

Combining Fourier and iterative methods in computer tomography. Analysis of an iteration scheme. The 2D-case

Per-Erik Danielsson and Maria Magnusson Seger CVL, Dept of EE, Linköping University

15 october, 2004

Report No. LiTH-ISY-R-2634

Content	Page
Summary	3
1. The proposal. Related work	5
2. Recurrence equations	6
3. Convergence conditions	6
4. The matrix QP representing space-invariant filters	10
5. Symmetry of the QP -matrix	12
6. Input data. The ramp-filter	13
7. Ray- and pixel-driven projection and back-projection	18
8. Basis and window functions	20
9. Fourier modeling of projection and back-projection	23
10. Projection aliasing. General formulas	26
11. Back-projection aliasing. General formulas	29
12. Aliasing in linear interpolation. Joseph's method	30
13. Anti-aliasing using advanced window functions	33
14. Proposals for alias-free and stable iterative reconstruction	38
15. Experiments	
15.1 Convergence experiments with truncated rampfilter	40
15.2 Aliasing artefacts for various interpolation techniques	42
16. Discussions and conclusions	48
Acknowledgements	49
References	49

Summary

Most contemporary CT-systems employ non-exact methods. This treatise reports on how these methods could be transformed from non-exact to exact reconstruction methods by means of iterative post-processing. Compared to traditional algebraic reconstruction (ART) we expect much faster convergence (in theory quadratic), due to a much improved “first guess” and the fact that each iteration includes the same non-exact analytical reconstruction step as the first guess.

This proposal for combining analytical and iterative reconstruction methods was first presented in [6]. Here it is redrawn (in Fig.1.1) and slightly reformulated mathematically. The iteration loop corresponds to a matrix that we call QP , where Q is a Fourier-based reconstruction technique, such as filtered back-projection, and P is the forward projection operator. In exact reconstruction $Q = P^{-1}$, for non-exact reconstruction $Q \neq P^{-1}$. The iteration will converge as the error matrix $\Delta = I - \alpha(QP) \rightarrow 0$. Initially, there were some hopes that the convergence rate would indeed be quadratic, meaning that the largest pixel or voxel error, a diagonal element δ_{kk} in Δ should decrease quadratically as $\delta_{kk}^{i/2}$ with the number of iterations i . However, the experiments indicate a slower convergence rate.

Column k of the QP matrix comprises the point-spread function for pixel k . If the point-spread functions are identical under circular permutation for all k , then QP represents a *space-invariant filter*. The convergence could then be understood in filter theoretical terms, by which the QP -matrix implements a filter h_{QP} and the eigenvalues λ_k of the matrix can be identified as the frequency components of this space-invariant point-spread function. It can then be shown that the iterative loop converges if and only if for all frequency components $H_{QP}(\bar{u})$ of this filter $0 < \max_{\bar{u}} H_{QP}(\bar{u}) < \frac{2}{\alpha}$, where α is the loop gain.

The criteria for symmetry of QP (but not space-invariance) seems to be fulfilled for many non-exact reconstruction techniques, including the PI-methods for helical cone-beam tomography. Certain convergence criteria, or at least rules of thumb for convergence, can then be formulated. In general these formulas tell us how stability and convergence for larger degrees of non-exactness in the analytical reconstruction (larger pixel errors) is obtained at the price of lower gain α and slower convergence rate.

Several pit-falls for a successful implementation of the above ideas are contained in the interpolation procedures necessary for back-projection into a digital (sampled) image and even more so in computing projections thereof. To this end we first we clarify that the choice between so called pixel- and ray-driven mode is insignificant for the outcome of projection as well as back-projection. Of utmost importance, however, is to choose a suitable basis function, or, equivalently, to choose the window (interpolation) function, which in the 2D-case is the Radon transform of the actual basis function. The window function also goes under names like projection, profile, footprint, and the like. Traditionally, when employed in filtered back-projection from measured projection data, simple linear interpolation seems to work well, while for iterative CT, we found the simple linear interpolation to be detrimental, to say the least. Various new forms of aliasing (new, since they are hardly mentioned in the literature) come forward when we model and analyze such projection and back-projection in the Fourier domain.

The underlying cause of the problem is that the bandwidth of a Cartesian sampled image varies with direction. Typically, the width of a linear interpolation filter is tuned to the maximum bandwidth of input data. In the present case input data is the measured projections, the bandwidth of which is set to the Nyquist limits $\pm \frac{1}{2} \Delta_x^{-1}$, where Δ_x is the sampling distance in the main directions of the image to be reconstructed. The most outspoken form of aliasing, that we have coined *DC-aliasing*, tend to appear for projections in or near 45° - and 135° -directions. It consists of diminished, but still rather strong copies of the DC-component of the image, which are aliased to appear at the frequencies $\pm (\sqrt{2} - 1) \Delta_x^{-1}$. Joseph's interpolation method improves this aliasing situation, but it is not the final answer. Another aliasing effect (un-coined so far) seems to be lurking at the frequency $\pm \frac{1}{2} (\sqrt{2} - 1) \Delta_x^{-1}$. The imaginary part of this aliased component is phase-shifted 180° which

switches the negative feed-back in the main loop to become positive and thereby making the feed-back positive instead of negative. Only the DC-aliasing effects has been clearly verified experimentally.

There are at least two practical solutions to the aliasing problem, both of which employ more advanced window functions. The one we have been using is called the SinCot filter. In the Fourier domain this filter strongly attenuates all frequency components to values below $-80DB$ outside the double Nyquist limit. An alias-free situation is then obtained by using a detector density $\Delta_t^{-1} = \frac{3}{2}\Delta_x^{-1}$. In this treatise experiments on convergence for non-exact filtered back-projection are demonstrated only for the 2D-case. In many cases satisfying results are obtained after two iterations. For grossly non-exact reconstruction convergence is obtained by lowering the gain factor α in the main loop, which is in full accordance with the theory.

1. The proposal. Related work

The present document outlines an idea to combine analytical reconstruction methods with iterative reconstruction techniques. Here, analytical techniques means filtered back-projection or direct Fourier methods, while iterative techniques stand for algebraic reconstruction in general. We hope to show that such a combined approach is able to turn a simple-to-implement but mathematically non-exact reconstruction method into an exact one. One of many examples of such non-exact reconstruction is the PI-method [13], [18] suggested for helical cone-beam CT. With the addition of iterative methods the hope is to achieve an image quality that for all practical purposes is as good, maybe even better than what could be obtained by exact analytical methods. The potential to improve over purely analytical methods stems from the ability inherent in iterative methods to model the projection data input more precisely, possibly also to optimize image quality in the presence of noise.

A drawback in the exact reconstruction methods for helical cone-beam CT that have been presented hitherto [19],[20],[21] is their inability to utilize all projection data regardless of speed in the data-capture process. Many non-exact methods, such as the “Book-let method” adopted by Siemens [22] and the “Wedge” method adopted by Philips [23], but not the PI-methods, can tolerate an almost continuous range translation velocities and still not let any detector readings go wasted. Obviously, this helps to make an optimal choice between speed, dose and signal-to-noise ratio. At the best, the exact methods and the PI-methods are only able to make use of all measurements for some fixed translation speeds such as one third, one fifth, one seventh of the maximum. Still, without loss of generality, since the authors of this report are well acquainted with the PI-methods we are going to use these for the final tests of the proposal below.

The specific combination of analytical and iterative technologies that we have taken an interest in is illustrated in Fig.1.1. We may view this block diagram either as filtered back-projection augmented with an iterative image enhancing post-processing loop, or as a simultaneous algebraic reconstruction (SIRT) procedure using *filtered* difference projections in the back-projection loop. The tomography literature is abundant with iterative reconstruction methods. For CT-reconstruction in general, however, the scheme of Fig. 1.1 is hardly described at all not, even if it might be identical to the one called ILIN180 and evaluated in the experimentally study by Nuyts et al, [1]. Only linear interpolation was exploited in projection and back-projection in [1]. One reason for the relatively meager interest in this type of iteration is probably that existing analytical methods have been both fast and sufficiently accurate. Therefore, the incentive has not been great to pursue a technique that is obviously much more computer demanding. In our own work we have discovered how a poorly designed projection operator easily devastates the result in iterative CT-reconstructions. Some previous attempts in the direction of Fig.1.1 might have cooled off because of this.

As a general mathematical inversion method, however, schemes similar to Fig.1.1 can be found in handbooks, e.g. in Numerical recipes [2], which refers to this as Schultz’s method or Hotelling’s method. The same subject, although tuned more specifically towards tomographic reconstruction is dealt with by Pan et al [4].

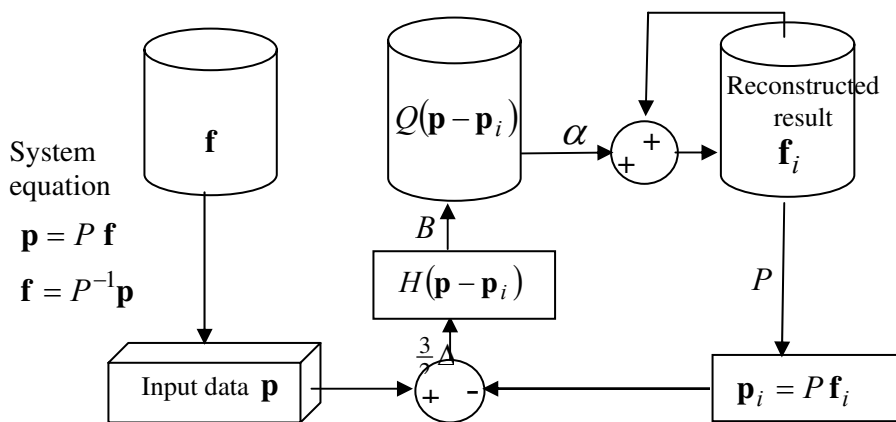


Fig.1.1 A proposal for combining analytical and iterative reconstruction techniques. The projection operator P and the reconstruction operator $Q \equiv BH \neq P^{-1}$ are known, while the exact inverse P^{-1} is unknown.

2. Recurrence equations

To explain Fig.1 we employ common matrix notation, where the image pixels (voxels) and the projection values (the line integrals through the images) constitute the vectors \mathbf{f} and \mathbf{p} , respectively. The matrices are the operators P for projection, B for back-projection, H for filtering, and Q for combined filtering and back-projection, respectively. The non-exactness is reflected in the fact that we may have $Q \equiv BH \neq P^{-1}$. The image may be two or three-dimensional. Assuming we initialize $\mathbf{f}_i \equiv \mathbf{f}_0 = 0$ we have

$$\mathbf{f}_1 \leftarrow \mathbf{f}_0 + \alpha Q(P\mathbf{f} - P\mathbf{f}_0) = \alpha QP\mathbf{f} \equiv \mathbf{f} - \Delta\mathbf{f} = \text{Correct result } \mathbf{f} - \text{artifact } \Delta\mathbf{f} \quad (2.1)$$

To get a quick feeling how the iterative loop works assume that the *gain* $\alpha = 1$. Then it is readily seen that the main purpose is to add negative errors in \mathbf{p}_i and subtract positive ones.

If the object function \mathbf{f} is a M -dimensional vector, the linear operator QP is defined by an $M \times M$ -matrix, which by our definition in (2.1) is the difference between the *unity matrix* I and the *error matrix* Δ . Hence,

$$\Delta\mathbf{f} = \mathbf{f} - \mathbf{f}_1 \quad \mathbf{f} = \mathbf{f}_1 + \Delta\mathbf{f}, \quad \Delta = I - \alpha QP \quad (2.2)$$

From Fig.1.1 we find that the iteration loop yields

$$\mathbf{f}_i \leftarrow \mathbf{f}_{i-1} + \alpha Q(P\mathbf{f} - P\mathbf{f}_{i-1}) = \mathbf{f}_{i-1} + \alpha QP(\mathbf{f} - \mathbf{f}_{i-1}) = \mathbf{f}_{i-1} + (I - \Delta)(\mathbf{f} - \mathbf{f}_{i-1}) = \mathbf{f} + \Delta(\mathbf{f}_{i-1} - \mathbf{f}) \quad (2.3)$$

$$\text{Therefore,} \quad \mathbf{f}_i - \mathbf{f} = \Delta(\mathbf{f}_{i-1} - \mathbf{f}) \quad (2.4)$$

$$\text{and} \quad \mathbf{f}_i = \mathbf{f} + \Delta(\mathbf{f}_{i-1} - \mathbf{f}) = \mathbf{f} + \Delta^2(\mathbf{f}_{i-2} - \mathbf{f}) = \dots = \mathbf{f} + \Delta^i(\mathbf{f}_0 - \mathbf{f})$$

With $\mathbf{f}_0 = 0$ we get

$$\mathbf{f}_i = \mathbf{f} - \Delta^i\mathbf{f} \quad (2.5)$$

The usefulness of the iterative scheme depends on two features of the matrix Δ^i , namely

- i) convergence conditions for Δ^i and
- ii) speed of this convergence

3. Convergence conditions. The symmetric QP-matrix

For any matrix, e.g. Δ^i it is well known that convergence, i.e. $\Delta^i \rightarrow 0$ for $i \rightarrow \infty$, occurs if and only if

$$|\lambda_{\max}(\Delta)| < 1, \quad (3.1)$$

where $|\lambda_{\max}(\Delta)|$ is the *largest magnitude of the eigenvalues* of Δ . If the matrix Δ is symmetric the eigenvalues are real. In the more general case that Δ is non-symmetric the eigenvalues are complex. Since $\Delta = I - \alpha QP$ the condition (3.1) can be expressed in terms of eigenvalues of the QP -matrix as follows.

$$|\lambda_{\max}(I - \alpha QP)| = \left| 1 - \alpha \lambda_k(QP) \right| = \left| \alpha \lambda_k(QP) - 1 \right| < 1 \quad (3.2)$$

Condition (3.2) does not seem to be very helpful, since we now require something from all eigenvalues of QP , not only the largest one in Δ . Furthermore, the condition now involves not only the eigenvalues $\lambda_k(QP)$ but also the gain α . However, without loss of generality we assume a common *positive gain*, i.e. we introduce the condition

$$0 < \alpha \quad (3.3)$$

by which (3.2) yields

$$\left| \lambda_k(QP) - \frac{1}{\alpha} \right| < \frac{1}{\alpha} \quad \forall k \quad (3.4)$$

If the eigenvalues are complex numbers, the condition (3.2) can be illustrated by the graph in Fig.3.1. All complex numbers inside the dashed circle satisfies the condition (3.4). For real eigenvalues (3.4) simplifies to $0 < \lambda_k(QP) < \frac{2}{\alpha}$. In our case, input (projections) and output (images) for reconstruction algorithms are real valued, not complex. Most reconstruction algorithms, including Fourier methods that are based on filtered back-projection, can be described and implemented by linear signal domain operators which make no use of complex coefficients. Hence, we allow ourselves to assume that QP contains no complex elements.

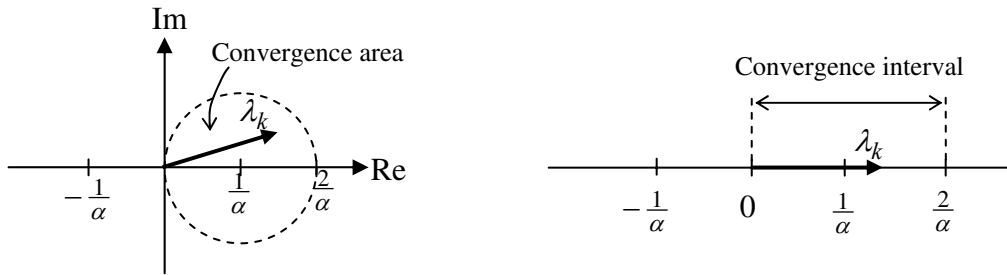


Fig. 3.1 Convergence for Δ^l expressed as conditions for complex and real eigenvalues λ_k of QP

In order to find convergence conditions that are more related to image artifacts, i.e. reconstruction errors caused by the non-exact filtered back-projection operator, let assume that the $n \times n$ matrix QP is indeed *symmetric*, i.e.

$$(QP)_{kl} = (QP)_{lk} \quad \text{for } \forall (k, l) \quad (3.5)$$

Also, let the input vector \mathbf{f} to the mapping QP be a single unit pulse at position k . The output vector, the result $QP\mathbf{f}$, is the discrete *point-spread function* Psf_k , which consists of the matrix elements of column k $(QP)_{1k}, (QP)_{2k}, \dots, (QP)_{kk}, \dots, (QP)_{nk}$. If all columns of QP are equal under permutation all point-spread functions Psf_k are identical, QP performs space-invariant filtering. However, the target application of the scheme in Fig.1.1 is non-exact reconstruction from cone beam projections, and such reconstruction is bound to have *space-variant point-spread functions*.

Fig.3.2 illustrates the relation between matrix elements and two space-variant point-spread functions laid out in a 2D image domain spanned by the spatial coordinate \bar{x} . The point-spread functions Psf_k and Psf_l are overlapping but not identical. From (2.2) we have $\alpha QP = (I - \Delta)$ from which follows that the symmetry (3.5) in QP implies that Δ is also symmetric. Furthermore, from (2.2) follows that the elements δ_{kk} and δ_{kl} in an arbitrary column k of Δ can be expressed by the corresponding elements in QP as

$$(QP)_{kk} = \alpha^{-1}(1 - \delta_{kk}) \quad (QP)_{lk, l \neq k} = -\alpha^{-1} \delta_{lk} \quad (3.6)$$

$$\delta_{kk} = 1 - \alpha(QP)_{kk} \quad \delta_{lk} = -\alpha(QP)_{lk, l \neq k} \quad (3.7)$$

Summing squared elements in column k of the matrix Δ yields

$$\sum_{1 \leq l \leq n} \delta_{lk}^2 = (1 - \alpha(QP)_{kk})^2 + \alpha^2 \sum_{l \neq k} (QP)_{lk}^2 = 1 - 2\alpha(QP)_{kk} + \alpha^2 \sum_{l=1}^n (QP)_{lk}^2 \quad (3.8)$$

Because of the symmetry of Δ , the left hand side of (3.8) is nothing but the k_{th} diagonal element $(\Delta^2)_{kk}$ of the matrix $\Delta^2 \equiv \Delta \cdot \Delta$.

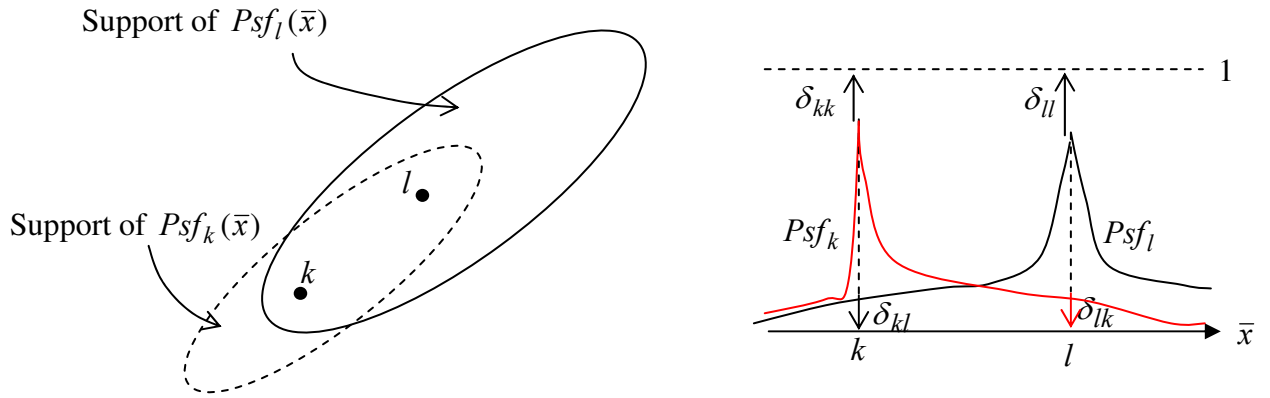


Fig.3.2 Two point-spread functions $Psf_k(\bar{x})$ and $Psf_l(\bar{x})$ assuming a symmetric QP -matrix. The symmetry implies $Psf(\bar{x} = k) \equiv (QP)_{kk}$, $Psf(\bar{x} = l) \equiv (QP)_{ll}$

Assume now that all columns are L_2 -normalized so that

$$\sum_{l=1}^n (QP)_{lk}^2 = 1 \quad \text{for } \forall k \quad (3.9)$$

by which we obtain the positive quantity

$$0 < (\Delta^2)_{kk} = \sum_{1 \leq l \leq n} \delta_{lk}^2 = 1 - 2\alpha(QP)_{kk} + \alpha^2 \quad (3.10)$$

Assume that all other normalized diagonal matrix element are larger or equal to $(QP)_{kk}$, which means that all other diagonal error matrix elements are smaller or equal to $(\Delta^2)_{kk}$. A necessary condition for convergence $\Delta^i \rightarrow 0$ for $i \rightarrow \infty$ is that

$$0 < (\Delta^2)_{kk} = 1 - 2\alpha(QP)_{kk} + \alpha^2 < 1 \quad \text{for all } k \quad (3.11)$$

Let us focus first on the *left hand inequality* in (3.11). We note that since all columns are L_2 -normalized

$$(QP)_{kk} \leq 1 \quad (3.12)$$

and also that the left hand inequality of (3.11) can be written as

$$(QP)_{kk} < \frac{1 + \alpha^2}{2\alpha} \quad (3.13)$$

The function to the right in (3.13) has a minimum = 1 for $\alpha = 1$ from which follows that the left hand of (3.11) is always true. The *right hand condition* of (3.11) can be written as

$$\alpha < 2(QP)_{kk} \quad (3.14)$$

Minimizing $(\Delta^2)_{kk} = 1 - 2\alpha(QP)_{kk} + \alpha^2$ as a function of α brings about

$$\alpha = (QP)_{kk} \quad (3.15)$$

and the minimum itself amounts to

$$\underset{Min}{(\Delta^2)_{kk}}(\alpha) = 1 - (QP)_{kk}^2 \quad (3.16)$$

It is tempting to take this quantity, or rather the inverse thereof, as a *general measure of quadratic convergence rate*. However, this is only true, or rather approximately true, if the diagonal elements $(QP)_{kk}$

dominate strongly over the non-diagonal ones. In the potential applications for Fig.1.1 we believe this is indeed the case.

Example. The scheme of Fig.1.1 is applied for 2D-reconstruction with no ramp-filtering or any other filter before back-projection. The remaining operations in the QP -loop will then deliver a strongly low-pass filtered result. We assume a circular image support having a diameter $1024\Delta_x$, where Δ_x is the image grid unit distance. The projection data are back-projected as they are, although divided (normalized) with the factor N_v , the number of views. The space-variant point-spread function can be derived as follows. See Fig. 3.3.

