X$-axis. The coordinate values along the line are $(X,Y)=R(\cos\theta, \sin\theta)$. Thus, we can rewrite (2.9) as

$$P(R,\theta) = \boldsymbol{F}(R\cos\theta,\ R\sin\theta) \tag{2.10}$$

which proves the projection slice theorem.

## 2.3 The Filtered Backprojection Method

### 2.3.1 Intuitive description

Today, the common method for CT-reconstruction is the Filtered Backprojection Method (FBM). The backprojection itself means that projection data are smeared back along the lines of their origin as shown in Fig. 2.7.

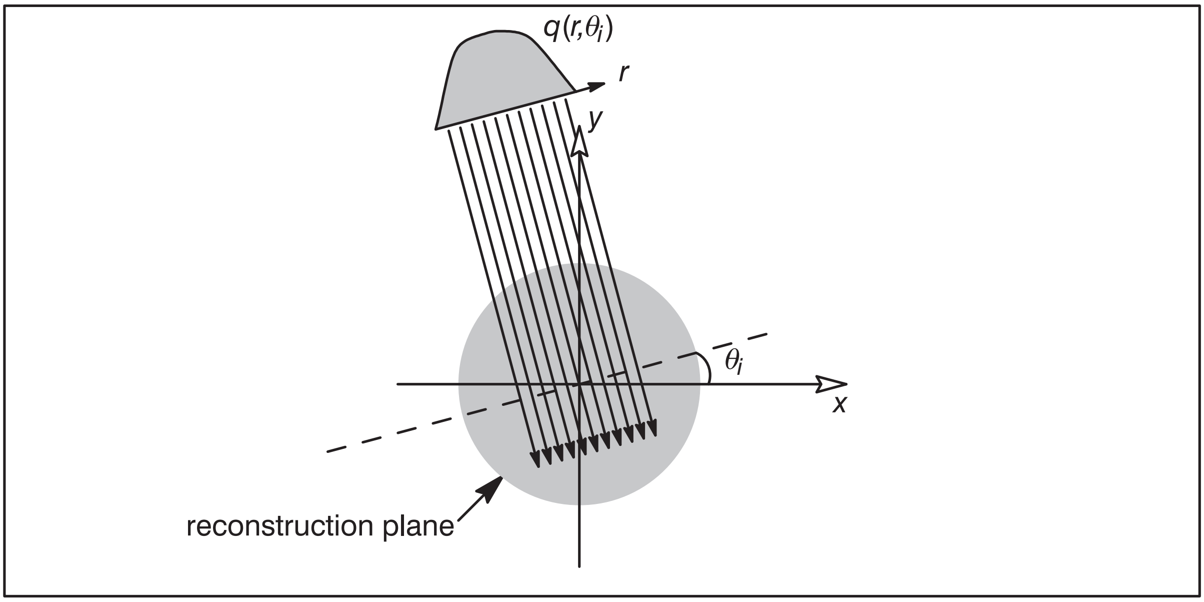


Fig. 2.7 A filtered projection $q(r,\theta_i)$ is smeared out (backprojected) over the reconstruction plane.

Projection followed by backprojection is an operation which causes some sort of low-pass filtering in the image, see Fig. 2.8. Projections $p(r,\theta)$ are taken of a point-shaped object as shown to the left. Then these projections are backprojected over the reconstructed image area as shown in b), Fig. 2.8. The whole procedure is repeated for different equidistant $\theta$:s, not only for the four backprojections shown in Fig. 2.8. We call the original function $f(x,y)$ and the result image $f_{blur}(x,y)$. Since the object is a single point, it is easy to see that $f_{blu}$r is not identical to $f$ but to $f$ convolved with the point spread function is $1/\varrho$, where $\varrho$ is the distance from the origin, i.e.

$$f_{blur}(x,y) = 1/\rho * f(x,y). \tag{2.11}$$

The Fourier transform of $1/\varrho$ is $1/P$ so that

$$\boldsymbol{F}_{blur}(X,Y) = 1/P \cdot \boldsymbol{F}(X,Y). \tag{2.12}$$

Thus, to compensate for the blurring filter $1/P$ inherent in the backprojection method, we can multiply $\boldsymbol{F}_{blur}(X,Y)$ with $P$, the inverse of $1/P$. One possibility to reconstruct $\boldsymbol{F}(X,Y)$ is then to perform

$$\boldsymbol{F}(X,Y) = P \cdot \boldsymbol{F}_{blur}(X,Y). \tag{2.13}$$

The functions $1/\varrho$, $1/P$ and $P$, are shown in Fig. 2.9.

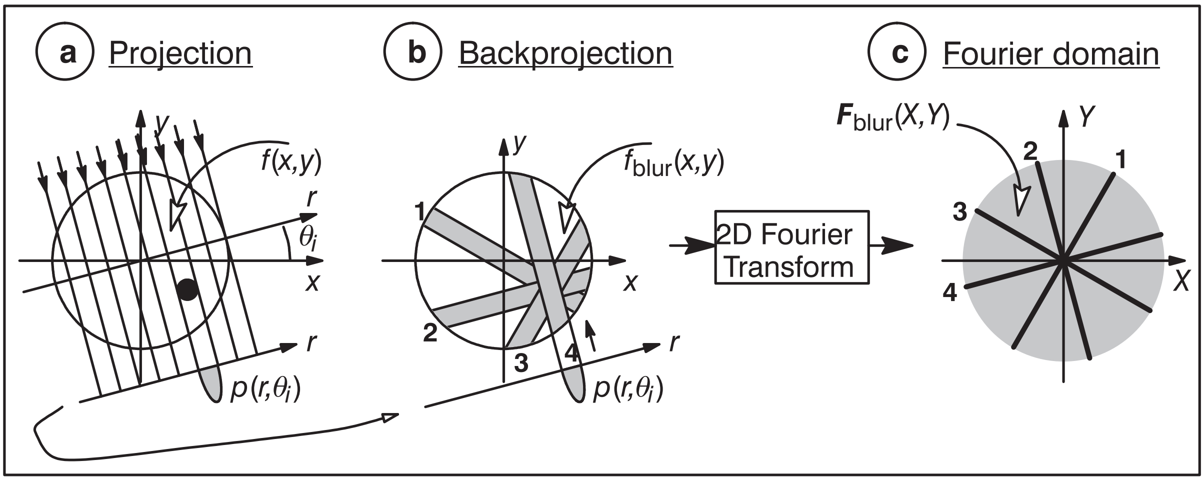


Fig. 2.8 A set of projections followed by a set of backprojections causes some sort of low-pass filtering in the image. This is easy to see when the object consists of a single point.

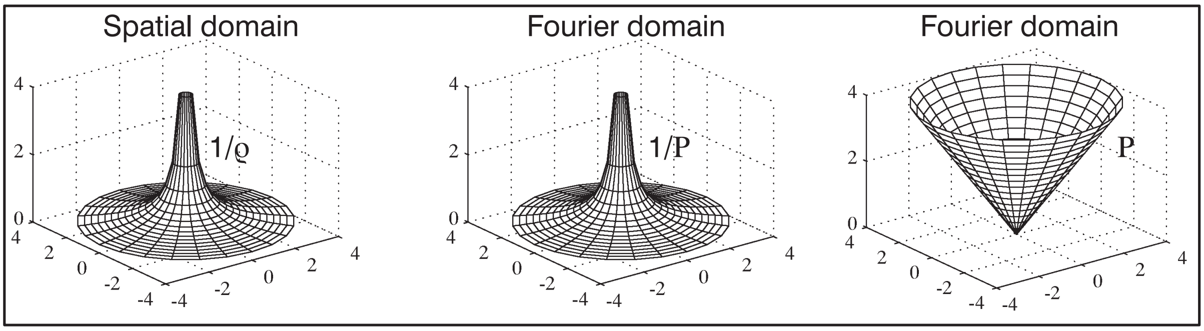


Fig. 2.9 Projection followed by backprojection causes low-pass filtering in the image. The equivalent filter function is $1/\varrho$ in the spatial domain or $1/P$ in the Fourier domain. To compensate for the low-pass filtering we can multiply the image with $P$ in the Fourier domain.

Instead of doing the compensating filtering in the 2D Fourier domain as indicated by (2.13), we can get the same effect by filtering each projection with the 1D ramp-filter

$$H(R) = |R| \tag{2.14}$$

shown to the right in Fig. 2.10. The reason is as follows. The backprojection in Fig. 2.8b is a straightforward summation. In the Fourier domain this is also a summation albeit now of slices through the origin shown in Fig. 2.8c. It is this 2D-function that according to (2.13) should be multiplied with $P$. However, if we multiply each contribution (i.e. the individual slices) with $H(R)=|R|$, we will get an identical result.

Alternatively, we can convolve each projection in the spatial domain with the inverse Fourier transform of $H(R)$, the 1D-function $h(r)$. If we assume that $H(R)$ is limited to the interval $-W$ to $W$ this convolution function is

$$h(r) \;=\; \frac{1}{2T^2} sinc\left(\frac{r}{T}\right) \;-\; \frac{1}{4T^2} sinc^2\left(\frac{r}{2T}\right) \tag{2.15}$$

where $T$ is the sampling interval ($T=1/(2W)$). The band-limited version of $H(R)$ and the corresponding function $h(r)$ are shown in Fig. 2.10.

An overview of the FBM is shown in Fig. 2.11. Note that the steps a2), a3), and a4) can be replaced by a convolution with $h(r)$.