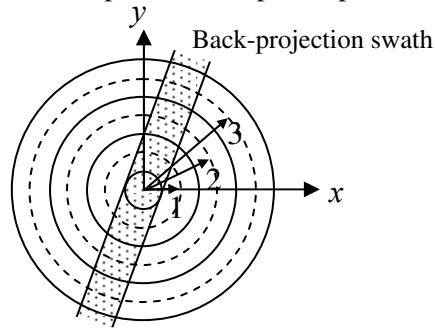


Fig. 3.3. A Circular symmetric point-spread function with density variation $(2\pi r)^{-1}$. The area is divided into a set of concentric rings at distance 1, 2, 3, The center pixel receives unit signal value provided the back-projected signal is normalized with the number of views N_v .

Let the input image be a single Dirac pulse located in a pixel in the exact image center. The projection-back-projection operation then brings about a correct pixel result identical to the input in the image center, i.e. $(QP)_{00} = 1$, $\delta_{00} = 0$. The space-invariant point-spread function is rotationally symmetric. At distance $r\Delta_x$ a ring-shaped area of width Δ_x contains $\approx 2\pi r$ pixels. The same unit signal value that was pumped back to the center pixel is evenly distributed to these pixels to yield the signal value $(2\pi r)^{-1}$. Hence, the sum of energies of elements $(QP)_{0k}$ at this distance r from the center yields $2\pi r(2\pi r)^{-2}$ and the total L_2 -sum amounts to

$$\beta^2 \equiv 1 + \sum_{r=1}^{512} 2\pi r \left(\frac{1}{2\pi r}\right)^2 \approx 1 + \frac{1}{2\pi} \int_1^{512} \frac{1}{r} dr = 1 + \frac{1}{2\pi} \ln 512 \approx 2 \quad (3.17)$$

Thus, a correct L_2 -normalization requires that we divide all projection data, not with N_v but with $\sqrt{2} N_v$. As a result we obtain for the diagonal elements $(QP)_{kk} = \frac{1}{\sqrt{2}}$, which also equals the square root of the sum of all off-diagonal elements $(QP)_{lk}$ in a column k . According to (3.15) optimal convergence rate is obtained with $\alpha = (QP)_{kk} = \frac{1}{\sqrt{2}}$ and from (3.16) we have

$$\left(\Delta^2\right)_{kk} = 1 - (QP)_{kk}^2 = \frac{1}{2} \quad (3.18)$$

which indicates a rather slow convergence factor of $\frac{1}{\sqrt{2}}$ per iteration.

4. The QP-matrix representing space-invariant filters

(In cooperation with Björn Johansson)

If the mapping QP is a linear system, which can be described with *space-invariant filters*, we can establish a perfect isomorphism between the world of linear algebra with its matrices and eigenvalues on one hand, and the world of signal processing with its convolutions, discrete Fourier transforms, and point-spread functions. More precisely, for any square matrix A , such as QP we claim the following.

i) When A is applied to an input image $f(x, y)$ consisting of a unit pulse at pixel k , the result is a point-spread which equals the matrix elements a_{kl} in row k . If this result repeats itself cyclically for all rows, the matrix A is equivalent to **cyclic** convolution with a space-invariant filter $h_A(x, y)$.

ii) Let $h \Leftrightarrow H$ stand for a Fourier pair. In signal processing notation, an image $f_{u,v}(x, y)$ consisting of a single frequency component at (u_1, v_1) , convolved with h_A yields

$$h_A * f_{u,v}(x, y) \Leftrightarrow H_A \delta(u - u_1, v - v_1) = H_A(u_1, v_1). \quad (4.1)$$

In matrix notation the same operation yields

$$A \mathbf{f}_{u,v} \Leftrightarrow H_A \delta(u - u_1, v - v_1) = H_A(u_1, v_1) \equiv \lambda_{u_1, v_1}(A) \delta(u - u_1, v - v_1), \quad (4.2)$$

where the single frequency image vector $\mathbf{f}_{u,v}$ is called an eigenfunction, its Fourier transform $\delta(u - u_1, v - v_1)$ is an eigenvector and the response $\lambda_{u_1, v_1}(A)$ is an eigenvalue of the matrix A . This interpretation is correct since we have found an eigenvector $\delta(u - u_1, v - v_1)$, which produces the same scalar result, whether we let the eigen vector be operated on by the matrix A or the eigenvalue λ_{u_1, v_1} .

iii) To get all eigenvalues efficiently, instead of applying one frequency component at a time, we can apply all of them at the same time. The suitable input image that contains all frequencies equally weighted is the unit pulse $\delta(0, 0)$ located at $x = 0, y = 0$. However, convolved with the filter h_A creates nothing but a copy of h_A . Hence, we have the following theorem.

Theorem on eigenvalues for space invariant QP-matrix

The eigenvalues of a circularly symmetric matrix $A = QP$ representing a space-invariant point-spread function are found by computing the discrete Fourier transform of the point-spread function of the corresponding cyclic filter h_A . The frequency components H_A are the eigenvalues of A .

Among the implications of the theorem we note the following. The point-spread function h_{QP} is real. The eigenvalues, i.e. the frequency components H_{QP} , are also real since QP is symmetric. Therefore, each eigenvector is actually the sum of the Dirac-pulse pair $\frac{1}{2} \delta(-u, -v) + \frac{1}{2} \delta(u, v)$. Hence, in this case we have

$$H_{QP}(u, v) \equiv H_{QP}(-u, -v) = \frac{1}{2} \lambda_{u,v}$$

Conditions (3.6) and (3.7) also implies evenness in the signal domain so that $h_{QP}(x, y) = h_{QP}(-x, -y)$ and that that all eigenfunctions are cosine functions. Any reader familiar with filter design should not be surprised by what we have found so far. We are studying a recursive filter, which essentially consists of a single negative feedback loop. The filters embedded in this loop are frequency dependent like any other filter. Negative values for certain frequencies in the Fourier transform of the non-recursive QP -filter means that these components are subjected to a 180° phase-shift each time they are propagated around the loop. Such frequency components may arrive with the input image, hidden among other benevolent components and impossible to detect in the image domain. Nevertheless, for these frequency components, the negative feedback loop will the turn into an amplifier with positive feedback. Inevitably, after some time, or rather after some iterations, the amplitude of these parts of the image will grow out of control and drench all other data.

For readers which are more familiar with signal processing and filter design it is also possible to avoid the matrix formulations altogether. Let \bar{u} be a vector in a multidimensional Fourier space. In (2.1) we described Fig1.1 by the formula

$$\mathbf{f}_i = \mathbf{f}_{i-1} + \alpha QP(\mathbf{f} - \mathbf{f}_{i-1}) \quad (2.1)$$

which translated to the Fourier domain yields

$$F_i(\bar{u}) = F_{i-1}(\bar{u}) + \alpha H_{QP}(\bar{u})(F(\bar{u}) - F_{i-1}(\bar{u})) \quad \text{for } \forall \bar{u} \quad (4.3)$$

$$F_i(\bar{u}) - F(\bar{u}) = (1 - \alpha H_{QP}(\bar{u}))(F_{i-1}(\bar{u}) - F(\bar{u})) \equiv \Delta^i(\bar{u})(F_{i-1}(\bar{u}) - F(\bar{u})) \quad \text{for } \forall \bar{u} \quad (4.4)$$

Clearly, as in (2.5) this reduces to

$$F_i(\bar{u}) = F(\bar{u}) - \Delta^i(\bar{u})F(\bar{u}) \quad \text{for } \forall \bar{u} \quad (4.5)$$

Notice that we make sure to follow each and every frequency component in F independently. Eq. (4.3) - (4.5) holds individually for each frequency component. For convergence we require likewise that

$$\Delta^i(\bar{u}) = (1 - \alpha H_{QP}(\bar{u}))^i \rightarrow 0 \quad \text{for } \forall \bar{u} \quad (4.6)$$

which requires that

$$|1 - \alpha H_{QP}(\bar{u})| < 1 \quad \Leftrightarrow \quad 0 < \alpha H_{QP}(\bar{u}) < 2 \quad \text{for } \forall \bar{u} \quad (4.7)$$

$$\text{If} \quad 0 < H_{QP}(\bar{u}) \quad \text{for } \forall \bar{u} \quad (4.8)$$

$$\text{then} \quad 0 < \max_{\bar{u}} H_{QP}(\bar{u}) < \frac{2}{\alpha} \quad (4.9)$$

Thus, (4.9) yields the following very precise convergence condition.

The Fourier transform H_{QP} of space-invariant point-spread function, h_{QP} , which appears in each row of the matrix QP , must be strictly positive and not exceed $\frac{2}{\alpha}$

Whether we employ the eigenvalue or the frequency domain criterion in (3.6) and (4.9), respectively, we must make certain that the filter h_{QP} that constitutes the QP -matrix is free from negative values in H_{QP} . One way to it is to make sure that all filters are even. Hence, it seems safer to use linear interpolation that is a sinc² function in the Fourier domain, than to use nearest neighbor interpolation, which is a sinc-function with negative lobes. Using an even filter function h_H and identical functions h_P and h_B means $P = B^T$, seems also safe, since a negative frequency component in $h_P(\bar{u})$ makes a positive component $h_{PQ}(\bar{u}) = h_P^2(\bar{u})$ in the concatenated total filter. However, as we will see in the sequel, aliasing phenomena may create unexpected problem in this respect.

Any reader familiar with filter design should not be surprised by what we have found so far. We are studying a recursive filter, which essentially consists of a single negative feedback loop. The forward filter h_{QP} (the point-spread function) in this loop is frequency dependent like any other filter. And placed in a recursive loop we must watch out for problems. Designing these filters so that their Fourier transforms are real and *positive*, makes life easy, while negative amplitude in the Fourier transform of the QP -filter will perform a 180° phase-shift for each iteration. Such frequency components may be lurking in the input image, hidden among the majority of benevolent components, impossible to detect in the image domain. For these frequency components, the negative feedback loop works as an amplifier with positive feedback. Inevitably, after some time, or rather after some iterations, the amplitude of these image components will grow out of control and drench all other data as illustrated in Fig.4.1.

Aliasing is an even more devious source of instability. As we will see in later sections of this documents linear interpolation using Joseph's technique [9] is quite effective in suppressing the most devastating form of aliasing, namely aliasing of the DC-component.

5. The symmetry of the QP-matrix

The QP -matrix embeds (at least) three operations; the forward projection P , the filtering H , and the back-projection B as illustrated in Fig.5.1. The projection P maps an input data vector \mathbf{f}_{in} (along the x -axis in Fig 5.1) in the image domain onto a projection vector \mathbf{p} (along the t -axis in Fig. 5.1) in the projection domain, in 2D also called Radon domain (but only in 2D), detector space, or projection space. The filtering H maps the projection vector \mathbf{p} back to the projection space forming the vector \mathbf{q} . Finally, the back-projection B takes this latter vector \mathbf{q} back to the image domain delivering the vector \mathbf{f}_{out} .

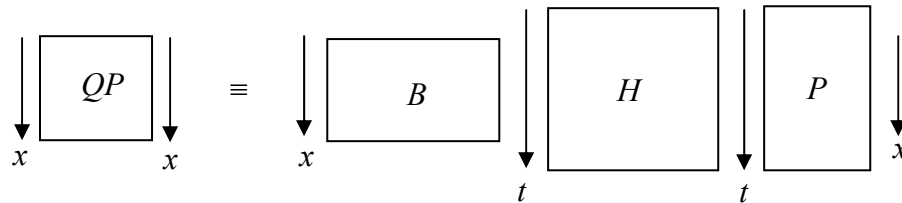


Fig. 5.1 The QP -matrix is the product of three matrices $B, H,$ and P . The input data axis x to the far right comprises all image data pixels, or its Fourier components, linearly arranged along this axis. Likewise, the axis t comprises all projection data.

We are interested to track down how a point-spread function comes about by following how a unit pulse in \mathbf{f}_{in} at one image point, corresponding to, say, row index k in the QP -matrix, is able to spread to give a non-zero result in \mathbf{f}_{out} at another image point, corresponding to, say, index l . See Fig. 5.2. Consider first a unit pulse at image point k , which by use of an interpolation function h_P delivers the signal value $h_P(a)$ to a projection ray. Via the ramp-filter function h_h this detector value will be transported with the factor $h_h(b)$ to a back-projection ray near image point l . Finally, via an interpolation with h_B the image point l receives $h_P(a) h_h(b) h_B(c)$ being one of many contributions to the matrix element $(QP)_{kl}$.

As shown in Fig.5.2, if a unit pulse injected at image point l , for each contribution path that went from k to l there is now one going from l to k , which amounts to $h_P(c) h_h(b) h_B(a)$. In principle we could nail down every such pair of contributions by following every non-zero path from k to l and from l to k , not only for the arbitrary projection we selected in Fig. 5.2 but for every other projection as well. In a typical case $(QP)_{kl}$ is accumulated from hundreds or thousands of such paths over which signal energy is transported. For symmetry we require that $(QP)_{lk} = (QP)_{kl}$ and if we find that the two mirror contributions in an arbitrary chosen pair are equal, the symmetry of the QP -matrix is secured. In the present case of Fig. 5.2 the symmetry would be secured by using the same even interpolation functions for projection and back-projection, i.e. $h_P \equiv h_B$.

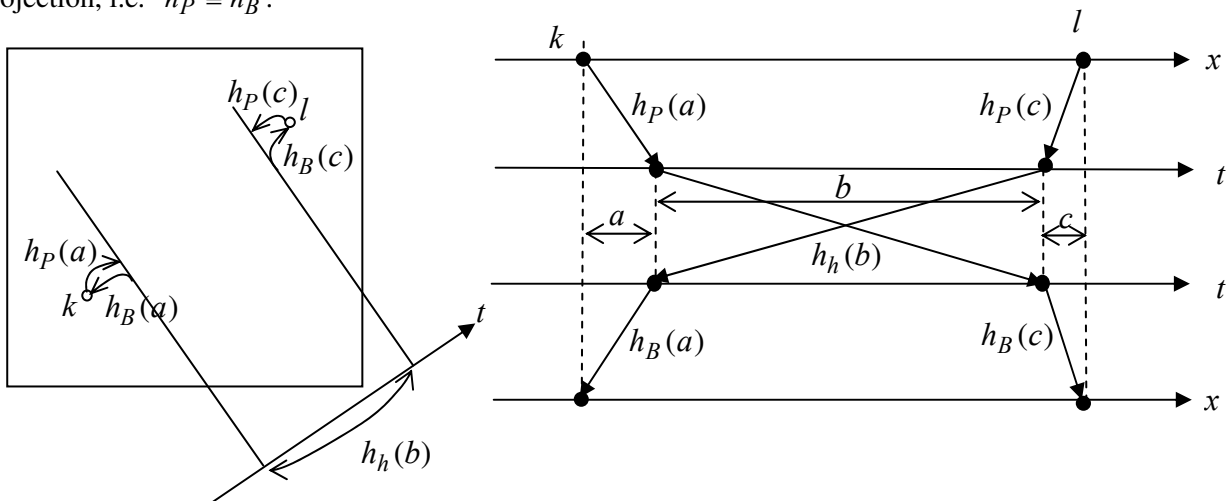


Fig.5. 2 The contributions $(QP)_{kl}$ and $(QP)_{lk}$ to point-spread functions Q_k and Q_l , respectively

In some of our own experiments [] we have deliberately employed two different interpolation techniques for projection and back-projection for the following reason. Knowing that after convergence, the result $P^{-1}\mathbf{b}$ depends on the projector P only, and not on the filter H nor the back-projector B , we like to simplify the latter operation as much as possible and back-projection usually employs simple linear interpolation. As will be described in later sections, in poorly designed interpolation methods, for projection rays are running exactly in the 45° -direction, the signal value distributed from one ray along its path may exceed the average with as much as 10% while another parallel ray may deliver 10% less than the average. For other angles the situation is much more equalized, which means that the final asymmetry in the QP -matrix will be much less than $\pm 10\%$. At the time of this writing, we feel that employment of different interpolation functions for forward and back-projection will upset the symmetry only marginally and that the previous conclusions on convergence still hold. More that perfect or near ideal interpolation, the symmetry of the QP -matrix is upheld by the symmetry of the ramp-filter h_H .

Non-exact 3D-reconstruction from cone-beam data, for instance the PI- method, comprises very much the same three basic steps of (forward) projection, ramp-filtering, and back-projection, although forward projection is there only when the method is extended with the iterative loop of Fig. 1.1. A major difference, however, is that the two projection rays, related to pixels k and l in Fig.5.2, are no longer in one plane. And from one view to the next the pixel k might “lose contact” with pixel l , i.e. all potential pathways have zero-weights. This is in stark contrast to the planar 2D-case, where each view brings about non-zero weighted connections, links, between any two pixels and contributes to the total matrix component $(QP)_{kl} = (QP)_{lk}$. However, for the discussion on partial contributions via identical paths between the two image points, the non-planar situation is irrelevant and Fig.5.2 is still valid. The symmetry argument still holds.

6. Input data. The Ramp-filter

Compared to the relatively low resolution images that we aim for in CT, the X-rays themselves are pencil-sharp and would be able to deliver projection data with very high spatial frequencies. However, the beam spot is not infinitely small, nor is the detector aperture. In addition, the movement of the gantry during exposure smears the view. Hence, already where measurements are captured and digitized the natural infinite bandwidth is compromised. Even so, to control the bandwidth and avoid corrupting aliasing effects at the time of data capture might be an important task. Exactly how this is done seems to be propriety information and not disclosed by the manufacturers. Therefore, discussions on initial bandwidth limitation has to be forsaken in this treatise.

In our experiments we are manufacturing our own projection data \mathbf{p} from mathematical phantoms. Our rays are line integrals that are just as pencil-sharp as the real X-rays and in principle, a dense set of rays is able to register arbitrary high frequency components. To somewhat mimic the fact that detector aperture Δ_d , is not infinitely small we may compute a detector value as a weighted sum of *several incoming rays*, for instance five. Given an image sampling density of Δ_x^{-1} in the main directions, the output image band-width is given and the ramp-filter should cut-off at the frequencies $\tau = \pm \frac{1}{2} \Delta_x^{-1}$. To avoid some of the potential aliasing problems already at the input, before or during the down-sampling to fit the detector aperture Δ_d we should apply a *low-pass filter*. In the most general case the detector aperture could be different from the detector sampling distance but for simplicity, we assume here that the sampling density along the detector axis $\Delta_t^{-1} = \Delta_d^{-1} \leq \Delta_x^{-1}$. In the Fourier domain an ideal low-pass filter is the rectangle function $\Pi(\tau \Delta_x)$, which in the signal domain is bound to produce unwanted ringing effects of type Gibbs phenomenon, see Bracewell (p 209). A better alternative is then the function

$$G(\tau) = \cos(\pi \tau \Delta_x) \Pi(\tau \Delta_x) \quad (6.1)$$

See Fig.6.1. A rectangle function $\Pi(\tau \Delta_x)$ becomes the function $\Delta_x^{-1} \text{sinc}(t \Delta_x^{-1})$ in the signal domain and the cosine function becomes a pair of Dirac-pulses. In the signal domain the filter $g(t)$ yields

$$\begin{aligned}
 g(t) &= \frac{1}{2} \left[\delta\left(t - \frac{\Delta_x}{2}\right) + \delta\left(t + \frac{\Delta_x}{2}\right) \right] * \text{sinc}\left(t \Delta_x^{-1}\right) = \frac{1}{2} \text{sinc}\left(\Delta_x^{-1}\left(t - \frac{\Delta_x}{2}\right)\right) + \frac{1}{2} \text{sinc}\left(\Delta_x^{-1}\left(t + \frac{\Delta_x}{2}\right)\right) = \\
 &= \frac{1}{2} \frac{\sin \frac{\pi t}{\Delta_x} \cos \frac{\pi}{2} - \cos \frac{\pi t}{\Delta_x} \sin \frac{\pi}{2}}{\frac{\pi t}{\Delta_x} - \frac{\pi}{2}} + \frac{1}{2} \frac{\sin \frac{\pi t}{\Delta_x} \cos \frac{\pi}{2} + \cos \frac{\pi t}{\Delta_x} \sin \frac{\pi}{2}}{\frac{\pi t}{\Delta_x} + \frac{\pi}{2}} = \frac{2}{\pi} \frac{\cos \frac{\pi t}{\Delta_x}}{1 - \left(\frac{2t}{\Delta_x}\right)^2}
 \end{aligned} \tag{6.2}$$

The convolution kernel $g(t)$ is seen to have minima at $t = \frac{3}{2}\Delta_x, \frac{5}{2}\Delta_x, \frac{7}{2}\Delta_x$, etc. The side lobes amplitudes at $t = 2\Delta_x, 3\Delta_x, 4\Delta_x$, etc are $\frac{2}{15\pi}, \frac{2}{35\pi}, \frac{2}{63\pi}$, etc. More generally, at $t = n\Delta_x$ the amplitude is $\frac{2}{\pi} \frac{1}{(2n)^2 - 1}$.

Thus, this bandwidth limitation could be implemented in the Fourier domain by transforming the over-sampled projections, multiply them with $G(\tau) = \cos(\pi\tau\Delta_x)\Pi(\tau\Delta_x)$ and transform the truncated result back to the signal domain. However, it might be just as convenient to produce a band-limited and down-sampled projection by using the function $g(t)$, or rather a smoothly truncated version thereof, as interpolation kernel in the above-mentioned domain down-sampling process.

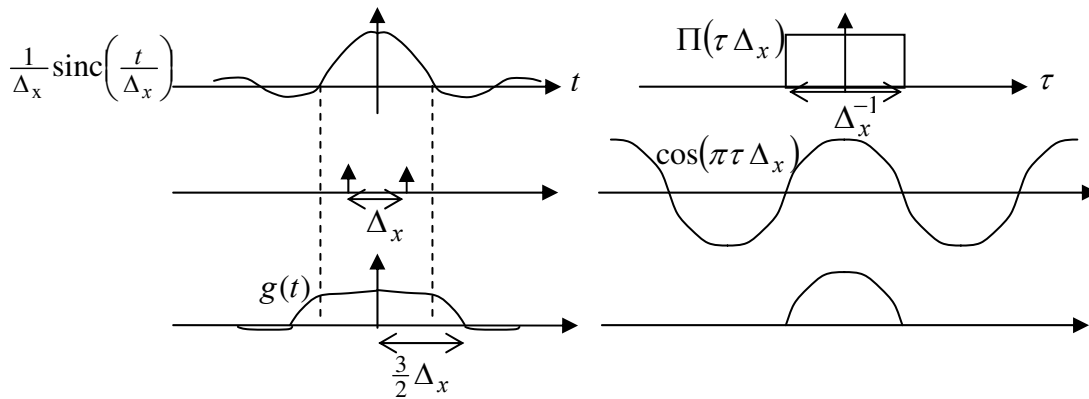


Fig 6.1. Band-limiting with a cosine function is equivalent to convolving the over-sampled projection with the function $g(t)$

Fig.6.2 below shows that we intend to employ the low-pass-filter only at input data. Inside the iterative loop we want to curb the high frequencies inside the pass-band as little as possible. After the band-limiting procedures input data are subjected to ramp-filtering, which includes ideal (razor sharp) band-limiting. In practice, *operator-selected filters* can be used to modify the ideal ramp-filter to yield different image qualities in terms of rotation-invariant frequency response. Various tissues and organs have different needs, become easier to read and interpret if the balance between high and low frequencies is set in a specific way. As a case in question, to reveal the smallest details, in images containing high contrast objects like bones, the highest possible frequencies should be made available with a minimum of attenuation. On the other hand, low contrast objects in soft tissue might be severely disturbed by ringing from nearby bone tissue unless the high frequency band is curbed smoothly. We abstain from using such operator or organ specific filters in our experiments.

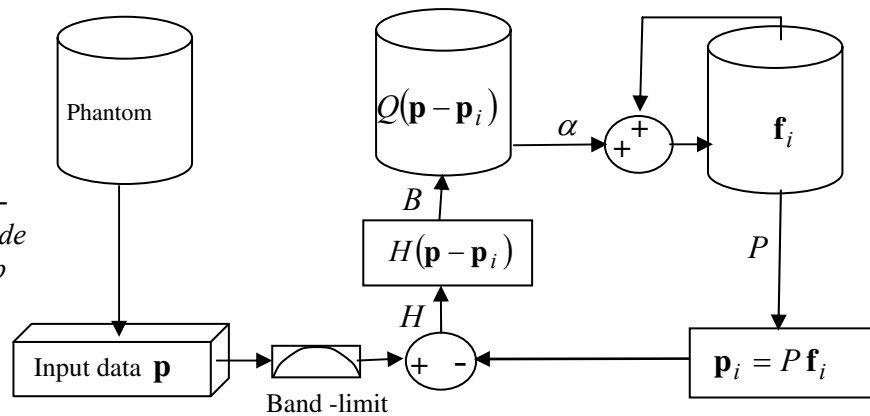


Fig. 6.2. Input data band-limiting takes place outside the iterative loop

Assume that the reconstruction should deliver an image of diameter $N_x \Delta_x$, where N_x is an integer power of 2, and Δ_x is the sampling distance in both x- and y-direction. Initially, from the ideal ramp-filter given as a continuous function $|\tau|$ in the Fourier domain we may create a discrete filter in the signal domain by band-limiting $|\tau|$ to the interval $\left[-\frac{1}{2}\Delta_x^{-1}, \frac{1}{2}\Delta_x^{-1}\right]$, and give it a repetition period Δ_x^{-1} . This is the function $H_1(\tau)$ which we write formally as

$$H_1(\tau) = \Pi(\tau \Delta_x) |\tau| * \text{III}(\tau \Delta_x) \quad (6.3)$$

We apply the inverse (continuous) Fourier transform \mathcal{F}^{-1} to obtain

$$h_1(t) = \mathcal{F}^{-1}[H_1(\tau)] = \Delta_x^{-1} \text{III}(t \Delta_x^{-1}) \Delta_x^{-1} \left[\frac{1}{2} \text{sinc}(t \Delta_x^{-1}) - \frac{1}{4} \text{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] \quad (6.4)$$

The function $h_1(t)$ is *the* ramp-filter, the one that is found e.g. in Kak-Slaney [17], Figure 3.15. We notice that $h_1(t) = 0$ for $t = 2\Delta_x, 4\Delta_x, 6\Delta_x, \dots$ and that this function extends to infinity. Note that $h(t) \Leftrightarrow H_1(\tau)$ form a continuous Fourier transform pair. The $H_1(\tau)$ function still features the rampfilter in a repeated version. We are now going to produce another ramp-filter $h_3(t)$ with a limited extension in the signal domain and a sampling density that is equal or higher than the basic sampling density in the image, i.e. $\Delta_t^{-1} \geq \Delta_x^{-1}$. But Δ_t^{-1} is also bounded upwards so that

$$\Delta_x^{-1} \leq \Delta_t^{-1} \leq 2\Delta_x^{-1} \quad (6.5)$$

Let the number of sampling points in the signal domain be N_l so that the total extension, the width of the filter is $N_l \Delta_t$. There are two requirements on N_l , namely that

- i) N_l is the smallest power of two that complies with (6.5), and
- ii) The width is larger than the double image width, i.e.

$$2N_x \Delta_x \leq N_l \Delta_t \quad (6.6)$$

These two conditions and the fact that we already assume that N_x is a power of two yields that in the normal case where $\Delta_t^{-1} > \Delta_x^{-1}$ we must require that

$$2N_x < N_l = 4N_x \quad (6.7)$$

from which follows that

$$N_l = 4N_x \frac{\Delta_t}{\Delta_x} \Leftrightarrow N_l \Delta_t = 4N_x \Delta_t^2 \Delta_x^{-1} \quad (6.8)$$

Under these premises we begin the actual filter design by defining

$$H_2(\tau) = \text{III}(\tau \Delta_t) * \Pi(\tau \Delta_x) |\tau| \quad (6.9)$$

The inverse (continuous) Fourier transform delivers

$$h_2(t) = \mathcal{F}^{-1}[H_2(\tau)] = \Delta_t^{-1} \text{III}(t \Delta_t^{-1}) \Delta_x^{-1} \left[\frac{1}{2} \text{sinc}(t \Delta_x^{-1}) - \frac{1}{4} \text{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] \quad (6.10)$$

The differences between the Fourier pairs $h_1(t) \Leftrightarrow H_1(\tau)$ and $h_2(t) \Leftrightarrow H_2(\tau)$ are easy to observe in Fig.6.3. Note that the filter coefficients, the discrete sample values in $h_1(t)$ and $h_2(t)$ are no longer identical. In $h_2(t)$ every third coefficient is zero. Where $h_1(t)$ has one single negative coefficient, $h_2(t)$ has one positive and one negative although their sum yields a negative sum, as we see from excerpts in the following table.

$\Delta_t^{-1} t$	0	1	2	3	4	5	6	7
$\Delta_x^{-2} h_2(t)$	$\frac{1}{4}$	0.0358	-0.1461	0	0.0410	-0.0482	0

Finally, we obtain the filter $h_3(t)$ of the correct length $N_l \Delta_t = 4N_x \Delta_t$ by limiting the extension in the signal domain to N_l samples, i.e.

$$h_3(t) = \Pi(t N_l^{-1} \Delta_t^{-1}) h_2(t) = \Pi(t N_l^{-1} \Delta_t^{-1}) \Delta_t^{-1} \text{III}(t \Delta_t^{-1}) \Delta_x^{-1} \left[\frac{1}{2} \text{sinc}(t \Delta_x^{-1}) - \frac{1}{4} \text{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] \quad (6.11)$$

In the Fourier domain this is equivalent to a convolution with the corresponding sinc-function so that

$$H_3(\tau) = \mathcal{F}\left[\Pi(t N_l^{-1} \Delta_t^{-1}) h_2(t)\right] = N_l \Delta_t \text{sinc}(\tau N_l \Delta_t) * H_2(\tau) = N_l \Delta_t \text{sinc}(\tau N_l \Delta_t) * \text{III}(\tau \Delta_t) * \Pi(\tau \Delta_x) |\tau| \quad (6.12)$$

In Figure 6.3 $H_3(\tau)$ is portrayed as a discrete function. Clearly, with sampling distance $N_l^{-1} \Delta_t^{-1}$ for this function, with a slight abuse of notation, we may also define the DFT-pair

$$H_4(\tau) = \text{DFT}[h_4(t)] = N_l \Delta_t \text{sinc}(\tau N_l \Delta_t) * \Pi(\tau \Delta_x) |\tau| \quad (6.13)$$

$$h_4(t) = \text{DFT}^{-1}[H_4(\tau)] = N_l^{-1} \Pi(t N_l^{-1} \Delta_t^{-1}) \Delta_t^{-1} \Delta_x^{-1} \left[\frac{1}{2} \text{sinc}(t \Delta_x^{-1}) - \frac{1}{4} \text{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] \quad (6.14)$$

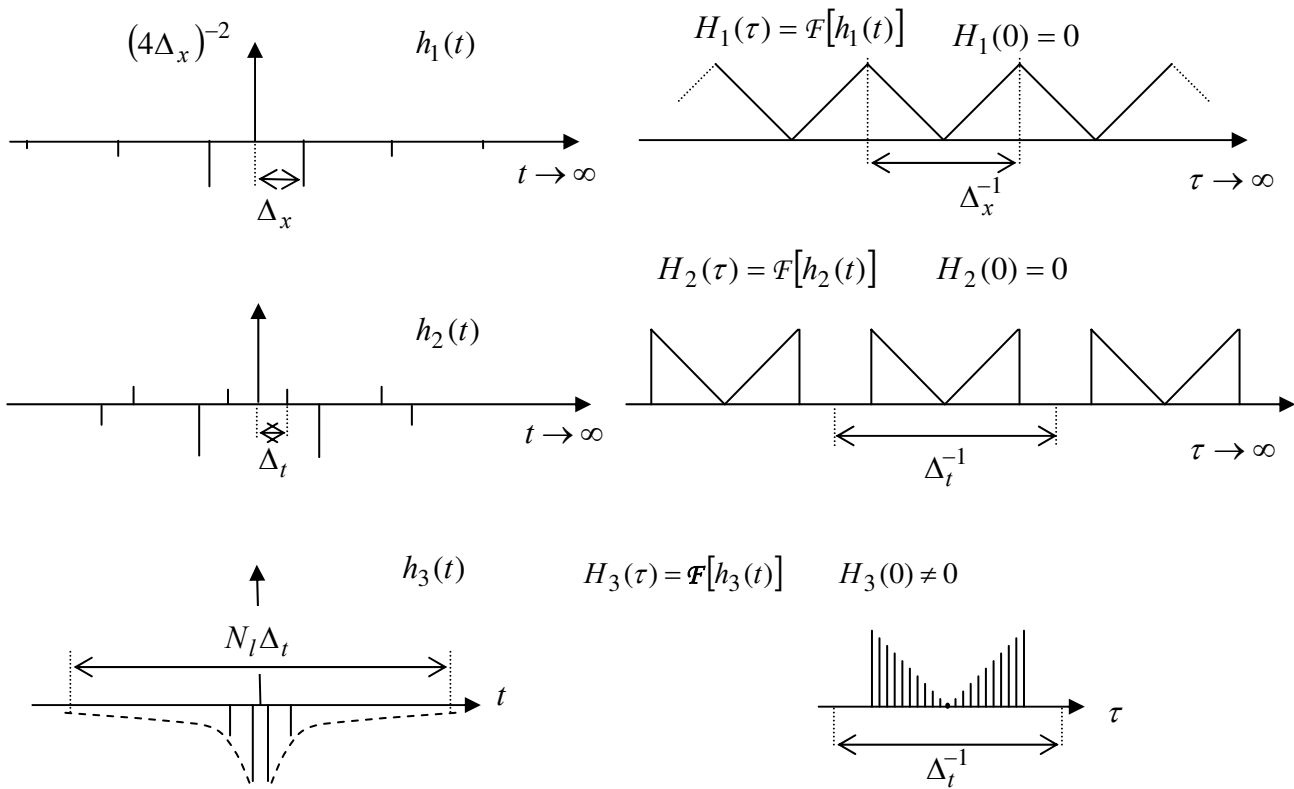


Figure 6.3. Ramp-filter design

Contrary to the fact that $H_1(0) = 0$, the DC-component $H_4(0) < 0$. More precisely we have

$$H_4(0) \approx 2 \int_0^{\frac{1}{2}\Delta_x^{-1}} N_l \Delta_t \operatorname{sinc}(\tau N_l \Delta_t) |\tau| d\tau = \frac{2}{\pi} \int_0^{\frac{1}{2}\Delta_x^{-1}} \sin(\pi \tau N_l \Delta_t) d\tau \quad (6.15)$$

Using the substitution $\sigma = \tau N_l \Delta_t$ we obtain

$$H_4(0) \approx \frac{2}{\pi} (N_l \Delta_t)^{-1} \int_0^{\frac{1}{2}\Delta_x^{-1} N_l \Delta_t} \sin(\pi \sigma) d\sigma = \frac{2}{\pi} (N_l \Delta_t)^{-1} \left[1 - \cos\left(\frac{\pi}{2} N_l \frac{\Delta_t}{\Delta_x}\right) \right] \quad (6.16)$$

With $N_l = 4N_x$ this is

$$H_4(0) \approx \frac{1}{8\pi} (N_x \Delta_t)^{-1} \left[1 - \cos\left(2\pi N_x \frac{\Delta_t}{\Delta_x}\right) \right] \quad (6.17)$$

Example 1. Special case $\Delta_t = \Delta_x$. This is included in the condition (6.5). Thus we may use $N_l = 2N_x$, by which (6.14) reduces to

$$h_4(t) = \mathcal{F}^{-1}[H_2(\tau)] = \text{III}\left(t \Delta_x^{-1}\right) \frac{1}{4} \Delta_x^{-2} \left[2\operatorname{sinc}\left(t \Delta_x^{-1}\right) - \operatorname{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] \quad (6.14a)$$

so that

$$h_4(t) = \mathcal{F}^{-1}[H_2(\tau)] = \text{III}\left(t \Delta_x^{-1}\right) \frac{1}{4} \Delta_x^{-2} \left[2\operatorname{sinc}\left(t \Delta_x^{-1}\right) - \operatorname{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] = \begin{cases} \frac{1}{4} \Delta_x^{-2} & \text{for } t = 0 \\ -\left(\pi t^2 \Delta_x\right)^{-2} & \text{for } t = 1, 3, 5, \dots \\ 0 & \text{for } t = 2, 4, \dots \end{cases} \quad (6.14b)$$

This is the case illustrated in Kak-Slaney, Chapter 3. Eq. (6.17) reduces to

$$H_4(0) = \frac{2}{\pi} N_x^{-1} \quad (6.17a)$$

An alternative way to compute $H_4(0)$ is based on the fact that the DC-component $H_2(0) = 0$. It follows then that not only is the sum of all function values in $h_4(t)$ also zero, but also that the sum of the truncated values in $h_4(t)$ (all negative) amounts to $-H_4(0)$. The length of the filter in the signal domain is $N_l \Delta_t = 2N_x \Delta_t$. Hence, with $N_l = 2N_x$ from (6.14.b) we obtain

$$H_4(0) = 2 \sum_{k=N_x}^{\infty} (\pi \Delta_x)^{-2} \left(\frac{1}{2k+1}\right)^{-2} \approx 2(\pi \Delta_x)^{-2} \int_{t=N_x}^{\infty} \frac{1}{2} t^{-2} dt = (\pi \Delta_x)^{-2} \frac{1}{2} N_x^{-1} \quad (6.17b)$$

where the factor 2 is due to the identical two halves of $h_2(t)$. Unfortunately, the formulas (6.17a) and (6.17b) differ by a factor $4\pi \Delta_x^2$.

Example 2. $\Delta_t = \frac{2}{3} \Delta_x$, $N_x = 128$, $N_l = 4N_x$, $\Delta_x = 1$.

Eq. (6.17a) implies $H_4(0) = \frac{1}{8\pi} \frac{1}{128} \frac{3}{2} \left[1 - \cos\left(2\pi 128 \frac{2}{3}\right) \right] \approx 4 \cdot 10^{-4} \left[1 + \frac{1}{2} \right] = 6 \cdot 10^{-4}$

Eq. (6.17b) implies $H_4(0) \approx 4 \cdot 10^{-4} \Delta_x^{-2}$

Example 3. (Siemens) $\Delta_t = \frac{16}{21} \Delta_x$, $N_x = 512$, $N_l = 4N_x$

Eq. (6.17) implies $H_4(0) = \frac{1}{8\pi} \frac{1}{512} \frac{21}{16} \left[1 - \cos\left(2\pi 512 \frac{16}{21}\right) \right] \approx 1.2 \cdot 10^{-4} \cdot [1 + 0.733] = 2 \cdot 10^{-4}$

Example 4. Destructive truncation. In a series of experiments we have investigated potential convergence problems of iterative FBP for cases where the discrete filter function $h_3(t)$ is first limited as before to the length $N_l \Delta_t$ and then truncated or rather zeroing all values for $|t| \leq N_c$ (as in cut) and the subject the N_l samples as a discrete limited length signal. Thus, this operation yields the filter $h_4(t)$

$$h_4(t) = \Pi[t(N_c \Delta_t)] h_2(t) = \Pi[t(N_c \Delta_t)^{-1}] \text{III}\left(t \Delta_t^{-1}\right) \frac{1}{4} \Delta_x^{-2} \left[2 \text{sinc}(t \Delta_x^{-1}) - \text{sinc}^2\left(\frac{1}{2} t \Delta_x^{-1}\right) \right] \quad (6.18)$$

We apply a DFT on the N_l samples in the interval $\left[-\frac{1}{2} N_l \Delta_t, \frac{1}{2} N_l \Delta_t\right]$ to obtain

$$H_4(\tau) = \mathcal{F}[h_4(t)] = \text{III}(\tau N_l \Delta_t) \Pi(\tau \Delta_x) \left[\text{sinc}(\tau N_c \Delta_t) * |\tau| \right] \quad (6.19)$$

As in the case of $H_3(\tau)$ we can derive $H_4(0)$ as

$$\begin{aligned} H_4(0) &= 2 \int_0^{\frac{1}{2} \Delta_x^{-1}} \text{sinc}(\tau N_c \Delta_t) |\tau| d\tau = 2(\pi N_c \Delta_t)^{-1} \int_0^{\frac{1}{2} \Delta_x^{-1}} \sin(\pi \tau N_c \Delta_t) d\tau = \\ &= 2(\pi N_c \Delta_t)^{-2} \left[1 - \cos\left(\frac{\pi}{2} N_c \frac{\Delta_t}{\Delta_x}\right) \right] \end{aligned} \quad (6.20)$$

7. Ray- and pixel-driven projection and back-projection

In the present context it may have some interest to recollect the classic algebraic reconstruction technique (ART) due to Kaczmarz [5]. See Fig. 7.1. Let the image $f_s(x_k, y_l)$ be sampled in Cartesian points denoted (x_k, y_l) , or simply (k, l) and let the parallel projection values produced by line integrals perpendicular to the angle θ be sampled at discrete points t_m , or simply m , along the detector axis t . (We apologize for changing the meaning of k and l from arbitrary pixel indices [in the previous sections] to grid points at x - and y -axes, respectively [in the following].) In ART projection and back-projection are traditionally executed in the following way, where the weight factors $w(k, l, m, \theta)$ are coefficients obtained as the common area for the square-shaped pixel (k, l) and the rectangular swath defined by the detector element m .

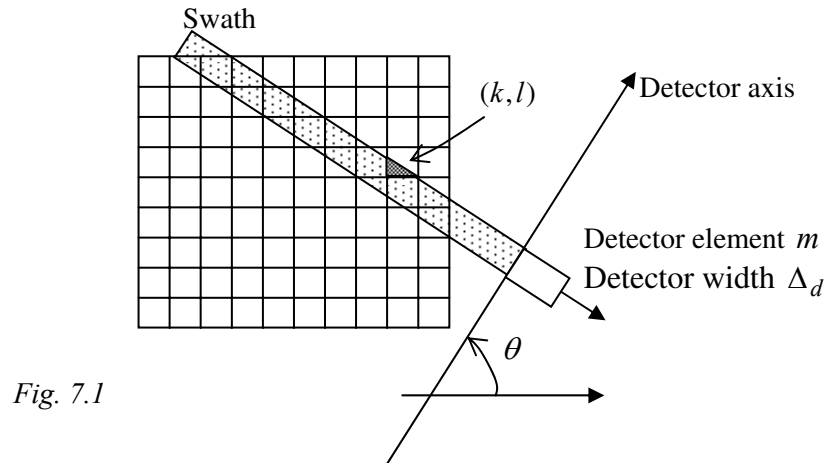


Fig. 7.1

Projection:
$$p_{m,\theta} = \sum_{k,l} w(k, l, m, \theta) f_s(k, l) \quad (7.1)$$

Back-projection
$$g_s(k, l) = \sum_{m,\theta} w(k, l, m, \theta) p_{m,\theta} \left(\sum_{k,l} w^2(k, l, m, \theta) \right)^{-1} \quad (7.2)$$

The normalization factor in (7.2) is computed as a summation of the squared non-zero weights the ray-sum $p_{m,\theta}$ will make use of when picking up contributions in (7.1). Evidently, if this ray-sum is pumped back into the image using (7.2) each single weight $w(k,l,m,\theta)$ comes in twice. Because of the given normalization in (7.2) the ray-sum be delivered back to the image in full and in the same proportion, which preserves numerical stability. Thus the projection and back-projection are using the same weights $w(k,l,m,\theta)$ to define the dependencies (linkages, couplings) between a ray and a pixel.

An alternative way would be to split the normalization equally and symmetrically between the two steps as in (7.3) and (7.4) which means that both steps are employing L_2 -normalization, which was one of the assumptions we used in the convergence analysis in Section 3 above.

Projection
$$p_{m,\theta} = \sum_{k,l} w_1(k,l,m,\theta) f_s(k,l) \left(\sum_{k,l} w_1 w_2(k,l,m,\theta) \right)^{-1/2} \tag{7.3}$$

Back-projection
$$g_s(k,l) = \sum_{m,\theta} w_2(k,l,m,\theta) p_{m,\theta} \left(\sum_{k,l} w_1 w_2(k,l,m,\theta) \right)^{-1/2} \tag{7.4}$$

Both projection and back-projection can be implemented in ray-driven or pixel-driven fashion, the two of which only differ in the order of which the contributions are accumulated. See Fig.7.2. According to most authors, e.g. [1], in *pixel-driven projection*, for a certain pixel (k,l) the innermost loop finds all the non-zero receiving rays (m,θ) , computes (or fetches pre-computed) proper weights $w(k,l,m,\theta)$, and accumulates the contributions to the corresponding detector values $p(m,\theta)$. Thereafter, the inner loop is devoted to another pixel coordinate. In Fig 7.2 a), we notice how the interpolation function $w(t)$ is visualized as resting on the t -axis to compute the proportions in which the incoming pixel value shall be distributed to (in this case) two detector sampling points. In the *ray-driven projection* shown in Fig.7.2 b) we instead visualize the interpolation function $w(t)$ as sliding on the ray while weighting and accumulating contributions along the way and finally delivering the total result as one final sample on the detector axis. Clearly, the result is the same as in the pixel-driven case as long as the same function w is employed. **Since the choice of pixel-driven or ray-driven mode is totally irrelevant to the result**, in the sequel we refrain from mentioning whether a certain projection or back-projection operation is to be implemented as pixel- or ray-driven. What matters is the interpolation function $w(t)$.