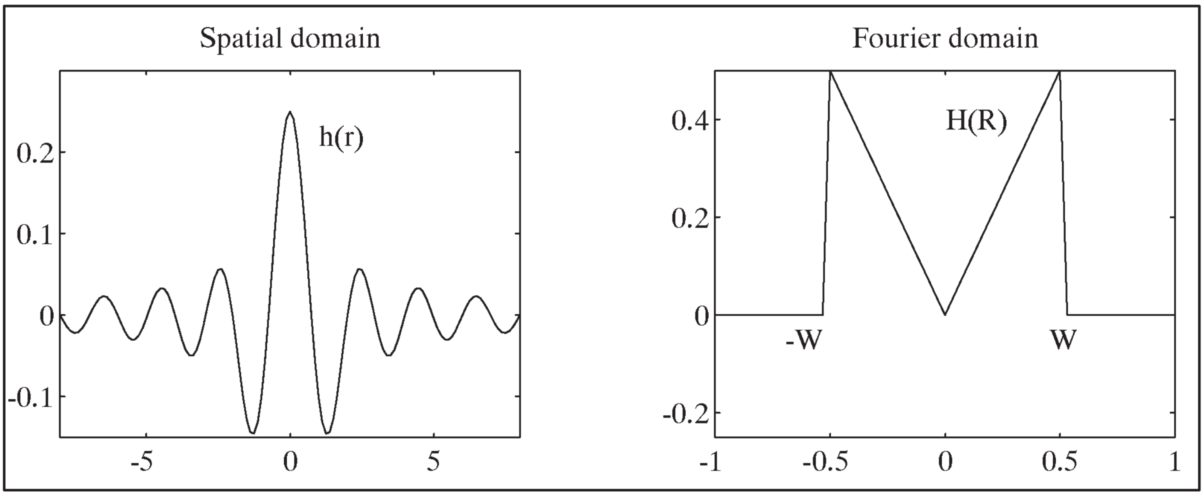


Fig. 2.10 The band-limited version of the ramp-filter $H(R)=|R|$ shown in both domains.

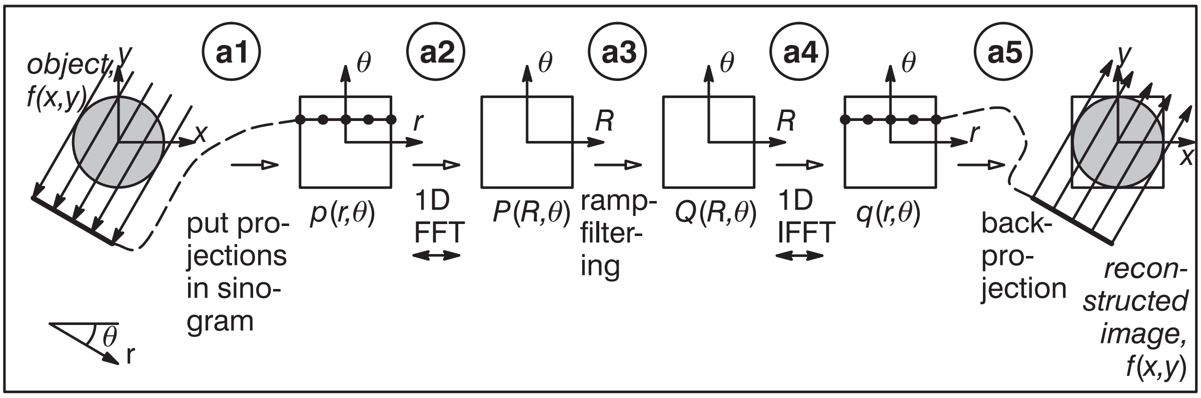


Fig. 2.11 An overview of the FBM.

**a1)** Take parallel projections around the object for a number of different equidistant angles $\theta$ and collect them in a so called sinogram $p(r,\theta)$ (the Radon transform).

**a2)** Take the 1D Fourier transform of each projection to obtain $P(R,\theta)$.

**a3)** Perform ramp-filtering according to: $Q(R,\theta) = |R| \cdot P(R,\theta)$.

**a4)** Take the 1D inverse Fourier transform in the $R$-direction. We now have filtered projections: $q(r,\theta)$.

**a5)** Perform backprojection with each filtered projection in $q(r,\theta)$.

## 2.3.2 Formal Proof

The 2-dimensional inverse Fourier transform of the object $f(x,y)$ is given by

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(X,Y)\, e^{j2\pi(Xx + Yy)}\, dXdY \tag{2.16}$$

We exchange the rectangular coordinate system in the Fourier domain $(X,Y)$, to a polar coordinate system $(R,\theta)$ by making the substitutions

$$X = R\cos\theta, \tag{2.17}$$

$$Y = R\sin\theta. \tag{2.18}$$

Using $dXdY{=}R\cdot dRd\theta$ we can now rewrite the formula as

$$f(x,y) = \int_{0}^{2\pi} \int_{0}^{\infty} F(R\cos\theta,\ R\sin\theta)\, e^{j2\pi R\,(x\cos\theta\ +\ y\sin\theta)}\, R\ dRd\theta \tag{2.19}$$