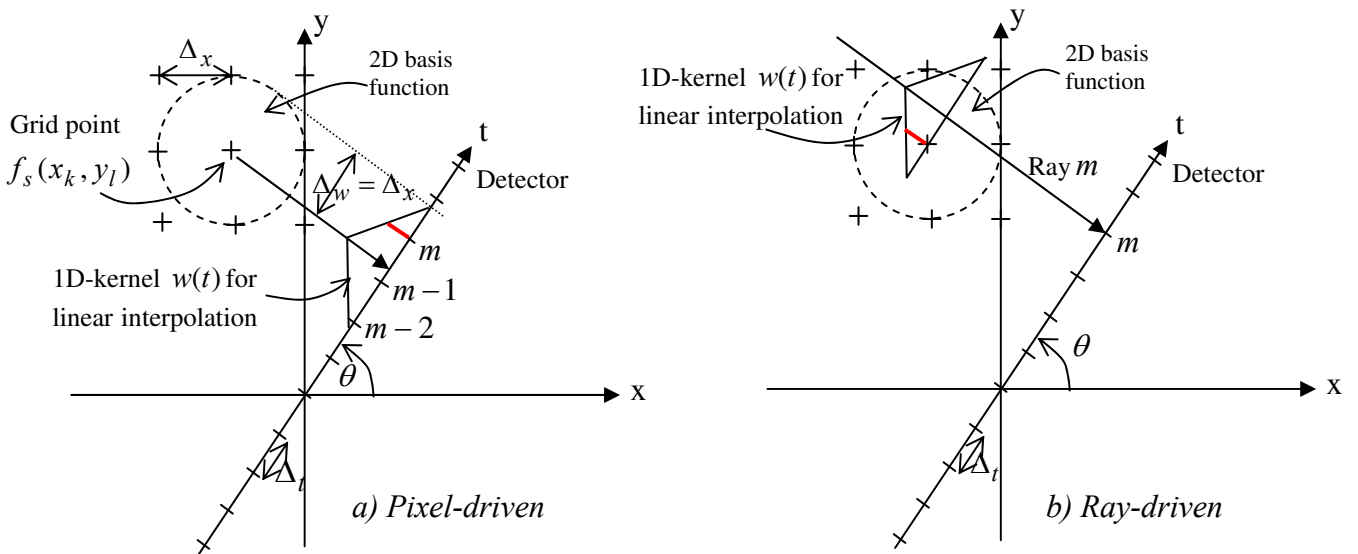


Figure 7.2. Pixel- and ray-driven projection with linear interpolation and a circular symmetric basis function

The coefficients $w(k,l,m,\theta)$ are derived very differently in Fig.7.1 and Fig.7.2. In Figure 7.1 area computations are necessary to retrieve each coefficient. To do this on the fly (on demand) is likely to be rather cumbersome and time consuming. Unfortunately, using pre-computed coefficients might also be a problem, since all four numbers k,l,i,θ are of the same order (normally 512) which makes the total number of coefficients $O(N^4) \approx 10^{11} - 10^{12}$. This constitutes a significant memory requirement for high-speed electronic hardware. In Fig 7.2 the weight factors $w(k,l,m,\theta)$ are obtained with a simple interpolation kernel, something which is much more amenable to fast computation.

Let us now take Fig.7.2 b) and reverse the arrows to obtain the ray-driven projections in Fig 7.3 a). Here, the projection rays are running vertically retrieving contributions from neighboring pixels located in the crossings between horizontal and vertical lines. These contributions are computed by linear interpolation, which is equivalent to convolve the sampled image with a triangle function. Let all grid points carry the same pixel value = 1. Evidently, regardless of its horizontal displacement, for each vertical unit distance $\Delta_y = \Delta_x$ any projection ray running in the vertical direction will obtain the contribution 1 (= unity). In Fig. 7.3 b) the situation is different. Here we obtain the following contributions per unit distance along the rays.

$$\text{For } p_1 : \quad \frac{1}{\sqrt{2}} \left(1 + 2 \left(1 - \frac{1}{\sqrt{2}} \right) \right) = \frac{3}{\sqrt{2}} - 1 = 1.12132 \quad (7.5)$$

$$\text{For } p_2 : \quad \frac{1}{\sqrt{2}} \left(2 \left(1 - \frac{1}{2\sqrt{2}} \right) \right) = \sqrt{2} - \frac{1}{2} = 0.914221 \quad (7.6)$$

Evidently, the projection in the 45° -direction is modulated by a false and unwanted signal with a basic period of $\frac{1}{\sqrt{2}} \Delta_x$ and an amplitude of about 10 % of the DC-component. We will refer to this effect as DC-aliasing and in subsequent sections we will notify its existence in the Fourier domain.

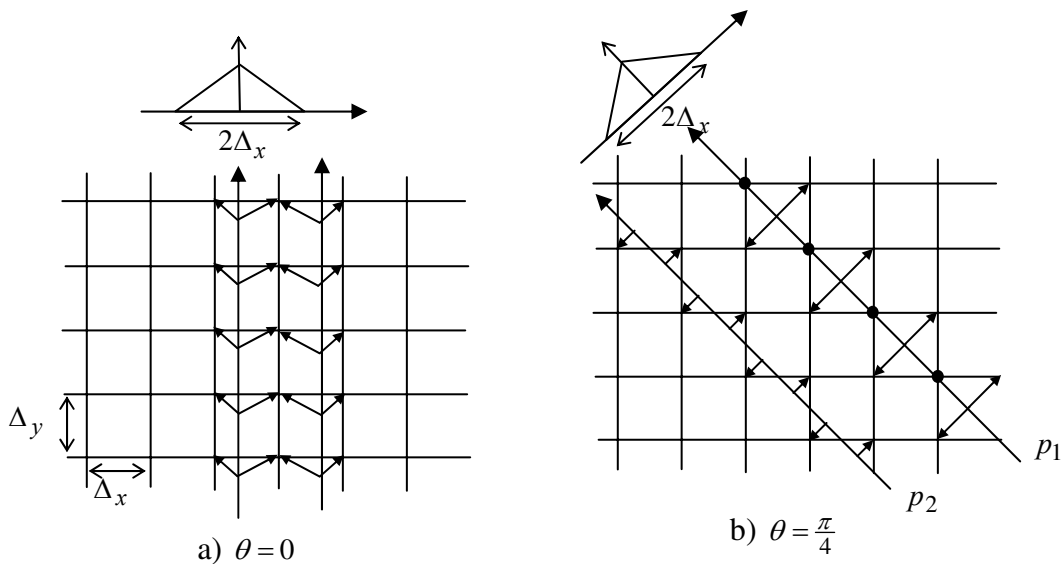


Figure 7.3 Ray-driven back-projection using the same interpolation as in Fig. 7.2 b)

8. Basis and window functions

To generalize and analyze the effect of interpolation in projection and back-projection we must understand that *line integrals along rays in a digital (sampled) image $f_s(x,y)$ can only be defined as line integrations in the underlying continuous function $f(x,y)$* . The sampled and the continuous functions are related via the *basis function $b(x,y)$* as

$$f(x, y) = b(x, y) * f_s(x, y) \quad (8.1)$$

where $f_s(x, y) = f(x, y) \Delta_x^{-1} \text{III}(x \Delta_x^{-1}, y \Delta_x^{-1})$

We now need the following Lemma on convolution and projection.

Lemma: The two operators convolution (*) and projection (\mathcal{R}_θ = line integration perpendicular to the angle θ) commute in the following sense. Given two functions $b_1(x, y)$ and $b_2(x, y)$, the following equality holds for their projections $\mathcal{R}_\theta b_1$ and $\mathcal{R}_\theta b_2$.

$$\mathcal{R}_\theta [b_1 * b_2] = \mathcal{R}_\theta b_1 * \mathcal{R}_\theta b_2 \quad (8.2)$$

Proof: The Fourier slice theorem tells us that projection of an image is to extract a radial slice of the Fourier transform of the image. Hence, we have two 1-1-correspondences between the two domains, namely that the projection operator (\mathcal{R}_θ) corresponds to the slicing operator (S_θ), and that the convolution operator corresponds to multiplication. Therefore, the Fourier transform of the left hand side of (8.2) yields $S_\theta [B_1 \cdot B_2]$ while the right hand side yields $S_\theta B_1 \cdot S_\theta B_2$. Therefore, since it is obvious that

$$S_\theta [B_1 \cdot B_2] \equiv S_\theta B_1 \cdot S_\theta B_2 \quad (8.3)$$

we also have (8.2). **End of proof**

Following Lewitt's notation [3], let us define the *window function* $w(t)$ corresponding to the basis function $b(x, y)$ as the Radon transform

$$w_\theta(t) \equiv \mathcal{R}_\theta b(x, y) \quad (8.4)$$

where the window function in the general case is different for different angles θ . We notice that for rotationally symmetric basis functions, the Radon transform can be replaced by the Abel transform. In the literature, this function goes under various other names, like profile, footprint, etc. Applying Lemma (8.2) and (8.4) to (8.1) yields

$$\mathcal{R}_\theta f(x, y) = \mathcal{R}_\theta b(x, y) * \mathcal{R}_\theta f_s(x, y) = w_\theta(t) * \mathcal{F}^{-1} S_\theta \mathcal{F} f_s(x, y) \quad (8.5)$$

Hence, we express and define the projection in the direction θ of the digital image as the projection of the underlying continuous image $f(x, y)$ according to the given definition. However, via the Lemma we find that we may also express this as a convolution with the basis function $b(x, y)$, or rather, as a 1D-convolution with the window function $w_\theta(t)$ along the detector axis t . The other component of the convolution is the projection of the sampled image $f_s(x, y)$, which can only be defined as $\mathcal{F}^{-1} S_\theta \mathcal{F} f_s$, i.e. the 1D inverse Fourier transform of a slice through the 2D Fourier transform of $f_s(x, y)$. From (8.5), in the Fourier domain $\mathcal{F} f_s(x, y)$ of the given digital image $f_s(x, y)$ we then have

$$\mathcal{F} \mathcal{R}_\theta f(x, y) = W_\theta(\tau) \cdot S_\theta \mathcal{F} f_s(x, y) \quad (8.6)$$

where $W_\theta(\tau)$ is the 1D-Fourier transform of the window function. A more detailed Fourier domain modeling of the projection operator is given in the following section.

Many forward projection methods for digital images have been proposed in the literature, some of them by defining a certain basis function $b(x, y)$, some of them by defining the corresponding window function $w(t)$, and some of them defined operationally only. A subset of these is portrayed in Fig. 8.1. In upper row, left, we recognize the circular symmetric basis function with a triangle shaped window function that we used in Fig 7.2 and Fig.7.3 without defining the basis function itself. Clearly, however, this basis function is found by applying the inverse Abel function to the triangle function $\Lambda(t \Delta_x^{-1})$ which yields the rotationally symmetric function

$$\Delta_x \pi^{-1} \cosh^{-1}(\Delta_x^{-1} r^{-1}) \Pi(r / 2 \Delta_x^{-1}) \quad (8.7)$$

A number of 3D forward projection methods, including Joseph's, are briefly described in the thesis by Turbell [13]. One of these is due to Siddon [14] who is calculating the weight $w(k, l, m, \theta)$ of a pixel value as the length the ray (i, θ) intersects the basis function $b_{Siddon}(x, y)$, which is constant over a square-shaped area around the sample point (x_k, y_l) . The window function $w_{Siddon}(t)$ that arises as the projection of the given basis function is highly dependent on the angle θ . Joseph's method [9] was originally described as a quick and easy technique to compute the line integrals as follows. If the angle θ is in the interval $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$, as is the case in Fig.8.1, the ray sum is computed as follows. For each horizontal grid-line, a new contribution is found by linear interpolation between the two sampling points nearest to the ray. The width (= half of the total base) of the triangular interpolation kernel measured perpendicular to the ray, i.e. laid out along the detector axis, is $\Delta_w = \Delta_x \cos \theta$ where θ is the smaller of the two difference angles between the detector axis and the x- and y-axes, respectively. The final ray-sum is multiplied with $(\cos \theta)^{-1}$ to compensate for varying step-lengths between two consecutive contributions along the ray, which results in the window function

$$w_\theta(t) = (\cos \theta)^{-1} \Lambda \left[t \Delta_x^{-1} (\cos \theta)^{-1} \right] \tag{8.8}$$

where Λ stands for triangle function. The window function (8.8) is implicit in Joseph's original procedural description. The underlying basis function exists as $\mathcal{R}^{-1}[w_\theta(t)]$ but is never used explicitly.

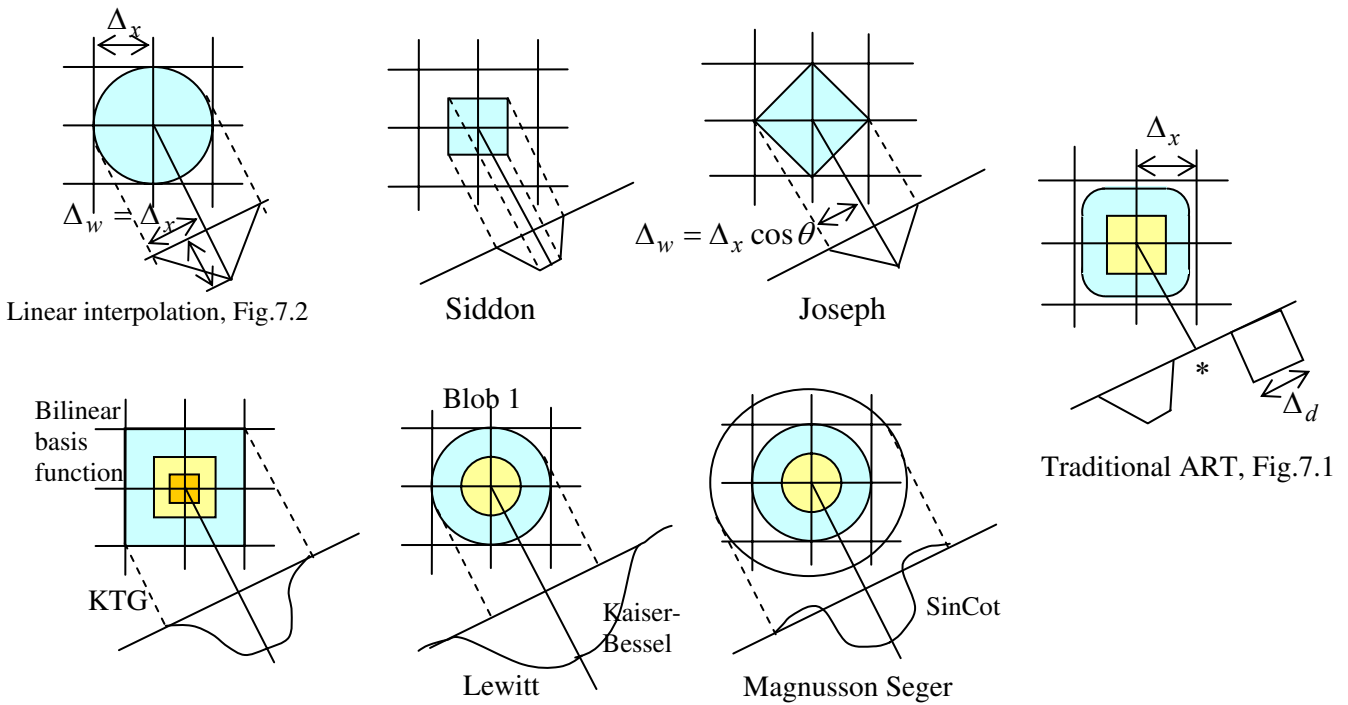


Fig. 8.1 Basis and window functions proposed for computing projections from

Another method presented in [13], [15] is the KTG-method, see also A sample from the underlying continuous function is obtained by bi-linear interpolation in the sampled 2D-image (tri-linear in the 3D-case of [13], [15]). Hence, in the 2D-case the basis function in Fig.8.1 lower left is defined as the product of two triangle functions. From this follows a window function $w(k, l, m, \theta)$ by computing the line integral of the projection ray (m, θ) through this basis function centered at (x_k, y_l) . In [13], [15], however, it is suggested that this line integral is either approximated by sampling and summing the basis function itself repeatedly and not too sparsely at equidistant points along the ray, or preferably, using the Simpson rule of integration. The KTG-method is similar to the Joseph's method in the sense that the underlying 2D-basis function is not rotationally symmetric but symmetric with respect to the Cartesian grid. The experiments in [13], [15] show that Joseph's method generate projections with slightly less errors (less aliasing) than the KTG-method, while Siddon's method is significantly inferior to the other two.

Two of the basis-window-function pairs in the lower row of Fig.8.1 are due to Lewitt [3] and Magnusson Seger [10], [11], [12]. Both are carefully designed to yield a window function, which is limited in the signal domain as well as in the frequency domain, although the latter limitation is not absolute. Typically, the window functions are such that in ray-driven modes, there are between four and six pixel contributions to be accumulated/delivered from/to each ray-value. Like-wise, in pixel-driven modes there are four to six detector contributions to be accumulated/delivered from/to each ray value. In the Fourier domain these window functions are designed to be strongly attenuating outside the Nyquist limits of the sampled image signal, especially outside the double Nyquist limits, where the attenuation typically is -60 DB or less. These window functions will be studied in more detail in subsequent sections.

To the far right is a pictorial description of the traditional basis-window function employed in traditional ART, Fig.7.1. This projection operator is always defined operationally. Therefore, the interpretation in Fig.8.1 and the analytical formulation to be given here might be hard to find in the literature. The operational tells that the window function coefficient $w_{ART\theta}(t)$ is portion of the actual square-shaped pixel area that is covered by a swath of the actual detector, the width of which is the detector aperture Δ_t . This can be expressed by the following two-step procedure.

- 1) Compute the same angular dependent and trapezoid window-function $w_{\theta S}(t)$ as in Siddon.
- 2) Convolve this result with a rectangle function $\Pi(t/\Delta_t)$ to get the window function as

$$w_{ART\theta}(t) = w_{\theta S}(t) * \Pi(t/\Delta_t) = \mathcal{R}_\theta \left[b_{Siddon} * \mathcal{R}^{-1}(\Pi(t/\Delta_t)) \right] = \mathcal{R}_\theta [b_{ART}(x, y)] \quad (8.9)$$

where we have employed the above lemma. Thus, from (8.9) we identify can also extract the basis function for traditional art as

$$b_{ART}(x, y) = \Pi(x/\Delta_x)\Pi(y/\Delta_x) * \Pi(r/\Delta_t) \left(\left(\frac{\Delta_x}{2} \right)^2 - r^2 \right)^{-\frac{1}{2}} \quad (8.10)$$

where the last part of this expression is rotationally symmetric but not identical to the disk $\Pi(r/\Delta_t)$. The window and basis and basis functions of (8.9) and (8.10) are unique in that they depend on the sampling densities of the image as well as of the detector.

9. Fourier modeling of projection and back-projection

In this section we will only deal with 2D-functions of finite extent, an arbitrary continuous version of which we denote $f(x, y)$. In mathematic notation a complete set of forward projections of $f(x, y)$ applied in all directions θ yields the Radon transform of $f(x, y)$, which we may write as

$$p(t, \theta) = \mathcal{R}f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - t) dy dx, \quad (9.3)$$

$$t_{\min} \leq t \leq t_{\max}, \quad -\frac{\pi}{2} < \theta \leq \frac{\pi}{2}$$

For a single projection in the θ -direction we will use the notation

$$p_\theta(t) = \mathcal{R}_\theta f(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - t) dy dx, \quad (9.4)$$

$$t_{\min} \leq t \leq t_{\max},$$

where \mathcal{R}_θ is the (forward) projection operator. The Radon transform is defined for continuous functions. Hence, to treat sampled images analytically we must find an underlying continuous function for which a sampled image is a representation. First, define a sampled version of the continuous $f(x, y)$ as

$$f_s(x, y) = \Delta_x^{-2} III\left(\frac{x}{\Delta_x}\right) III\left(\frac{y}{\Delta_x}\right) f(x, y) = f(x, y) \sum_{k=-\infty}^{\infty} \delta(x - k\Delta_x) \sum_{l=-\infty}^{\infty} \delta(y - l\Delta_x) \quad (9.5)$$

where the two one-dimensional functions III are sampling operators, and Δ_x is the sampling distance in both x- and y-direction. To produce projections of the sampled image, we must interpret each sample value $f_s(x_k, y_l)$ as a coefficient to be multiplied with a 2D *basis function* $b(x, y)$. Then, a continuous function $\tilde{f}(x, y)$ from the given samples is obtained as

$$\tilde{f}(x, y) = b(x, y) * f_s(x, y) = b(x, y) * \Delta_x^{-2} III\left(\frac{x}{\Delta_x}\right) III\left(\frac{y}{\Delta_x}\right) f(x, y) \quad (9.6)$$

The discrepancies between $f(x, y)$ and $\tilde{f}(x, y)$ are due to the unavoidably non-perfect nature of the basis function $b(x, y)$. In any case, given the basis function $b(x, y)$, from (9.6) we obtain the projection of the sampled image $f_s(x_k, y_l)$ just as in (9.4) as

$$p_\theta(t) = \mathcal{R}_\theta \tilde{f}(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{f}(x, y) \delta(x \cos \theta + y \sin \theta - t) dy dx, \quad t_{\min} \leq t \leq t_{\max} \quad (9.7)$$

In the sequel we will use the alternative notation

$$p_\theta(t) = \mathcal{R}_\theta[\tilde{f}(x, y)](t) = \mathcal{R}_\theta[b(x, y) * f_s(x, y)](t), \quad t^2 = x^2 + y^2 \quad (9.8)$$

Using the above in Lemma (8.2) we obtain

$$p_\theta(t) = \mathcal{R}_\theta[b(x, y)] * \mathcal{R}_\theta[f_s(x, y)] \quad (9.9)$$

Replacing $f_s(x, y)$ in (9.9) with the expression in (9.6) yields

$$p_\theta(t) = \mathcal{R}_\theta[b(x, y)] * \mathcal{R}_\theta\left[\Delta_x^{-2} III\left(\frac{x}{\Delta_x}\right) III\left(\frac{y}{\Delta_x}\right) f(x, y)\right], \quad t^2 = x^2 + y^2 \quad (9.10)$$

by which we obtain the Fourier domain expression

$$P_\theta(\tau) = B(u, v) \cdot F_s(u, v) = B(u, v) [III(u\Delta_x) III(v\Delta_x) * F(u, v)] \quad \theta = \arctan \frac{v}{u}, \quad \tau^2 = u^2 + v^2 \quad (9.11)$$

A simplistic Fourier spectrum is shown in Figure 9.1. Repeated copies of $F(u, v)$, the Fourier transform of the underlying continuous image function $f(x, y)$, are indicated by sets of concentric circles. The basis (interpolation) function $B(u, v)$ will modulate these copies so that in $F_s(u, v)$ and its projection slices, only the central copy will appear with full strength while the amplitude of the other copies decrease gradually with increasing distance to the origin. We will study the details of this effect in the sequel. Fig 9.1 is a follow-up of the above Lemma (8.2), which we have found to be an indispensable tool to visualize and analyze projection, (ramp)filtering and back-projection in the same domain.

In **back-projection** the initial Fourier domain of the sampled image is an empty space waiting to be filled in by band-limited projection data. See Fig.9.2. The grid-points are the Fourier transform of the image sampling grid. The sampled and ramp-filtered projection indicated in red consists of repeated copies of the Fourier components inside the Nyquist boundaries. The repetition pitch equals the detector sampling density Δ_t^{-1} in all directions θ .

To see the complementary symmetry between projection and back-projection, let us note that in Fig.9.1 the contributions **from** a slice in $F(u, v)$ **to** a projection $P_\theta(\tau)$ were modulated by the basis function $B(u, v)$, or rather by the window function $W(\tau)$. In Fig.9.2 the contributions **to** a slice in $F(u, v)$ **from** a projection $P_\theta(\tau)$ will be modulated by the same window function $W(\tau)$ while building up a continuous and non-band-limited function. All contributions outside the central Nyquist circle are aliasing. The final sampled $F_s(u, v)$ is obtained by convolving the total accumulated result with the Shah-function in Fig.9.2 represented by the grid-points. The advantage of modeling projection in this seemingly complicated manner is that we can actually visualize the aliasing contributions along each slice and at which spatial frequency position they will appear.

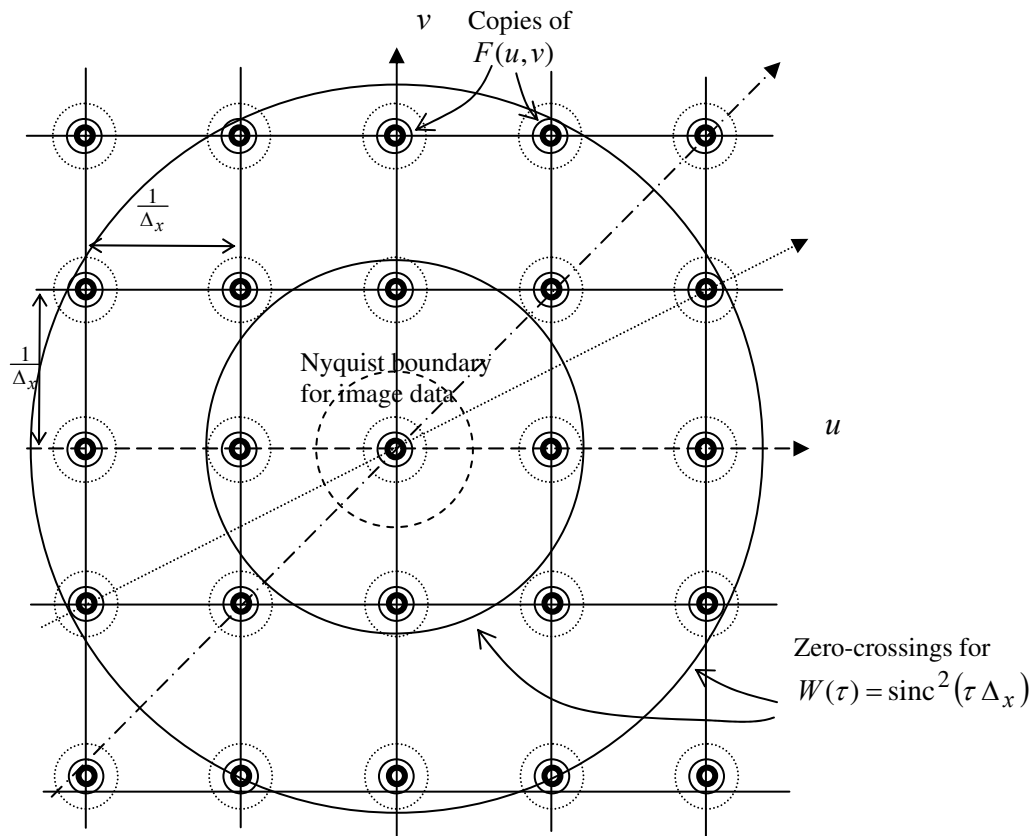


Fig. 9.1. Projection. The Fourier domain $F_s(u, v)$ of the sampled image $f_s(x, y)$ from which projections will be extracted by interpolation in the image $\tilde{f}(x, y) = b(x, y) * f_s(x, y)$. In the Fourier domain this process correspond to slicing the Fourier domain along radial lines while multiplying with the window function $W(\tau)$. Three directions of special interest are indicated, namely $\theta = 0, \theta = \arctan \frac{1}{2}$, and $\theta = \frac{\pi}{4}$.

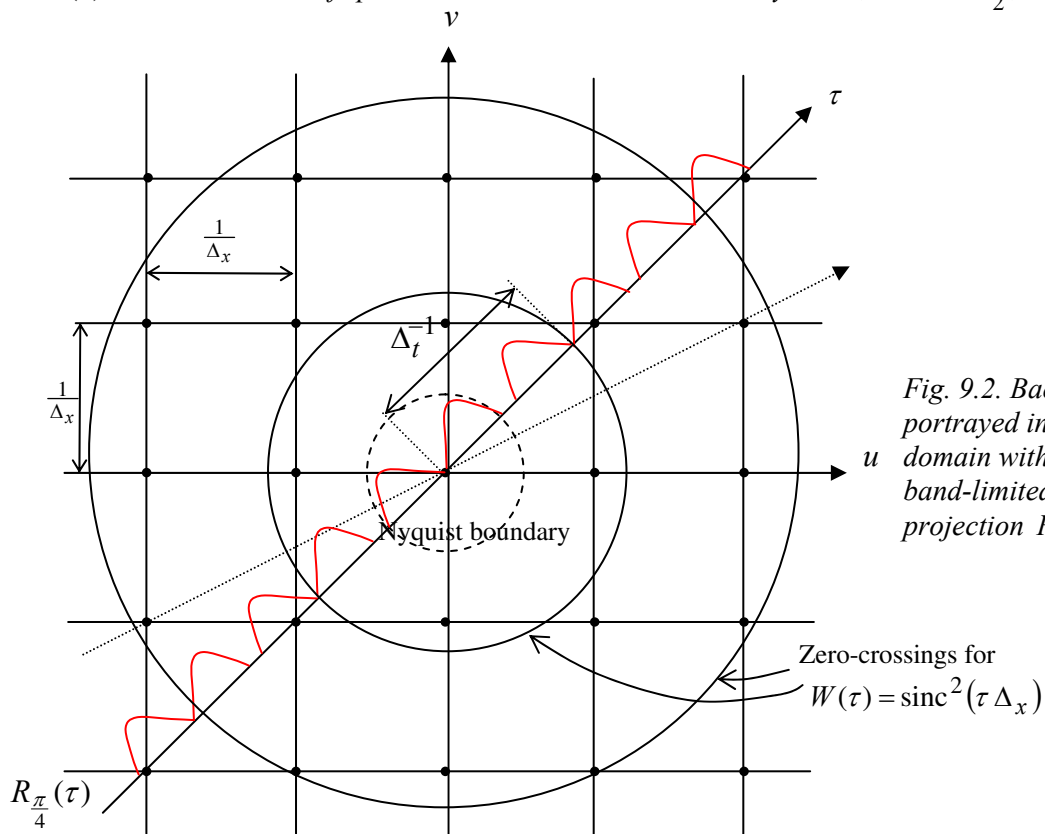


Fig. 9.2. Back-projection portrayed in the Fourier domain with a sampled band-limited ramp-filtered projection $R_{\frac{\pi}{4}}(\tau)$ in red

In subsequent chapters various forms of aliasing, especially so called DC-aliasing, will be discussed and analyzed. The ramp-filtering that precedes the back-projection has almost but not completely eliminated the DC-component in Fig.9.2. Hence, we may expect the danger of DC-aliasing to be alleviated but not eliminated. In fact, practical experiments have shown, however, that DC-aliasing is a problem also in back-projection. Fortunately, the solution is simple, namely to use a window function the Fourier transform of which is zero-valued for $\tau = k\Delta_t^{-1}$. A common window function is the triangle function (linear interpolation). By choosing a width $\Delta_w = \Delta_t$ the suppressing of DC-aliasing is guaranteed.

As we will see, DC-aliasing for *back-projection* stems from a mismatch between the width Δ_w of the interpolation filter and the *detector sampling distance* Δ_t . In *projection*, on the other hand, DC-aliasing stems from a mismatch between Δ_w and certain periods $\Delta_x, \frac{1}{\sqrt{2}}\Delta_x, \frac{1}{\sqrt{5}}\Delta_x$ which are linked to the basic *image sampling distance* Δ_t .

10. Projection aliasing. General formulas

From Fig.9.1 follows that projections in the four directions $\theta = 0, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}$ are unique in the sense that the corresponding slices in the Fourier domain exhibit a perfect periodicity with periods Δ_x^{-1} and $\sqrt{2}\Delta_x^{-1}$, respectively. In certain other directions periodicity with larger periods may also arise, e.g. in the direction $\theta = \arctan \frac{1}{2}$ with the period $\sqrt{5}\Delta_x^{-1}$, $\theta = \arctan \frac{1}{3}$, period = $\sqrt{10}\Delta_x^{-1}$, and $\theta = \arctan \frac{2}{3}$, period = $\sqrt{13}\Delta_x^{-1}$.

All these cases are unique in that the slice-line strikes a DC-component copy not too far from the origin. The longer from origin such a first hit (or near hit) occurs, the smaller is the potential for aliasing. For most directions such an event happens so far from origin that the decay any sensible interpolation function $w(t)$ has nullified the aliasing effect. Therefore, in the sequel we will concentrate our investigation to the two cases $\theta = 0$ and $\theta = \frac{\pi}{4}$, i.e. one main and one diagonal direction. Aliasing effects in directions $\theta = \frac{\pi}{2}$ and $\theta = \frac{3\pi}{4}$, respectively, are obviously the same. We will specifically point out cases of *DC-aliasing* and *Conjugated aliasing*, both of which appear at two specific frequencies $\pm \tau_0$ and $\pm \tau_1 = \frac{1}{2}\tau_0$, respectively.

The case $\theta = \frac{\pi}{4}$ is illustrated in Fig.10.1, where input data from the image appears in the top row, the non-sampled projected result in the middle row, and one of several aliasing contributions appears in the bottom row. We have chosen a detector sampling distance $\Delta_t^{-1} \approx 1.3\Delta_x^{-1}$, which seems to rhyme well with the 672 detector channels in Siemens' Sensation machine since $\frac{672}{512} = 1.3125$. However, the formulas to be developed below do not depend on this relation. Both Δ_x^{-1} and Δ_t^{-1} appear as free parameters. Likewise, the linear interpolation function, which in the Fourier domain in Fig.10.1 is the function $\Delta_x \text{sinc}^2(\Delta_w \tau)$, just one function of many that can be plugged into the formulas below.

Let $p_{\frac{\pi}{4}}(t) = \mathcal{F}^{-1} \left[P_{\frac{\pi}{4}}(\tau) \right]$ stand for the projection of the under-lying continuous band-limited image. The Fourier transform of the projection of the sampled image is a repeated version of $P_{\frac{\pi}{4}}(\tau)$ that we prefer to denote $S_{\frac{\pi}{4}}(\tau)$ shown at the top of Fig.10.1. Clearly, we have

$$S_{\frac{\pi}{4}}(\tau) = \sum_{n=-\infty}^{+\infty} P_{\frac{\pi}{4}}\left(\tau + n\sqrt{2}\Delta_x^{-1}\right) \quad (10.1)$$

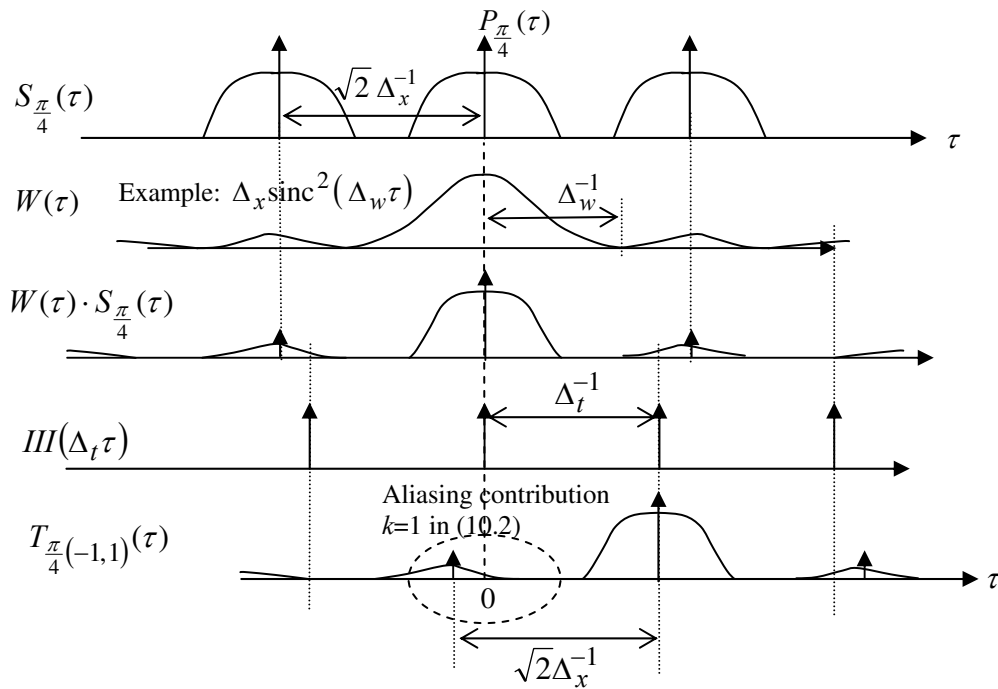


Fig.10.1. Aliasing in the 45° -projection of a sampled image

The interpolated version of $s_{\frac{\pi}{4}}(t) = \mathcal{F}^{-1} \left[S_{\frac{\pi}{4}}(\tau) \right] = \mathcal{F}^{-1} \left[W(\tau) S_{\frac{\pi}{4}}(\tau) \right]$. Let $T_{\frac{\pi}{4}}(\tau)$ be the Fourier transform of the sampled version thereof that yields

$$T_{\frac{\pi}{4}}(\tau) = \sum_{k=-\infty}^{\infty} W(\tau + k\Delta_t^{-1}) S_{\frac{\pi}{4}}(\tau + k\Delta_t^{-1}) = \sum_{k=-\infty}^{\infty} W(\tau + k\Delta_t^{-1}) \sum_{n=-\infty}^{+\infty} P_{\frac{\pi}{4}}(\tau + n\sqrt{2}\Delta_x^{-1} + k\Delta_t^{-1}) \quad (10.2)$$

For any k except $k=0$ an aliasing contribution is likely to appear in the Nyquist interval $|\tau| < \frac{1}{2}\Delta_x^{-1}$. The bottom row of Fig.1.1 shows the following contribution for the term $(k, n) = (-1, 1)$.

$$T_{\frac{\pi}{4}(-1,1)}(\tau) = W(\tau - \Delta_t^{-1}) P_{\frac{\pi}{4}}(\tau + \sqrt{2}\Delta_x^{-1} - \Delta_t^{-1}) \quad (10.3)$$

Correspondingly, for $(k, n) = (1, -1)$ we get an equal and symmetrically located contribution

$$T_{\frac{\pi}{4}(1,-1)}(\tau) = W(\tau + \Delta_t^{-1}) P_{\frac{\pi}{4}}(\tau - \sqrt{2}\Delta_x^{-1} + \Delta_t^{-1}) \quad (10.4)$$

For normal interpolation functions $W(\tau)$ these two contributions are the most prominent ones. They give rise to **DC-aliasing** in $T_{\frac{\pi}{4}}(\tau)$ for

$$\tau = \pm\tau_0 \equiv \pm(\sqrt{2}\Delta_x^{-1} - \Delta_t^{-1}), \quad (10.5)$$

each of which amounts to

$$T_{\frac{\pi}{4}(1,-1)}(\sqrt{2}\Delta_x^{-1} - \Delta_t^{-1}) = T_{\frac{\pi}{4}(-1,1)}(-\sqrt{2}\Delta_x^{-1} + \Delta_t^{-1}) = W(\sqrt{2}\Delta_x^{-1}) P_{\frac{\pi}{4}}(0) \quad (10.6)$$

Conjugated aliasing, defined as the case when $T_{\frac{\pi}{4}(1,1)}(\tau) = T_{\frac{\pi}{4}(1,-1)}^*(-\tau)$, occurs at

$$\tau = \tau_1 \equiv \frac{1}{2}\tau_0 \equiv \frac{1}{2}(\sqrt{2}\Delta_x^{-1} - \Delta_t^{-1}) \quad (10.7)$$

since

$$\begin{aligned}
 T_{\frac{\pi}{4}(-1,1)}(\tau_1) &= W\left(\frac{1}{2}\left(\sqrt{2}\Delta_x^{-1} + \Delta_t^{-1}\right)\right)P_{\frac{\pi}{4}}(-\tau_1) \\
 T_{\frac{\pi}{4}(1,-1)}(-\tau_1) &= W\left(-\frac{1}{2}\left(\sqrt{2}\Delta_x^{-1} + \Delta_t^{-1}\right)\right)P_{\frac{\pi}{4}}(\tau_1)
 \end{aligned} \tag{10.8}$$

and because the projections are real in the signal domain, while the interpolation function $w(t)$ is not only real but even as well. Therefore $P_{\frac{\pi}{4}}(\tau)$ and $P_{\frac{\pi}{4}}(-\tau)$ are conjugated, i.e. $P_{\frac{\pi}{4}}(\tau) = P_{\frac{\pi}{4}}^*(-\tau)$, while $W(\tau) = W(-\tau)$.

In the complete aliased result above in eq. (10.2) the center lobe contribution $(k, n) = (0, 0)$ and the first two side lobe contributions $(k, n) = (-1, 1)$ and $(k, n) = (1, -1)$ to the frequency component at τ_1 yields

$$W(\tau_1) \left[P_{\frac{\pi}{4}}(\tau_1) + P_{\frac{\pi}{4}}(-\tau_1) \right] + W\left(\frac{1}{2}\left(\sqrt{2}\Delta_x^{-1} + \Delta_t^{-1}\right)\right) \left[P_{\frac{\pi}{4}}^*(\tau_1) + P_{\frac{\pi}{4}}^*(-\tau_1) \right] \tag{10.9}$$

Conjugated aliasing as in (10.11) is a potential *problem for the stability of the main loop* because the imaginary part of the frequency components $P_{\frac{\pi}{4}}(\tau)$ and $P_{\frac{\pi}{4}}(-\tau)$ will experience a positive instead of a negative feedback.

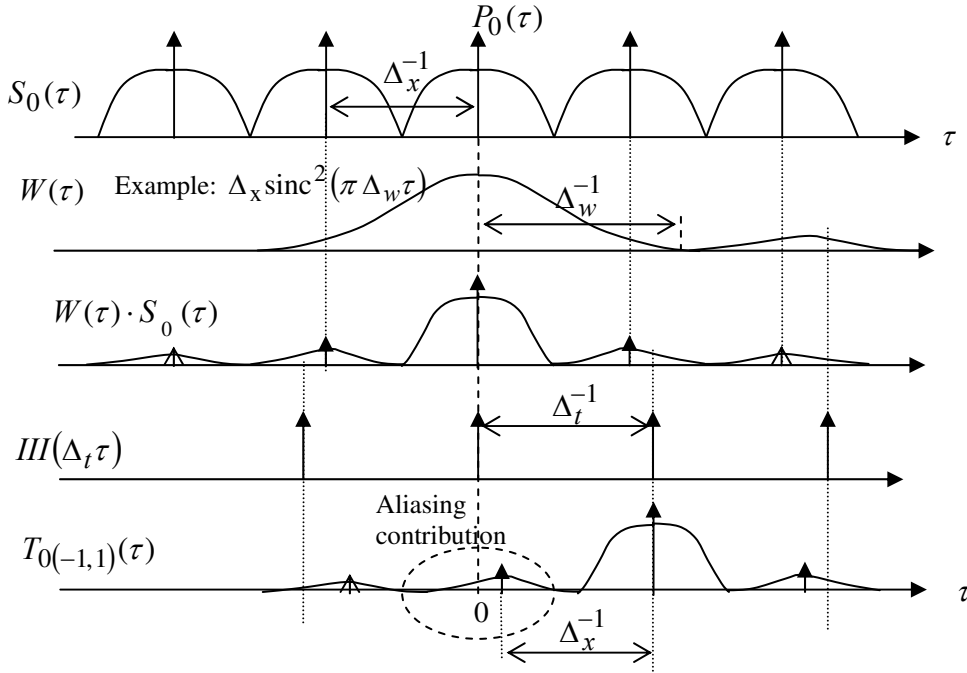


Fig.10.2. Aliasing in the 0^0 -projection of a sampled image

For projection in the direction $\theta = 0$ the aliasing situation is illustrated in Fig.10.2. Eq. (10.3) and (10.4) will now take the following forms.

$$T_{0(-1,1)}(\tau) = W\left(\tau - \Delta_t^{-1}\right)P_0\left(\tau + \Delta_x^{-1} - \Delta_t^{-1}\right) \tag{10.10}$$

and

$$T_{0(1,-1)}(\tau) = W\left(\tau + \Delta_t^{-1}\right)P_0\left(\tau - \Delta_x^{-1} + \Delta_t^{-1}\right) \tag{10.11}$$

respectively. Here, *DC-aliasing* occurs at

$$\pm \tau_0 = \pm\left(-\Delta_x^{-1} + \Delta_t^{-1}\right) \tag{10.12}$$

for which

$$T_{0(-1,1)}(-\Delta_x^{-1} + \Delta_t^{-1}) = T_{0(1,-1)}(\Delta_x^{-1} - \Delta_t^{-1}) = P_0(0) W(\Delta_x^{-1}) \quad (10.13)$$

Conjugated aliasing occurs for the frequencies

$$\tau = \pm \tau_1 \equiv \pm \frac{1}{2}(\Delta_x^{-1} - \Delta_t^{-1}) \quad (10.14)$$

for which

$$\begin{aligned} T_{0(-1,1)}(\tau_1) &= W\left(\frac{1}{2}(\Delta_x^{-1} + \Delta_t^{-1})\right) P_0(-\tau_1) = W\left(\frac{1}{2}(\Delta_x^{-1} + \Delta_t^{-1})\right) P_0^*(\tau) \\ T_{0(1,-1)}(-\tau_1) &= W\left(-\frac{1}{2}(\Delta_x^{-1} + \Delta_t^{-1})\right) P_0(\tau_1) = W\left(\frac{1}{2}(\Delta_x^{-1} + \Delta_t^{-1})\right) P_0^*(-\tau) \end{aligned} \quad (10.15)$$