The integral can be split into two for $0 \le \theta < \pi$ and $0 \le (\theta + \pi) < \pi$, respectively. The result is

$$f(x,y) = \int_{0}^{\pi} \int_{0}^{\infty} F(R\cos\theta,\ R\sin\theta)\, e^{j2\pi R\,(x\cos\theta + y\sin\theta)}\, R\ dRd\theta\ +$$

$$+ \int_{0}^{\pi} \int_{0}^{\infty} F(R\cos(\theta + \pi), R\sin(\theta + \pi))\, e^{j2\pi R\,(x\cos(\theta+\pi) + y\sin(\theta+\pi))}\, R\ dRd\theta \tag{2.20}$$

Since

$$R\cos(\theta + \pi) = -R\cos\theta, \tag{2.21}$$

$$R\sin(\theta + \pi) = -R\sin\theta, \tag{2.22}$$

the formula (2.20) yields

$$f(x,y) = \int_{0}^{\pi} \int_{0}^{\infty} F(R\cos\theta,\ R\sin\theta)\, e^{j2\pi R\,(x\cos\theta + y\sin\theta)}\, R\ dRd\theta\ +$$

$$+ \int_{0}^{\pi} \int_{0}^{\infty} F(-R\cos\theta,\ -R\sin\theta)\, e^{j2\pi(-R)\,(x\cos\theta + y\sin\theta)}\, R\ dRd\theta\ =$$

$$= \int_{0}^{\pi} \left[ \int_{-\infty}^{\infty} F(R\cos\theta,\ R\sin\theta)\ |R|\ e^{j2\pi R\,(x\cos\theta + y\sin\theta)}\, dR \right] d\theta. \tag{2.23}$$

Using $r{=}x\cdot\cos\theta+y\cdot\sin\theta$, we get

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} \boldsymbol{F}(R\cos\theta,\ R\sin\theta)\ |R|\ e^{j2\pi Rr}\, dR\right] d\theta. \tag{2.24}$$

Finally, we utilize the projection slice theorem (2.10) to obtain

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} P(R,\theta)\ |R|\ e^{j2\pi Rr}\, dR\right] d\theta. \tag{2.25}$$

Inside out (2.25) is the same recipe as was illustrated in Fig. 2.11. The first computation is taking Fourier transforms of the projections $p(r,\theta)$ to obtain $P(R,\theta)$. The second one is

$$Q(R,\theta) = P(R,\theta)\cdot|R|, \tag{2.26}$$

so that (2.25) yields

$$f(x,y) = \int_0^{\pi}\left[\int_{-\infty}^{\infty} Q(R,\theta)\ e^{j2\pi Rr}\, dR\right] d\theta. \tag{2.27}$$

After 1D inverse Fourier transforms, the remaining step is the backprojection expressed by

$$f(x,y) = \int_0^{\pi} q(r,\theta)\, d\theta. \tag{2.28}$$

From (2.1) we know that $r = x\cdot\cos\theta + y\cdot\sin\theta$ so that (2.28) can be rewritten as

$$f(x,y) = \int_0^{\pi} q(\underbrace{x\cos\theta + y\sin\theta}_{r},\ \theta)\, d\theta. \tag{2.29}$$

This is the backprojection operation. See Fig. 2.12. For every point $(x,y)$ in the image there corresponds one certain $r{=}x\cdot\cos\theta+y\cdot\sin\theta$ in the projection $q(r,\theta)$, $\theta$ fix. According to (2.29) it is the value $q(r,\theta)$ that should be accumulated to $f(x,y)$. The points in the image that should receive the same contribution $q(r,\theta)$ are all found on the line $r{=}x\cdot\cos\theta+y\cdot\sin\theta$ which is the line AA in Fig. 2.12.

Thus, the value of the filtered projection, $q(r,\theta)$, will make the same contribution to all the points on the line AA. That is, in the reconstruction process, for every $\theta$ the filtered projections $q(r,\theta)$ are smeared out or backprojected over the image plane.

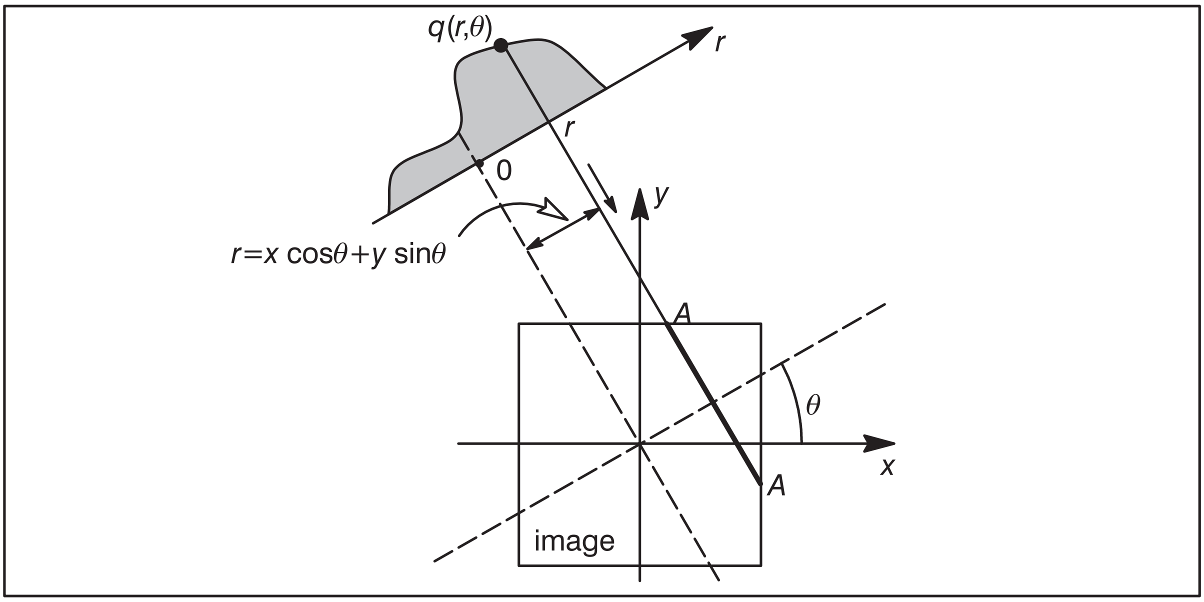


Fig. 2.12 Backprojection takes place when, for a given $\theta$, a filtered projection $q(r,\theta)$ is smeared out over the reconstruction plane along lines of constant $r$.

By insertion of (2.3) in (2.6) and (2.6) in (2.25), we get a fully unrolled formula (2.30) for all the steps in the FBM.

$$f(x,y) =$$

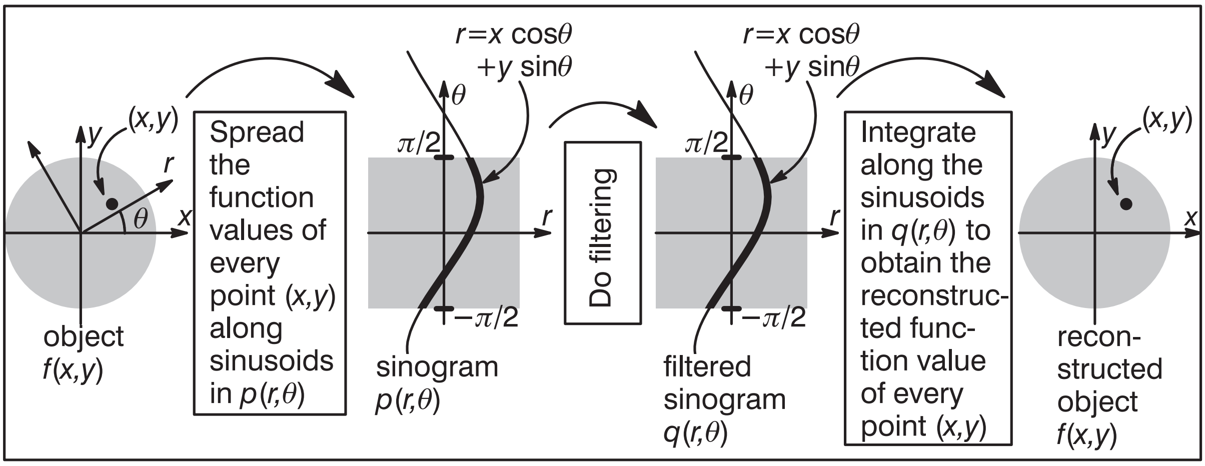
$$= \underbrace{\int_0^\pi \underbrace{\left[\int_{-\infty}^{\infty} \underbrace{\left[\int_{-\infty}^{\infty} \underbrace{\left[\int_{-\infty}^{\infty} f(\underbrace{r\cos\theta - s\sin\theta}_{x}, \underbrace{r\sin\theta + s\cos\theta}_{y})\, ds\right]}_{a1} e^{-j2\pi Rr}dr\right]}_{a2} \underbrace{|R|}_{a3} e^{j2\pi Rr}dR\right]}_{a4} d\theta}_{a5} \tag{2.30}$$

## 2.3.3 An Alternative View on Projection and Backprojection

An alternative method for backprojection is the following. We noticed in Fig. 2.4 that a point in $f(x,y)$ delivered its value to the sinogram in the form of a sinusoid. Since this is the effect of a *projection* it follows that *backprojection* means that points on this sinusoid are delivering their values back to the very same point (see Fig. 2.13).