11. Back-projection aliasing. General formulas

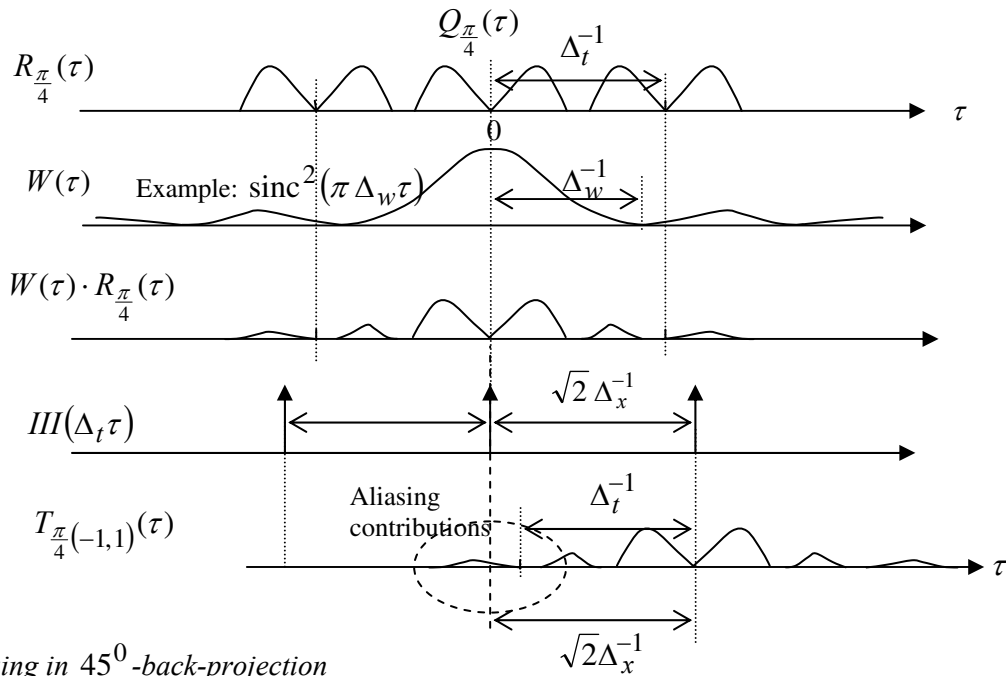


Fig 11.1 Aliasing in 45° -back-projection

Input data in Fig.11.1 is a ramp-filtered projection $Q_{\frac{\pi}{4}}(\tau)$, the repeated version of which is called $R_{\frac{\pi}{4}}(\tau)$. Contrary to forward projection, input data for back-projection are not dominated by a large DC-component. Hence, *DC-aliasing* should not be a major problem in back-projection. In fact, it is common to use $\Delta_w = \Delta_t$ in which case the sinc²-function is zero at $|\tau| = \Delta_t^{-1}$. Then, the DC-aliasing effect in Fig.11.1 would be perfectly suppressed, which is probably the reason why this phenomenon is not mentioned in the literature. However, as indicated in Fig.11.1, when $\Delta_w^{-1} \neq \Delta_t^{-1}$, a small but not quite negligible aliased DC-component will appear in $T_{\frac{\pi}{4}}(-1,1)(\tau)$ at $\tau = \tau_0 = \sqrt{2}\Delta_x^{-1} - \Delta_t^{-1}$ together with other aliased frequency components. In full correspondence to (10.2), the information that is transferred from the projection to the image is given as

$$T_{\frac{\pi}{4}}(\tau) = \sum_{k=-\infty}^{\infty} W\left(\tau + n\sqrt{2}\Delta_x^{-1}\right) R_{\frac{\pi}{4}}\left(\tau + n\sqrt{2}\Delta_x^{-1}\right) = \sum_{k=-\infty}^{\infty} W\left(\tau + n\sqrt{2}\Delta_x^{-1}\right) \sum_{n=-\infty}^{+\infty} R_{\frac{\pi}{4}}\left(\tau + n\sqrt{2}\Delta_x^{-1} + k\Delta_t^{-1}\right) \quad (11.1)$$

In the same manner as for projection we find that *conjugated aliasing* in back-projection appears for the very same frequency

$$\tau = \tau_1 \equiv \pm \frac{1}{2} \left(\sqrt{2} \Delta_x^{-1} - \Delta_t^{-1} \right) \quad (11.2)$$

In the direction $\theta = 0$ the formula (11.1) modifies to

$$T_{\frac{\pi}{4}}(\tau) = \sum_{k=-\infty}^{\infty} W\left(\tau + n\Delta_x^{-1}\right) R_{\frac{\pi}{4}}\left(\tau + n\Delta_x^{-1}\right) = \sum_{k=-\infty}^{\infty} W\left(\tau + n\Delta_x^{-1}\right) \sum_{n=-\infty}^{+\infty} R_{\frac{\pi}{4}}\left(\tau + n\Delta_x^{-1} + k\Delta_t^{-1}\right) \quad (11.3)$$

so that *conjugated aliasing* occurs for

$$\tau = \tau_1 \equiv \pm \frac{1}{2} \left(\Delta_x^{-1} - \Delta_t^{-1} \right) \quad (11.4)$$

which is identical to the formula (10.14).

12. Aliasing in linear interpolation. Joseph's method

In this chapter we will only use linear interpolation. In general we are then employing the window function

$$w(t) = \frac{\Delta_x}{\Delta_w} \Lambda\left(\frac{t}{\Delta_w}\right)$$

Assume that we employ the window function $w(t) = \Lambda\left(\frac{t}{\Delta_x}\right)$, i.e. the width $\Delta_w = \Delta_x$ as in Fig.7.2. From

(10.12) we gather that there will be no DC-aliasing in the $\theta = 0^0$ forward projection. The zero points of the sinc^2 -function coincides perfectly with the repeated versions of the DC-component. Unfortunately, this is not true for the direction $\theta = \frac{\pi}{4}$, where the situation is similar to what was demonstrated in Fig.10.1. The DC-aliasing effect is strong enough to completely ruin any attempt to utilize the projections for reconstruction as shown by the simple experiment in Fig. 13.4, top row. A rather strong unwanted frequency component appears in the sinogram at $\theta = \pm 45^0$. We count manually to find 9 periods over 22 pixel distances Δ_x which is a period of $2.4444 \Delta_x$ and a frequency of $0.4091 \Delta_x^{-1}$. Inserting $\Delta_t = \Delta_x$ in formula (10.5) yields the prediction that the period of this checkered band should have a frequency of $(\sqrt{2} - 1) \Delta_x^{-1} = 0.4142 \Delta_x^{-1}$, which seems to be close enough to confirm the DC-aliasing theory. Furthermore, the relative *amplitude* of this frequency component can be estimated from (10.6) to yield

$$2 W\left(\sqrt{2} \Delta_x^{-1}\right) P_{\frac{\pi}{4}}(0) = 2 \text{sinc}^2 \sqrt{2} = 2 \frac{\sin^2(\sqrt{2}\pi)}{2\pi^2} = 0.094,$$

which is not far from the amplitude estimate $\frac{1}{2}(p_1 - p_2) = 0.1035$ we get from (7.5 and (7.6).

Joseph's method [9] computes projections as illustrated in Fig.12.1. In the direction interval $\frac{\pi}{4} < \theta \leq \frac{3\pi}{4}$, for each vertical line of grid points crossed by the projection ray, the two nearest grid points (x, y) and $(x, y + \Delta_x)$ contribute to the projection with the linearly interpolated value

$$\sin^{-1} \theta \left[(1-a) f_s(x, y) + a f_s(x, y + \Delta_x) \right], \quad (12.1)$$

where a is the intercept distance between the projection ray and the point (x, y) . For $-\frac{\pi}{4} \leq \theta < \frac{\pi}{4}$ a similar definition holds so that the interpolated value shall be computed as

$$\cos^{-1} \theta \left[(1-a) f_s(x, y) + a f_s(x + \Delta_x, y) \right] \quad (12.2)$$

Thus, the window functions w_J , corresponding to (12.1) and (12.2) are

$$w_J(t) = \frac{1}{\sin \theta} \Lambda\left(\frac{t}{\Delta_x \sin \theta}\right) \quad \text{and} \quad \frac{1}{\cos \theta} \Lambda\left(\frac{t}{\Delta_x \cos \theta}\right) \quad (12.3)$$

respectively, which are triangle functions, the width of which varies from Δ_x in the direction $\theta = 0$ and $\theta = \frac{\pi}{2}$ to $\frac{1}{\sqrt{2}}\Delta_x$ along the diagonals, while the height at the same time varies from 1 to $(\sqrt{2})^{-1}$. As a consequence, the zero crossings for the Fourier transform

$$W_J(\tau) = \Delta_x \text{sinc}^2(\tau \Delta_x \cos \theta) \quad (12.4)$$

are not found along circles as in Fig.9.1 but along concentric squares as illustrated in Fig 12.2. DC-aliasing will then be perfectly suppressed for all directions. This is most beneficial for the diagonally oriented projection $\theta = \frac{\pi}{4}$ shown in Fig.12.3. The remaining aliasing is due to second order *shoulders* emanating from the first two copies of $P_{\frac{\pi}{4}}(\tau)$. From our experiments, see Fig. 13.4, we have found that although these contributions are much subdued by the sinc^2 -function they are still afflicting the result.

Fig. 12.1 Joseph's method for computing projections in eq.10.2

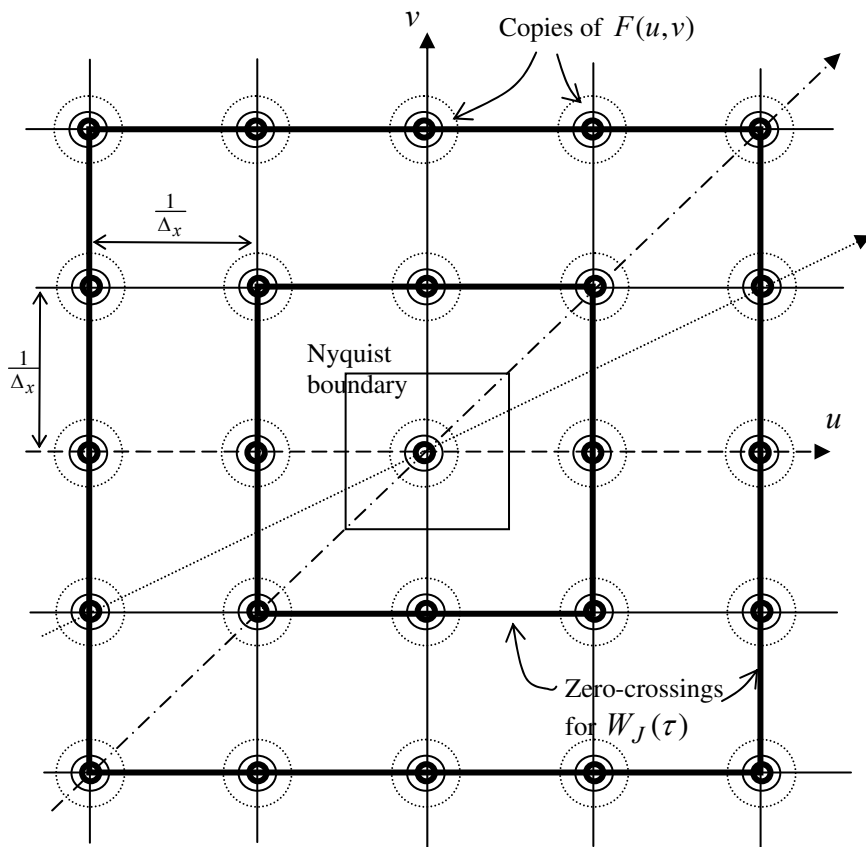
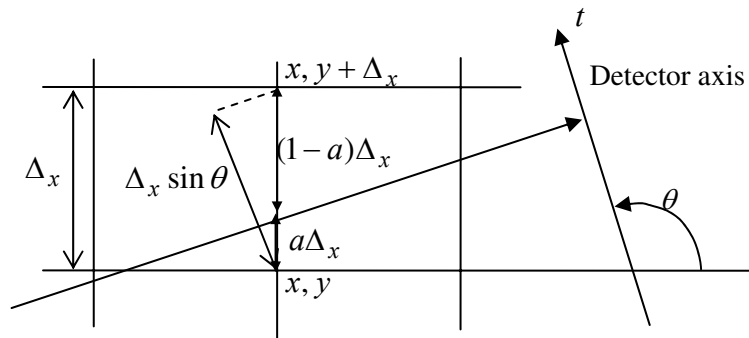


Figure 12.2. Joseph's method

Compare with Fig.9.1. $F_s(u, v)$ of $f_s(x, y)$ is overlaid with the projections $P_0(\tau)$, $P_{\arctan \frac{1}{2}}(\tau)$, and $P_{\frac{\pi}{4}}(\tau)$. Zero-crossings coincide with the DC-components of repeated $F(u, v)$ copies.

In Fig.12.3 the projection employs a detector sampling interval $\Delta_t = \frac{2}{3}\Delta_x$. To decrease this quantity means that the distance Δ_t^{-1} increases in the Fourier domain, which will diminish the aliasing contributions in the bottom line of Fig.12.3. As an example, the shoulder aliasing in Fig. 12.3 will be entirely withdrawn from the Nyquist interval if we make $\Delta_t^{-1} \geq \Delta_x^{-1}\left(\sqrt{2} + \frac{1}{2}\right) \approx 1.92\Delta_x^{-1}$.

Accepting this remedy, remaining aliasing stems from the second and higher order lobes of the sinc²-function. The sum of these is of the same order as the second one that we just got rid of which means that the aliasing contributions are *halved*, which is -6 DB in power reduction. Another doubling of Δ_t^{-1} brings about another halving of the aliasing etc. Undoubtedly, sufficient aliasing reduction is possible to obtain in this manner.

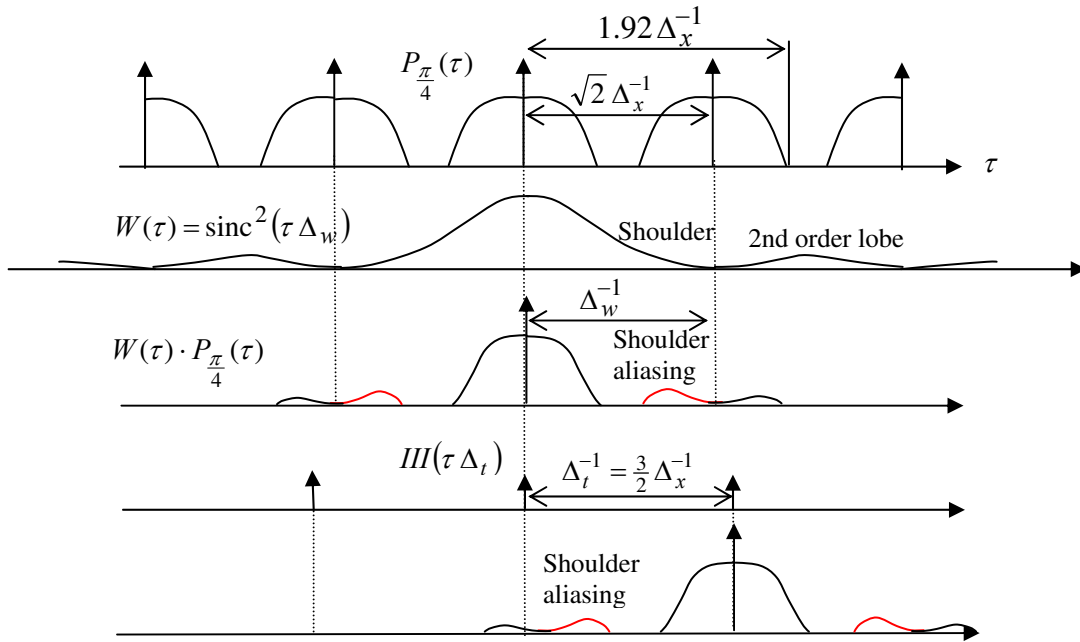


Fig. 12.3. Projection with Joseph's method and $\Delta_t = \frac{2}{3}\Delta_x$, $\theta = \frac{\pi}{4}$

Let us make a rough estimate of these aliasing contributions in more absolute terms. The second side lobe of a sinc²-function has a peak amplitude of $\left(\frac{2}{3\pi}\right)^2 \approx 0.04503$. This peak appears in “no mans land”, i.e. between the Nyquist lobes of the signal in Fig.12.3, which is to say that the dominating DC- and lower frequency components are safely suppressed. Let us do some guess-work. Assume that the maximum effective aliasing amplitude for remaining aliased frequencies is about 16 times smaller than $\left(\frac{2}{3\pi}\right)^2$, say, 0.0032 or -50 DB . To bring this amplitude down to 0.0002 or -74 DB , which we aim to do in the following section, would require no less than a 16-tupling of the detector density $\Delta_t^{-1} = \Delta_x^{-1}$ which would raise the computation cost with a factor of 16 in the present 2D-case, and 64 in the 3D-case.

It maybe noted that the scheme of Fig.1.1 requires that projection data \mathbf{p}_i produced from the digitized result must conform to the given input format for \mathbf{p} . If we increase the detector density of \mathbf{p}_i the projections \mathbf{p} must follow pace. However, and fortunately, since an increase in Δ_t^{-1} is equivalent to do zero-padding of \mathbf{p} before Fourier transformation, the subtraction $\mathbf{p} - \mathbf{p}_i$ in the Fourier domain becomes almost effortless.

An interesting **extension and improvement** of Joseph's method was employed by Nuyts et al in [1]. According to [3], in this implementation both the interpolation function width Δ_w and the detector sampling distance Δ_t varies as

$$\begin{aligned} \Delta_t = \Delta_w = \Delta_x \cos \theta & \quad \text{for } -\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4} \\ \Delta_t = \Delta_w = \Delta_x \sin \theta & \quad \text{for } \frac{\pi}{4} \leq \theta \leq \frac{3\pi}{4} \end{aligned} \quad (12.5)$$

Fig. 12.4, bottom row, shows that for projections in the diagonal directions (as well as for the main directions along the x-and y-axes), not only is the DC-component suppressed, but the remaining aliasing contributions from the two nearest side lobes sum up in a perfect symmetric pattern. Furthermore, these contributions fall in place inside the Nyquist limits exactly in the proper positions without producing any false frequency contributions whatsoever. For other directions θ the benefit of the variable sampling density $\Delta_x \cos \theta$, alternatively $\Delta_x \sin \theta$, is less obvious.

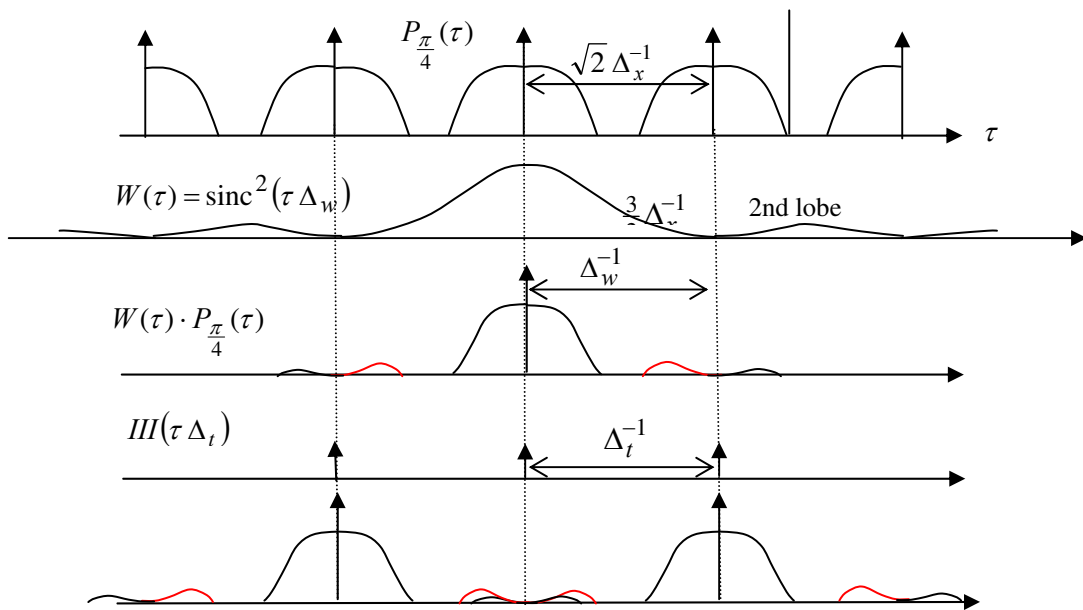


Fig 12.4. Projection with extended Joseph's method $\Delta_t = \Delta_w = \Delta_x \cos \theta$, $\theta = \frac{\pi}{4}$ proposed by Nyuts [3]

It was also pointed out by Nyuts [3] that this method is really designed for speed. The set of parallel projection rays in a projection will intercept the vertical (or horizontal) grid lines at same horizontal (vertical) position. Hence in projection and backprojection the same two interpolation coefficients can be employed repeatedly in $O(N)$ different interpolation steps.

Normally, the detector density is given by the physical design of the tomograph rather than being a free parameter that can be tampered with in the reconstruction algorithm. However, it is not uncommon that the first step in the reconstruction is rebinning which immediately opens the possibility to introduce variable detector density in this step to be used thereafter. Such is the case for most non-exact helical cone beam reconstructions methods, which are the main target for the present enhancement approach.

Finally, it should be mentioned that **the linogram** [27],[11] also carries the same feature as the extended Joseph method of being tuned to the Cartesian sampling pattern. The linogram is a data set obtained from a parallel projection sinogram by resampling to the sampling densities given by (12.5). On top of this, the linogram also requires resampling in the angular direction. In a linogram the angular sampling density should vary from Δ_{θ_0} , for projection angles aligned with the Cartesian image grid, to

$$\Delta_{\theta_0} \cos^2(\theta - \theta_0), \text{ alternatively } \Delta_{\theta_0} \sin^2(\theta - \theta_0) \quad \text{for an arbitrary projection angle } \theta.$$

13. Anti-aliasing using advanced window functions

In this section we will describe and apply alternative rotation-invariant window functions w_t that are zero, or at least for our purposes have “negligible amplitudes”, for frequencies outside the *double* Nyquist interval $|\tau| < \Delta_x^{-1}$. From experiments we have adopted a rule-of-thumb that “negligible amplitudes” means

$$\text{amplitudes} < 0.0002, \quad \text{i.e. power} < -74 \text{ DB}. \quad (13.1)$$

We are going to compute projections from a digital image given in a Cartesian uniform grid with density Δ_x^{-1} . As illustrated in Fig.13.1, if input data are band-limited so that frequencies are zero outside the Nyquist interval $|\tau| < \frac{1}{2}\Delta_x^{-1}$ in any direction a detector density

$$\Delta_t^{-1} \geq \frac{3}{2}\Delta_x^{-1} \quad (13.2)$$

will then be sufficient to eradicate all aliasing in the projection operation. Thus, there is no need to apply Joseph’s technique to combat DC-aliasing in the critical 45°-directions. All is taken care of by the nearly perfectly band-limited and rotation-invariant window function $W(\tau)$.