For a better understanding of the linogram method to be presented later in this thesis, it might be useful to give the following alternative definitions of the two operations projection and backprojection. A projection as defined by the Radon transform is obtained by a correlation (or convolution) of a line with the image $f(x,y)$. A point in $p(r,\theta)$ has a point-spread function in the form of a line in $f(x,y)$. Conversely, a point in $f(x,y)$ has a point-spread function in the form of a sinusoid in $p(r,\theta)$. As shown in [Dan91] this observation is the basis for an alternative, totally general way to produce a convolution result which in [Dan91] is called the General Hough Transform, GHT. It is illustrated for the projection step to the left in Fig. 2.13, i.e. for each point in $f(x,y)$ the GHT generates a contribution in the form of a sinusoid. In conventional backprojection, on the other hand, we see from Fig. 2.12 that we are already using this GHT-technique. The alternative way is the convolution style shown to the right in Fig. 2.13. For each point $(x,y)$ in the result we integrate along the sinusoidal point-spread function in $q(r,\theta)$.

Now, if we do backprojection with non-filtered projections, we get a blurring effect which stems from the fact that the points of this sinusoid are parts of other sinusoids as well, and therefore spread their values to other parts of the image. The previously encountered ramp-filter is necessary for exact reconstruction.



y
(x,y)
r
θ
x
object
f(x,y)
Spread the function values of every point (x,y) along sinusoids in p(r,θ)
r=x cosθ
+y sinθ
θ
π/2
r
−π/2
sinogram
p(r,θ)
Do filtering
r=x cosθ
+y sinθ
θ
π/2
r
−π/2
filtered sinogram
q(r,θ)
Integrate along the sinusoids in q(r,θ) to obtain the reconstructed function value of every point (x,y)
y
(x,y)
x
reconstructed object
f(x,y)

Fig. 2.13 Alternative procedures for projection and backprojection when using sinograms.

We summarize the above observations as follows.

**Proposition 3** In the final backprojection step, the resulting image $f(x,y)$ can be obtained from the filtered projections $q(r,\theta)$ in the following two ways.

**A)** For each $(r,\theta)$ deliver the contribution $q(r,\theta)$ to $f(x,y)$ for all $(x,y)$ along the line $r = x \cdot \cos\theta + y \cdot \sin\theta$. This is Fig. 2.12.

**B)** For each $(x,y)$ compute the integral along the sinusoid $r = x \cdot \cos\theta + y \cdot \sin\theta$ in $q(r,\theta)$. This is the right part of Fig. 2.13.

## 2.3.4 Time Complexity

At this point a few words about time complexity seems appropriate. To obtain a faster reconstruction method than the FBM is the driving force behind this thesis and the final word on complexity will not be given until chapter 9. However, already here the following observations on the FBM complexity should be considered. Let $N$ be the number of pixels in the $x$- and $y$-direction of the image $f(x,y)$ to be reconstructed and let $M$ be the number of projections. As we will see later in this thesis, it is common practice to use $M \approx 2N$ and a number of line integrals (detector values) $\approx N$. Zero-padding to avoid circular convolution (see section 4.2) brings this number up to $2N$. The filtering step in the FBM includes FFT, multiplication with a ramp-filter in the Fourier domain and IFFT. FFT of a complex array of $N$ data points takes $1/2 \cdot N \log N$ so called butterfly operations (Butt). If the data points are real, this is reduced with a factor of 2. Hence the **filtering** requires the following number of butterfly operations

$$M \cdot \frac{1}{2} \cdot \frac{1}{2} 2N \log 2N + [\mathrm{Mul}] + M \cdot \frac{1}{2} \cdot \frac{1}{2} 2N \log 2N \approx$$
$$\approx N^2 \log 2N + [\mathrm{Mul}] + N^2 \log 2N \approx 2N^2 \log N \text{ Butt}$$

[Mul] is the negligible effort to multiply the Fourier transformed projections with the ramp-filter. The fundamental butterfly operation of the FFT is typically executed with 10 Floating Point OPerations (FLOP).

The most taxing procedure in FBM is the final **backprojection** step. In many CT-systems, special or semi-special hardware (backprojection boards) are used to gain speed in this process. Here, we are disregarding this fact. Instead we will try to translate the backprojection operation into FLOP. We think this is justified because *if* new methods (linogram and direct Fourier methods) should gain interest, be accepted, and becoming commercialized, then they would enjoy the same speed benefit of being cast in special hardware. To level the competitors we could not find a better measure than FLOP counts.