Note that the projection result $W(\tau) \cdot P(\tau)$ in the middle row of Fig.13.1 contains high frequency components (in red), picked up from the first left and right hand copies of $P(\tau)$ in the first row. These components constitute potential shoulder aliasing. The bottom row shows that the sampled projection carries aliased frequency components within its own Nyquist band $\pm \frac{1}{2}\Delta_t^{-1}$, but also that these can be eliminated, whenever we want to. If not done beforehand, the band limitation from $\pm \frac{1}{2}\Delta_t^{-1}$ to $\pm \frac{1}{2}\Delta_x^{-1}$ will take place during subsequent ramp-filtering.

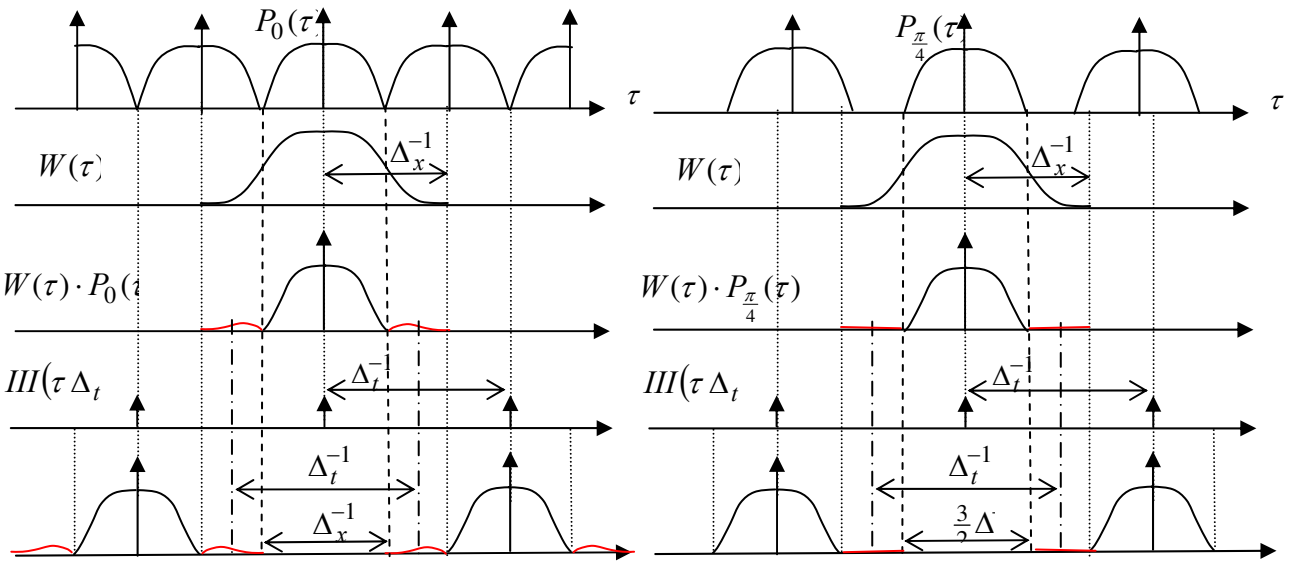


Fig. 13.1. A window function $w(t) \Leftrightarrow W(\tau)$ that suppresses all frequencies above the **double** Nyquist limits $\tau > \pm \Delta_x^{-1}$ eliminates all aliasing inside the single Nyquist limits $\tau \leq \pm \frac{1}{2}\Delta_x^{-1}$, provided that the detector density $\Delta_t^{-1} \geq \frac{3}{2}\Delta_x^{-1}$. The case $\theta = 0$ (to the left) is the worst case. Other directions, for instance $\theta = \frac{\pi}{4}$, have wider margins.

Spline-functions can be seen as generalizations of the rectangle function employed in nearest neighbor interpolation and the triangle function employed in linear interpolation. In the terminology of Unser [25] the rectangle and the triangle functions are B-spline functions of order 0 and 1, respectively. B-spline functions of any order can be generated by successive convolutions of the rectangle function by itself. We are only

interested in functions of odd orders, which make positive functions in the Fourier domain. The triangle function convolved with itself makes a B-spline of order 3, having a total width that is twice the triangle function. The triangle function is a 2-point interpolator, the B-spline of order three is a 4-point interpolator, the B-spline of order five is a 6-point interpolator, etc. In the Fourier domain these B-splines make sinc^2 , sinc^4 , and sinc^6 -functions, respectively. The second side-lobe of these three functions, the one that we want to be negligible in Fig.13.1, amounts to numbers given in Table 13.1.

Only the B-spline of order 5, the six-point interpolator satisfies the criterion (13.1) Compared to linear interpolation the computation cost increases three-fold. Unfortunately, the aliasing suppression comes with a rather strong and unwanted smoothing effect (see Appendix). At the Nyquist limits $\pm \frac{1}{2} \Delta_x^{-1}$, the sinc^6 -function, which is a six-point window function in the signal domain, has melted down to a trickle, namely to $\left(\frac{2}{3\pi}\right)^6 \approx 0.0666$. To compensate for this smoothing Unser [23] advocates preprocessing with approximate inverse filters. To lift the response from 0.0666 to 0.5 at the Nyquist limit, such a filter should magnify 7.5 times for $\tau = \pm \frac{1}{2} \Delta_x^{-1}$ while tapering off to unity for $\tau = 0$ and $\tau = \pm \Delta_x^{-1}$.

Sinc ²	Sinc ⁴	Sinc ⁶
$\left(\frac{2}{3\pi}\right)^2 = 0.0451$	$\left(\frac{2}{3\pi}\right)^4 = 0.00203$	$\left(\frac{2}{3\pi}\right)^6 = 0.000091$
-27 DB	-54 DB	-81 DB

Table 13.1 Amplitudes and power attenuation of the second side-lobe for window functions of type B-spline

From the literature we can find at least two other candidates for the wanted window functions in Fig. 13.1. Rotationally symmetric two-dimensional basis functions called **Blobs** were proposed by Lewitt [10] to yield a window function that is a generalized Kaiser-Bessel function. There are two versions that we think are worthwhile to consider. For simplicity we call them here Blob1 and Blob 2. The corresponding window functions $w_{b1}(t)$ and $w_{b2}(t)$ are shown in Fig.13.2. In the Fourier domain $W_{b1}(\tau)$ tapers off to zero already at the Nyquist limits $\tau = \pm \frac{1}{2} \Delta_x^{-1}$. Thus, shoulder aliasing is suppressed by the mere design of the filter $w_{b1}(t)$ at the cost of strong smoothing and low-pass filtering. Furthermore, from the power spectra (see Appendix) we notice that the suppression of frequencies outside the double Nyquist limits is not sufficient for W_{b1} , according of our rule-of-thumb (13.1). $W_{b2}(\tau)$, on the other hand, tapers off to zero at $\tau = \pm \Delta_x^{-1}$ and has an attenuation for higher frequencies that meets the above requirement (13.1). In principle this filter, which also has a parameter that can change some of its characteristics, would be possible to use as described in Fig.13.1.

The **SinCot-filter** $w_{SCM}(t)$ was first presented in [11] and [12]. A main parameter of this filter is denoted M , which is an even integer that defines the width $-\frac{M}{2} \Delta_x \leq t \leq \frac{M}{2} \Delta_x$ of the filter. We will call such a filter an M -point filter since one interpolation result makes use of M input points. The filter function is

$$w_{SCM}(t) = \frac{1}{M\Delta_x} \left(a + (1-a) \cos \frac{2\pi t}{M\Delta_x} \right) \sin \frac{\pi t}{\Delta_x} \cot \frac{\pi t}{M\Delta_x} \quad \text{for } -\frac{M}{2} \Delta_x \leq t \leq \frac{M}{2} \Delta_x \quad (13.3)$$

$$w_{SCM}(t) = 0 \quad \text{elsewhere}$$

The parameter a changes the shape in both domains. When $M = 4$, to obtain maximum flatness in the inner part of the Nyquist interval of W_{SC4} we set $a = 0.605$. As can be seen from Figures 13.2 and 13.3, outside the double Nyquist limit, for $M = 4$ the suppression of side-lobes is strong ($\leq -60\text{DB}$). The attenuation increases somewhat, but not so significantly by using a six-point filter. i.e. to set $M = 6$ in (13.3).

The formula (13.3) contains the factor $a + (1 - a) \cos \frac{2\pi t}{M\Delta_x}$ which is called the Hamming window. An extended version of the SinCot filter comprises two Hamming windows which then yields

$$w_{SCM2}(t) = \frac{1}{M\Delta_x} \left(a + (1 - a) \cos \frac{2\pi t}{M\Delta_x} \right) \left(b + (1 - b) \cos \frac{2\pi t}{M\Delta_x} \right) \sin \frac{\pi t}{\Delta_x} \cot \frac{\pi t}{M\Delta_x} \quad (13.4)$$

We have found that using $M = 6$, $a = 0.75$, and $b = 0.53$ the corresponding Fourier transform $W_{SCM2}(\tau)$ is less low-pass filtering inside the Nyquist interval $\tau < \left| \frac{1}{2} \Delta_x^{-1} \right|$ while attenuating strongly for $\tau > \left| \Delta_x^{-1} \right|$. A presentation of SinCot- filter design is found in the Appendix.

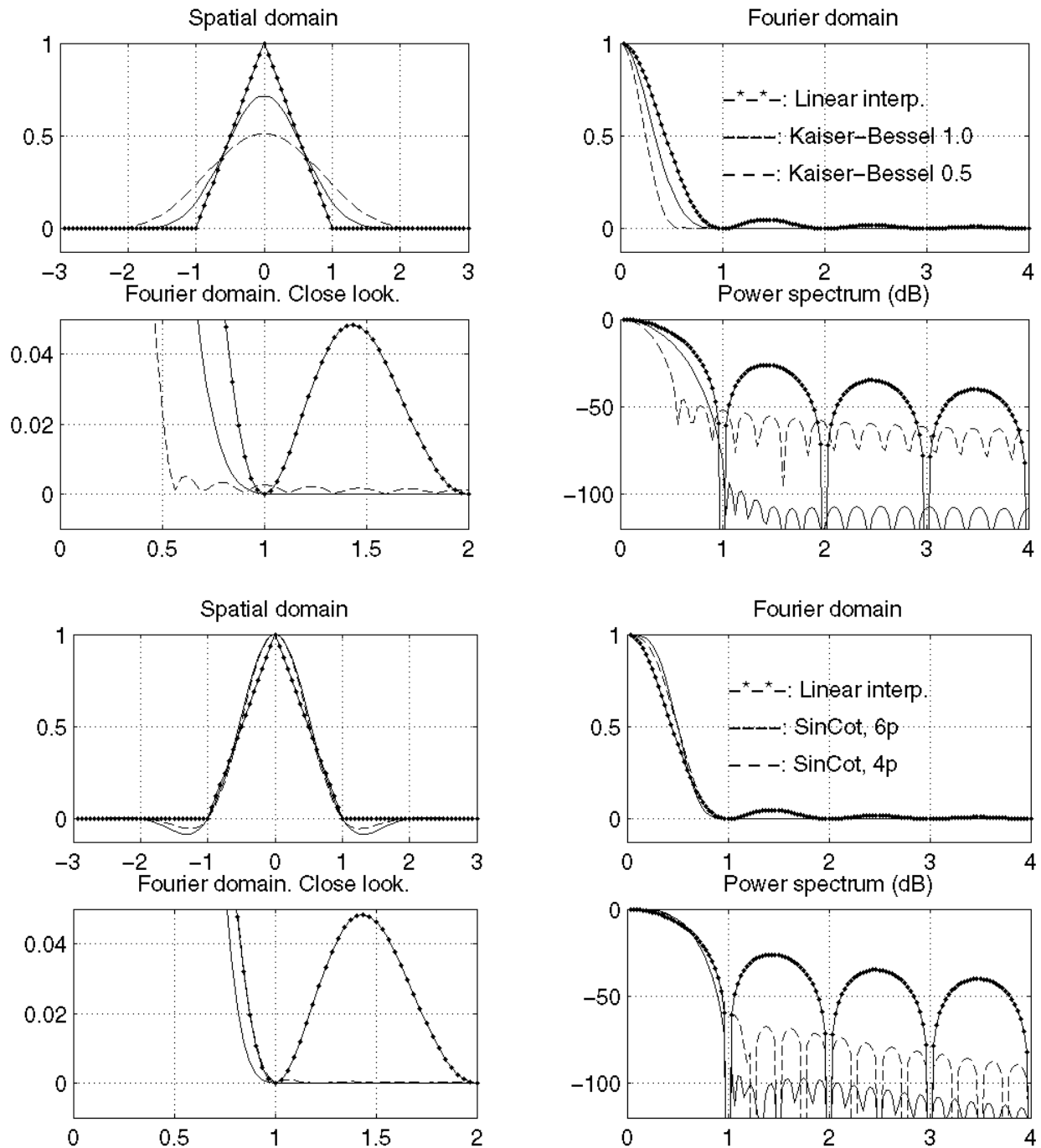


Fig. 13.2 A comparison of five interpolation (window) functions

A definitive advantage of SinCot filters over the Blob filters is *less low-pass filtering* in the Nyquist interval ± 0.5 . A further potential advantage is that the 4-point Sin-Cot filter has zero-crossings in the Fourier domain for $\Delta_w^{-1}, 2\Delta_w^{-1}, 3\Delta_w^{-1}$, etc. just like the sinc^2 -function. Therefore, this filter can possibly be employed with some advantage in the rotation-variant style of Joseph, having a width that matches the repeated copies of $F(u, v)$ in Fig. 9.1 to obtain absolute suppression of DC-components in both zero and 45-degree directions.

Fig. 13.3 demonstrates the importance of minimizing aliasing in *forward projection* by a simple experiment, which is nothing but an attempt to reconstruct an image from projections taken from an already digitized image. This original is not quite but almost identical to the middle image in the bottom row.

The result in the *middle, top row* of Fig. 13.3 is subjected to heavy DC-aliasing of the type explained by Fig.10.1 and already touched upon in Fig. 7.3. The speckled band at 45° in the sinogram (*left, top row*) is a clear visual indication that this effect in forward projections is detrimental and must be avoided.

The result in the *middle, second row* of Fig.13.3 demonstrates that Joseph's technique with parameters as in Fig.12.3 works rather well although not perfectly. The DC-aliasing in the 45° is certainly gone but some other remaining aliasing is still there. The detector density $\Delta_t^{-1} = \frac{3}{2}\Delta_x^{-1}$ is still not sufficient to suppress aliasing from the shoulder of the first sinc^2 -lobe and various parts from second one.

The experiment in the *third row* of Fig. 13.3 is congruent with Fig.13.1 since it uses a 4-point SinCot-filter W_{SC4} in the *projection* step and produces a seemingly spotless result. The *back-projection* in this last experiment was done using the simpler window function used in the top row experiment. Replacing this by W_{SC4} had no visible effects in this specific case.

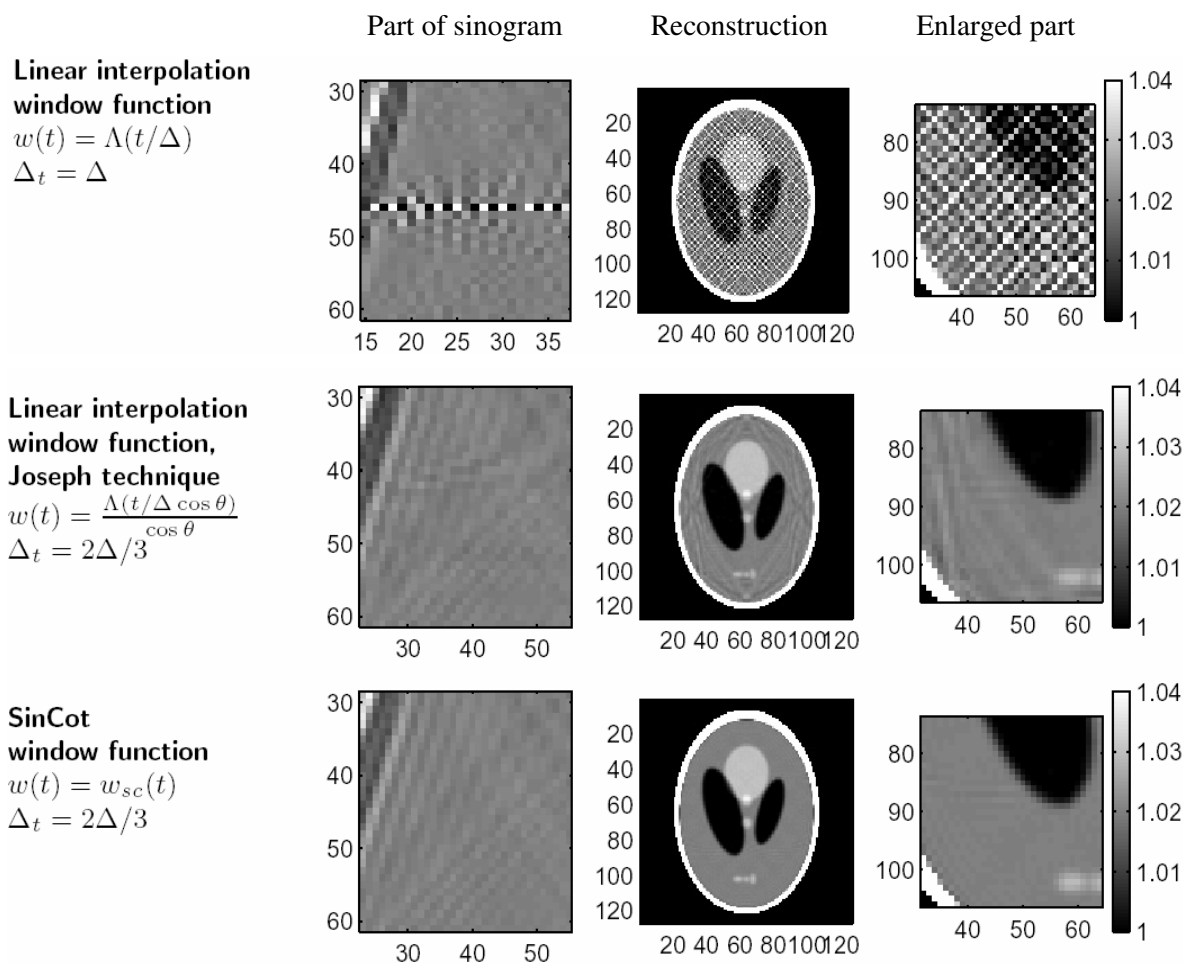


Fig.13.3 Reconstruction via projections of a digital image using different window functions $w(t)$ in forward projections but the same $\Lambda(t/\Delta_t)$ in back-projection. The quantity Δ_x is denoted Δ in this figure.

14. Proposals for alias-free and stable iterative reconstruction

Considering the theoretical discussions and the experimental verifications of aliasing problems in projection and back-projection we propose the following solutions.

Alternative 1. In the *projection* employ Joseph's technique as in Fig.12.3, but replace the linear interpolation window function with a SinCot-function. To avoid shoulder aliasing for all directions, *increase the detector sampling density* to

$$\Delta_t^{-1} = \left(\sqrt{2} + \frac{1}{2}\right)\Delta_x^{-1} = 1,92\Delta_x^{-1}$$

In the *back-projection* (see previous Figures 9.2 and 11.1 and Fig. 14.1 below), the repeated copies of the filtered projections in the top row will be moved apart leaving an empty space of $0,92\Delta_x^{-1}$ in between as shown in Fig. 14.1. This space nullifies most but not all shoulder aliasing. However, thanks to the SinCot-filter the *projection* (bottom half of Fig.14.1) is now completely alias-free just as in Fig.13.1. Furthermore, since the SinCot filter has less low-pass character than the linear interpolator, all frequency components in incoming data are quite faithfully reconstructed.

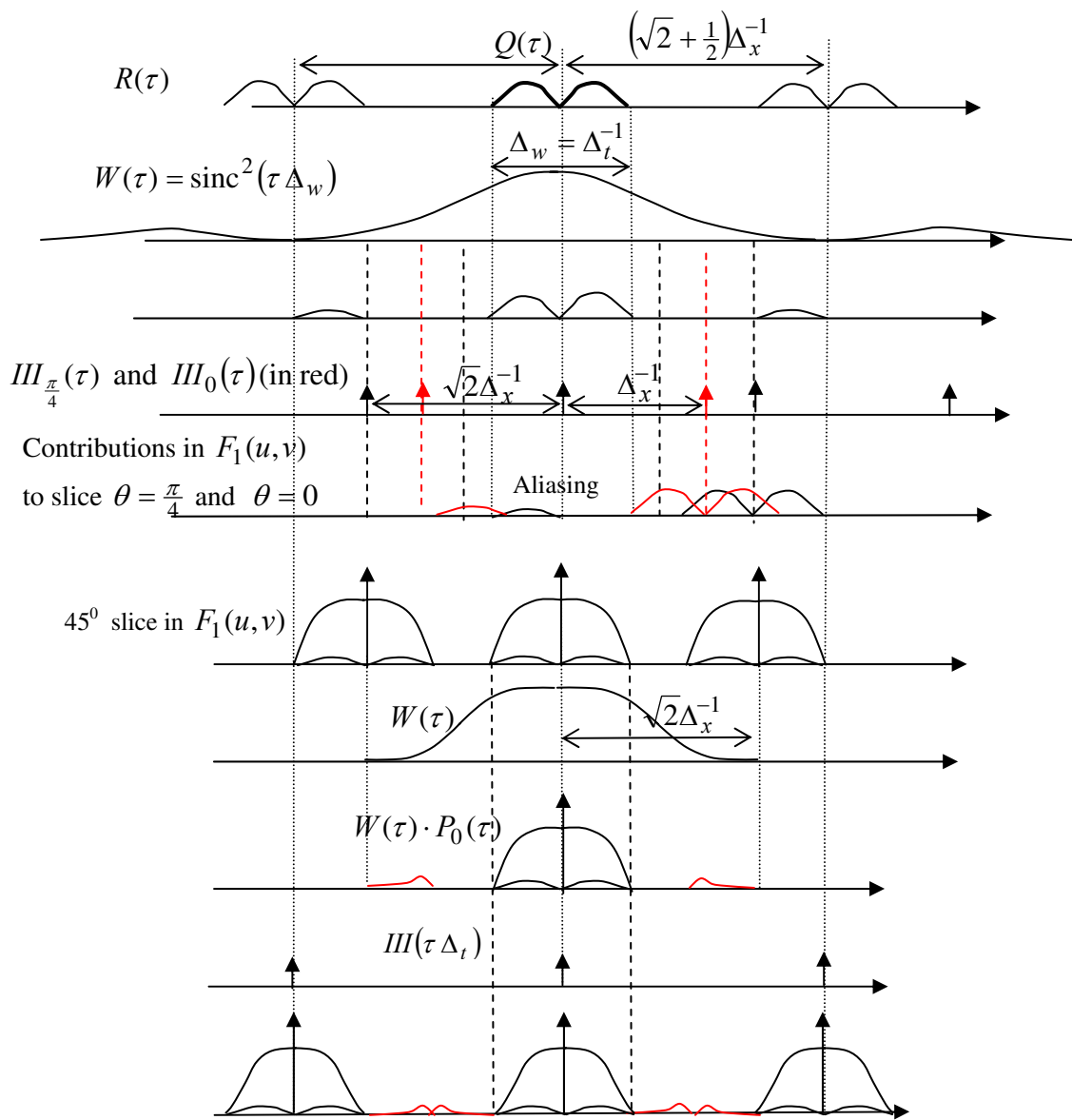


Fig. 14.1. Back-projection with linear interpolation and $\Delta_w^{-1} = \Delta_t^{-1}$, followed by projection using SinCot filter and Joseph's principle. Detector sampling density: $\Delta_t^{-1} = \Delta_x^{-1}\left(\sqrt{2} + \frac{1}{2}\right) \approx 1.914$

Alternative 2. Employ detector *sampling density* $\Delta_x^{-1} = \frac{3}{2}\Delta_x^{-1}$ and the same *SinCot function* having the *fixed width* Δ_x^{-1} in the Fourier domain for both projection and back-projection. The back-projection and the forward projection operations in Fig.14.2 will not produce any aliasing what so ever since the extra free space between the input copies eliminates potential shoulder contributions and the sampling density is large enough to avoid overlaps.