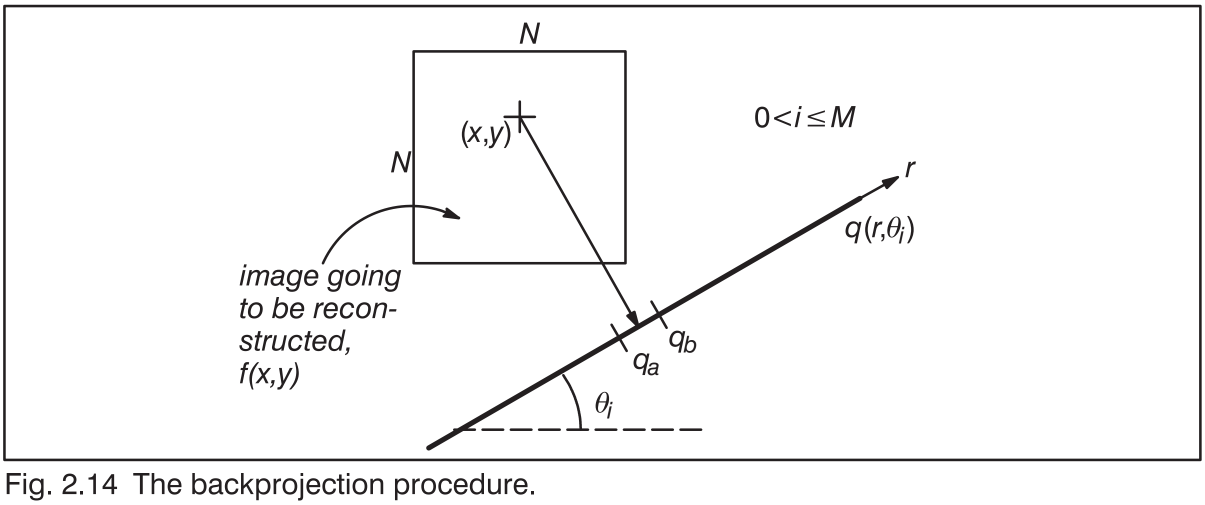


Fig. 2.14 The backprojection procedure.

The backprojection procedure is illustrated in Fig. 2.14. The pixel centered at $(x,y)$ in the output image is going to get its proper contribution from filtered projection data $q(r,\theta_i)$. For parallel projections, the coordinates $(x,y)$ are inserted in the line equation $r = x \cdot \cos\theta_i + y \cdot \sin\theta_i$ which is illustrated by the arrow from $(x,y)$ drawn perpendicular to the projection line. The projection value $q(r,\theta_i)$ is found at exactly this $r$ and obtained by interpolation from at least two neigbours $qa$, $qb$. As a final step, this value is accumulated to the pixel value $f(x,y)$. Clearly, since we have $N^2$ pixels and $M$ projections $(0 < i \leq M)$, the backprojection has an inner loop which is executed $MN^2 = \mathcal{O}(N^3)$ times. The loop requires computation of $r$, interpolation and accumulation. It seems unlikely that any general purpose computer could do this with less than five FLOP.

In fact, for fanbeam projections, the distance $l$ from the X-ray source to $(x,y)$ constitutes an individual weight factor $1/l^2$ for each pixel. This is the so called magnification factor (which is unity for parallel projections) and in the inner loop it has to be either computed or pre-computed and retrieved. In any case it will underscore that five FLOP for the inner loop in backprojection is a rather optimistic estimation. We summarize the time complexity estimations as follows.

$$\text{Filtering :} \qquad MN\log 2N \approx 2N^2 \log N \,\text{Butt} = 20N^2 \log N \text{ FLOP.} \tag{2.31}$$

$$\text{Backprojection :} \quad 5MN^2 \text{ FLOP} \approx 10N^3 \text{ FLOP.} \tag{2.32}$$

Since log$N$ is in the range of 10 we may claim that the approximate the total is

$$\text{FBM :} \qquad 200N^2 + 10N^3 \text{ FLOP.} \tag{2.33}$$

Accordingly, the FBM has the complexity $\mathcal{O}(N^3)$. For $N{=}512$, the second backprojection term dominates heavily over the first filtering step. The total computation is then 1.39 GFLOP of which 96% is backprojection.

## 2.4 The Direct Fourier Method

Already in the early 70's the Direct Fourier Method (DFM) was known in many circles [Tre89]. This method is a direct application of the projection slice theorem which indicates that by taking projections of $f(x,y)$ at angles $\theta_1, \theta_2, \ldots, \theta_\mathrm{k}$ and Fourier transforming each of these, we can determine the values of $\boldsymbol{F}(X,Y)$ on radial lines as shown in Fig. 2.15a. If an infinite number of projections were taken, $\boldsymbol{F}(X,Y)$ would be known at all points in the $XY$-plane, and it would seem as if the object function $f(x,y)$ could be recovered by using the inverse Fourier transform,

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{F}(X,Y)\, e^{j2\pi(Xx + Yy)}\, dX\, dY. \tag{2.34}$$

In practice only a finite number of projections of an object can be acquired, each projection consisting of a finite number of points. Therefore, $\boldsymbol{F}(X,Y)$ is only known at discrete points along the radial lines. To compute the inverse Fourier transform with a Fast Fourier Transform (FFT) we must determine the values on the square grid in Fig. 2.15b by some more or less sophisticated interpolation.

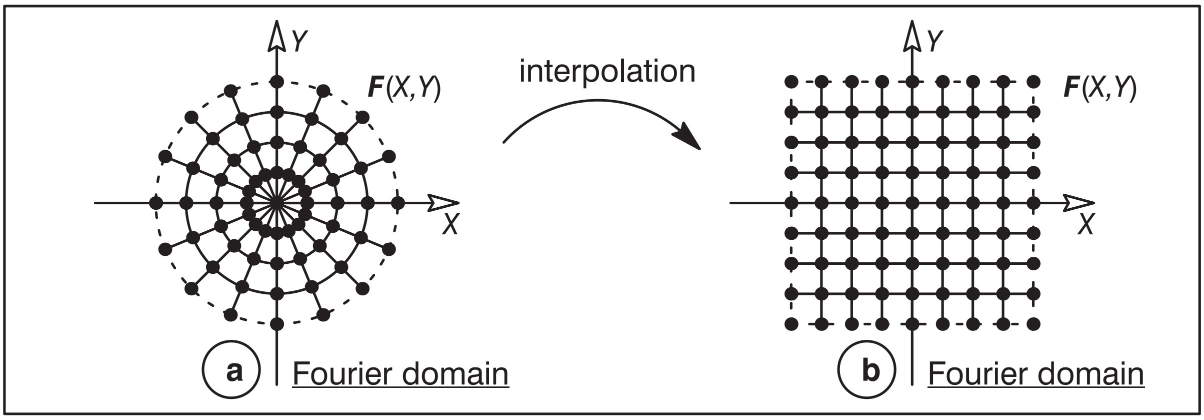


Fig. 2.15 **a)** The Fourier transform of the projections gives values along radial lines in the Fourier domain of the object. **(a–b)** Interpolation is used to get values on a square grid.

A summary of the DFM is shown in Fig. 2.16, where we store the projections intermediately in the Radon space. This is quite natural because of the symmetrical relation between Radon and Fourier spaces as expressed in the projection slice theorem.

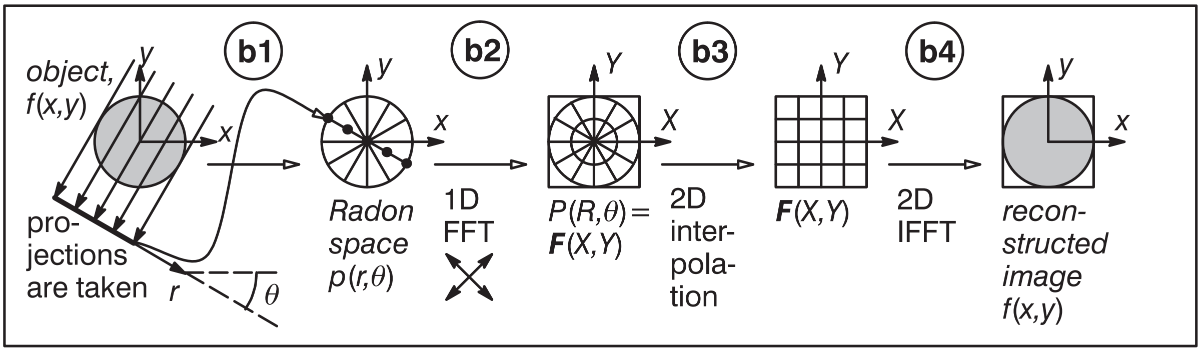


Fig. 2.16 An overview of the DFM.

- **b1)** Take parallel projections *p*(*r*,*θ*) around the object *f*(*x*,*y*) for a number of different equidistant angles *θ*. Put them intermediately in the Radon space.
- **b2)** Take 1D FFTs along the projection direction to arrive in ***F***(*X*,*Y*), the 2D Fourier transform of object.
- **b3)** Do 2D interpolation in the (*X*,*Y*)-space to obtain values on a square grid.
- **b4)** Perform 2D IFFT to receive the reconstructed image.

A few words should be said about time complexity of the DFM. Let $N$ and $M$ be the same as for the FBM in the previous section. ($M \approx 2N$ is the number of projections and $N$ is the number of pixels along the image side.) For real data, the first 1D FFT in the DFM has the computation complexity

$$const_1 \cdot M \cdot \frac{1}{2} \cdot \frac{1}{2} N \log N = \mathcal{O}(N^2 \log N).$$

The complexity of the interpolation is $const_2 \cdot N^2 \approx \mathcal{O}(const_2 \cdot N^2)$, which becomes $\mathcal{O}(N^2 \cdot \log N)$ if the Fourier domain interpolation of one point is of $\mathcal{O}(\log N)$. The last 2D IFFT step has the complexity

$$const_3 \cdot 2 \cdot N \cdot \frac{1}{2} \cdot \frac{1}{2} N \log N = \mathcal{O}(N^2 \log N).$$

Examples of things that influences the factors $const_1$ and $const_3$ are the increased data size due to zero-padding for avoiding circular convolution (see section 4.2). The total complexity for the DFM is then $\mathcal{O}(N^2 \log N)$, provided that the Fourier plane interpolation of one point is not larger than $\mathcal{O}(\log N)$.

It is this low complexity $\mathcal{O}(N^2 \log N)$ which makes the DFM interesting as an alternative to the FBM provided the issue on image quality and artifacts can be resolved. We will investigate this topic in more detail in chapter 8.