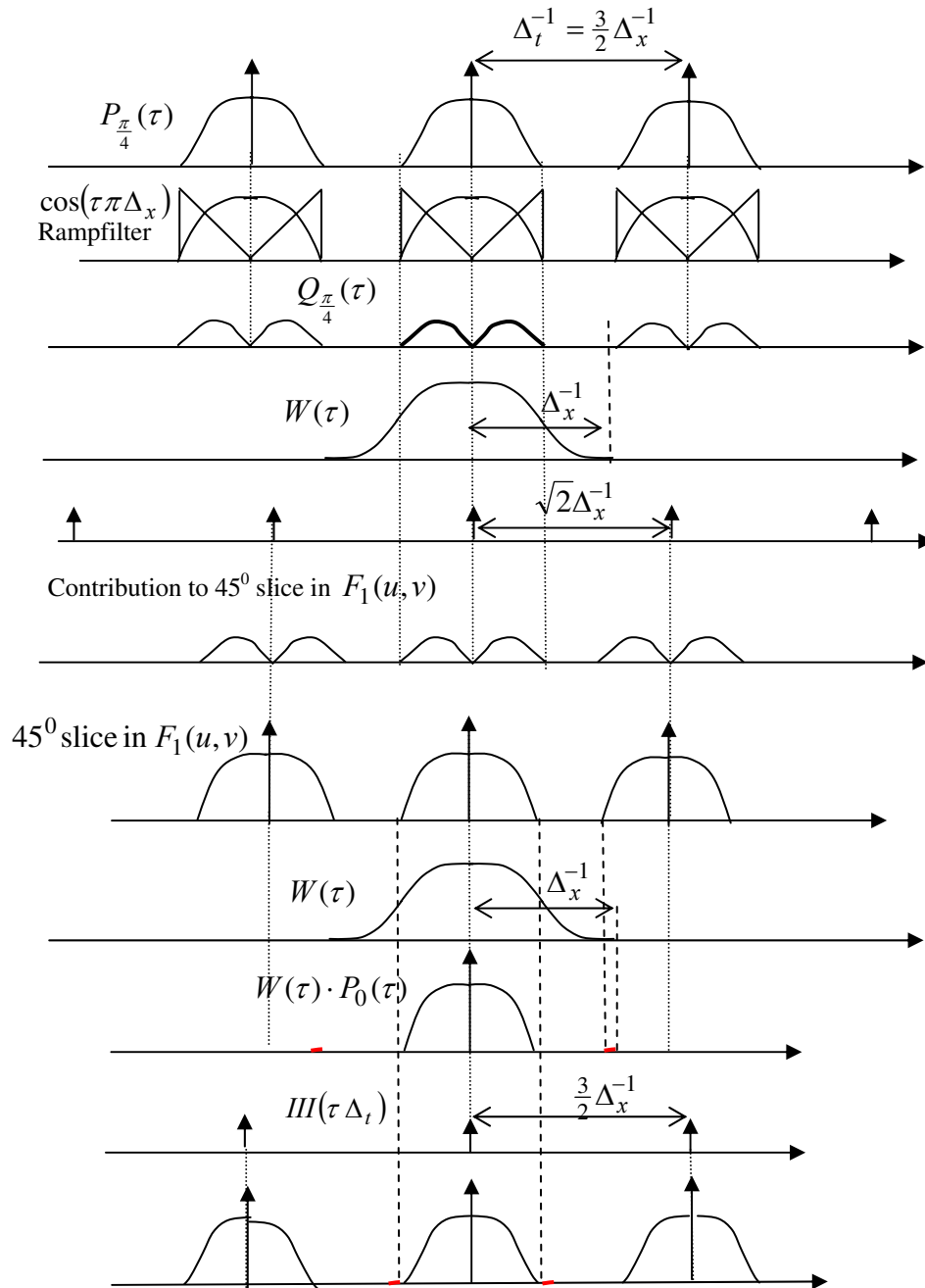


Fig. 14.2. Aliasing-free reconstruction of the \mathbf{f}_1 -image from projection data \mathbf{b} with ramp-filtering and back-projection followed by alias-free projection using Sin-Cot filter with $\Delta_t^{-1} = \frac{3}{2}\Delta_x^{-1}$

15. Experiments

15.1 Convergence experiments with truncated rampfilter.

November 2003 (Maria Magnusson Seger)

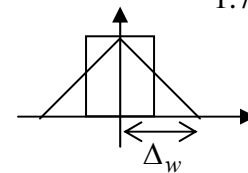
The following statement is a corner-stone of the basic scheme in Fig.1.1. Even if the reconstruction operator Q differs from P^{-1} , the final result will converge to $P^{-1}\mathbf{p}_0$ if we have modeled the forward projection operator P correctly. To find out if the basic scheme works in a 2D-reconstruction case we could then deliberately make the FBP reconstruction defect. In the present experiments we have accomplished this by truncating the ramp-filter convolution kernel so that its support is no longer twice the image size but various percentages thereof. The following parameters are common to all three experiments in this series of experiments.

Image size 128x128, **Image sampling density** Δ_x in both x-and y

Number of views $N_v = 180 \approx \frac{\pi}{2} 128$ **Detector sample distance** $\Delta_t = \frac{\Delta_x}{1.7}$

Phantoms Mathematical Shepp-Logan

Projections somewhat smoothed and band-limited to $(2\Delta_x)^{-1}$

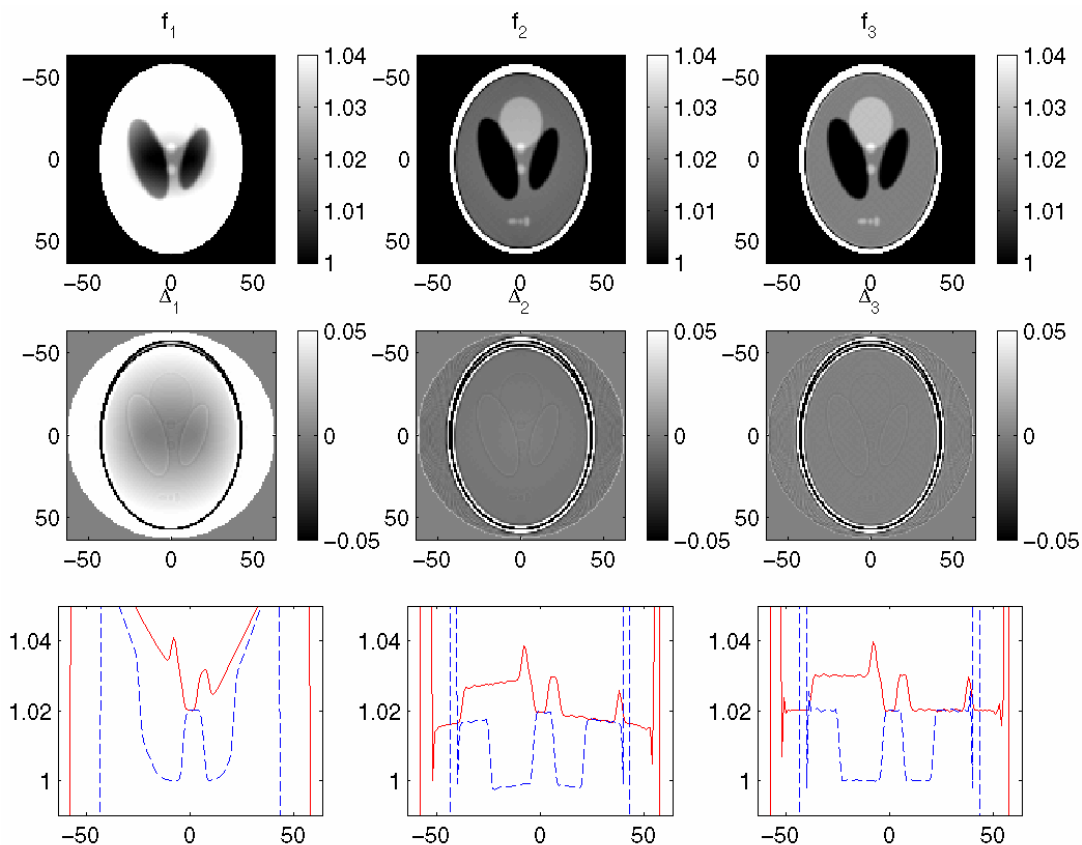


Reconstruction parameters

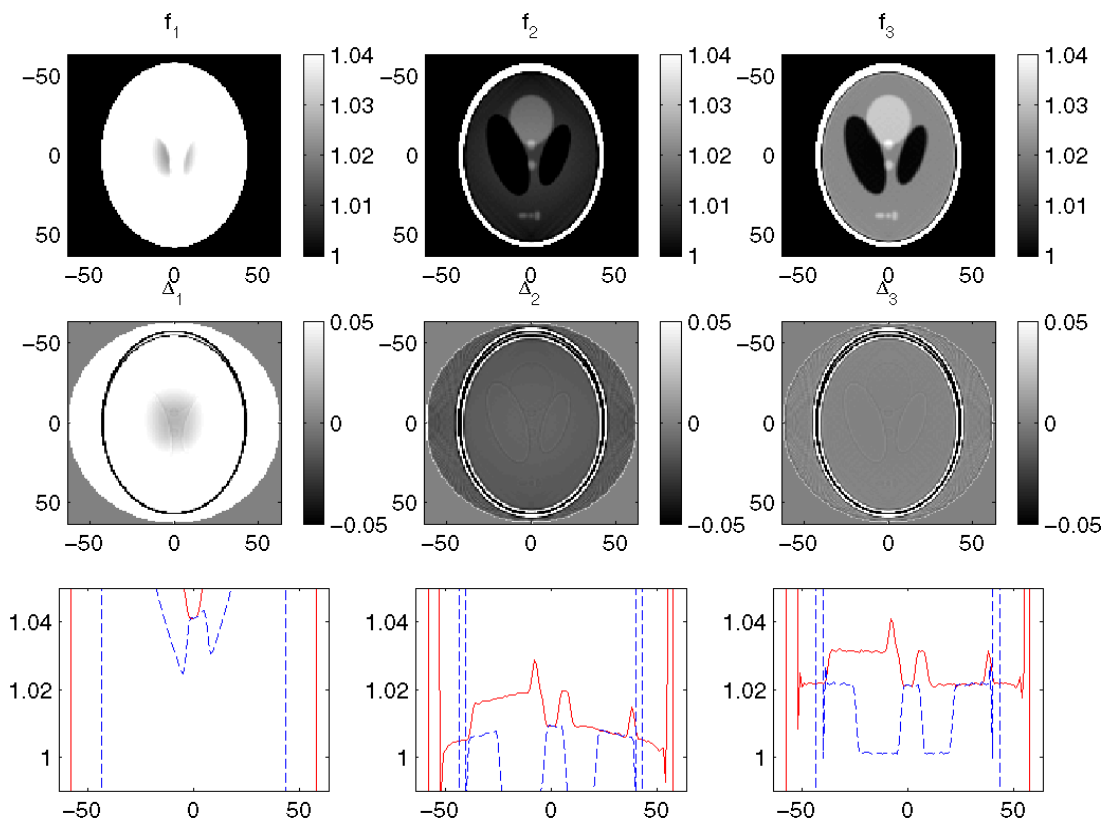
Back-projection: Linear interp. $\Delta_w = \Delta_t$ Forward projection: 4-point SinCot $\Delta_w = \Delta_x \cos \theta$ (Joseph)

Ramp-filter : Truncated in signal domain, cos-weighted in Fourier domain

Experiment 1 Gain $\alpha = 1$ Rampfilter truncated to 0.47 of nominal width



Experiment 2 Gain $\alpha = 1$ Rampfilter truncated to 0.37 of nominal width

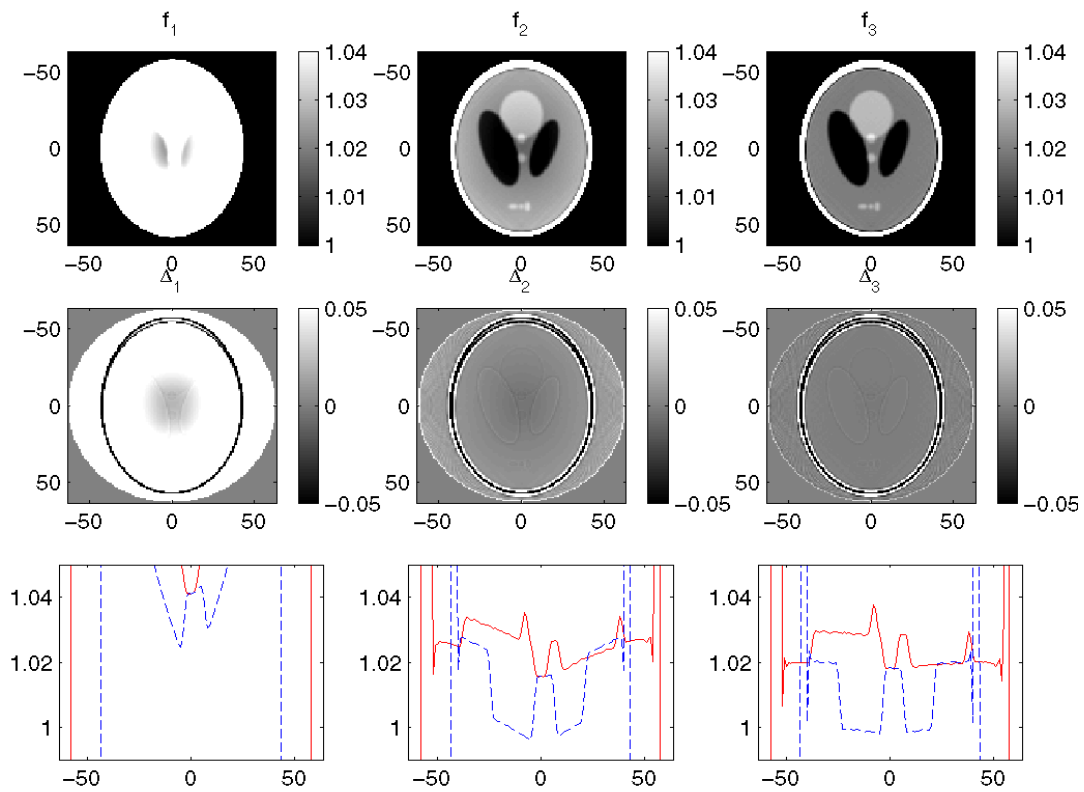


Experiment 3 Rampfilter truncated to 0.37 of nominal width

Gain $\alpha = 1$

Gain $\alpha = 0.8$

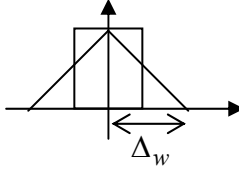
Gain 0.8



15.2 Aliasing artefacts for various interpolation techniques

February 2004 (Johan Sunnegårdh) and May 2004 (Maria Magnusson Seger)

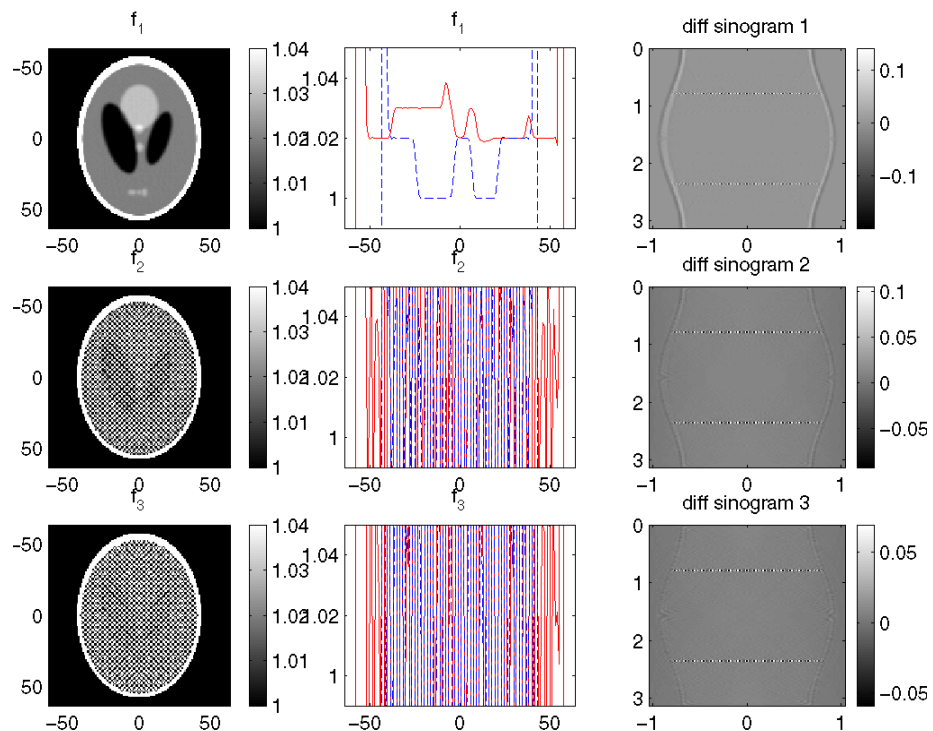
Most of the basic parameters for this series are identical to the ones in 15.1. They are repeated here for the sake of completeness.

Image size	128x128,	Image sampling density	Δ_x in both x-and y
Number of views	$N_v = 180 \approx \frac{\pi}{2} 128$	Detector sample distance	$\Delta_t = \frac{\Delta_x}{1.5}$
Phantoms	Mathematical Shepp-Logan		
Projections somewhat smoothed and band-limited to	$(2\Delta_x)^{-1}$		
Reconstruction parameters			
Back-projection interpolation	Linear, 4-point or 6-point SinCot. Width $\Delta_w = \Delta_x$		
Forward projection interpolation	6-point SinCot filter $\Delta_w = \Delta_x$		
Ramp-filter	Truncated in signal domain, cos-weighted in Fourier domain		

Experiment 1.

$$\frac{\Delta_x}{\Delta_t} = 1$$

Linear interp.



No Joseph

Catastrophy
due to DC-aliasing.

Note, however, that the
sinogram differences are
decreasing

Experiment 2.

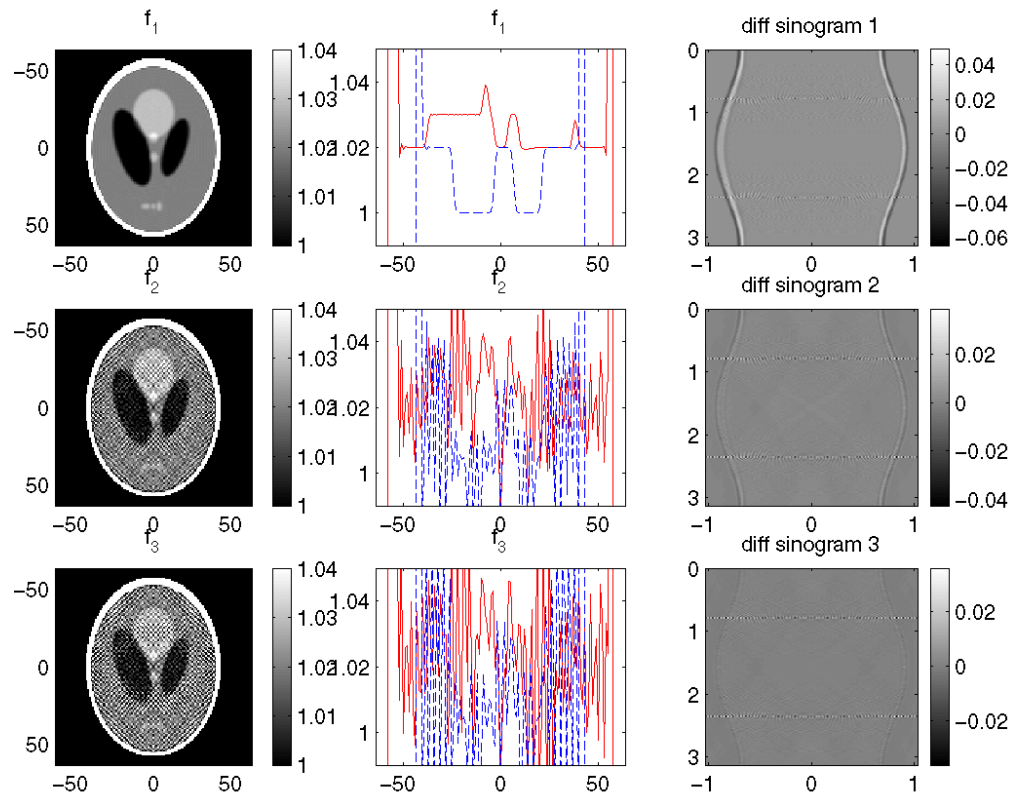
$$\frac{\Delta_x}{\Delta_t} = 1.92$$

Linear interp.

No Joseph

Shoulder aliasing is diminished but DC-aliasing is still very serious.

Experiments 1 and 2 proves that linear interpolation without Joseph's or some related technique will never work in iterative reconstruction



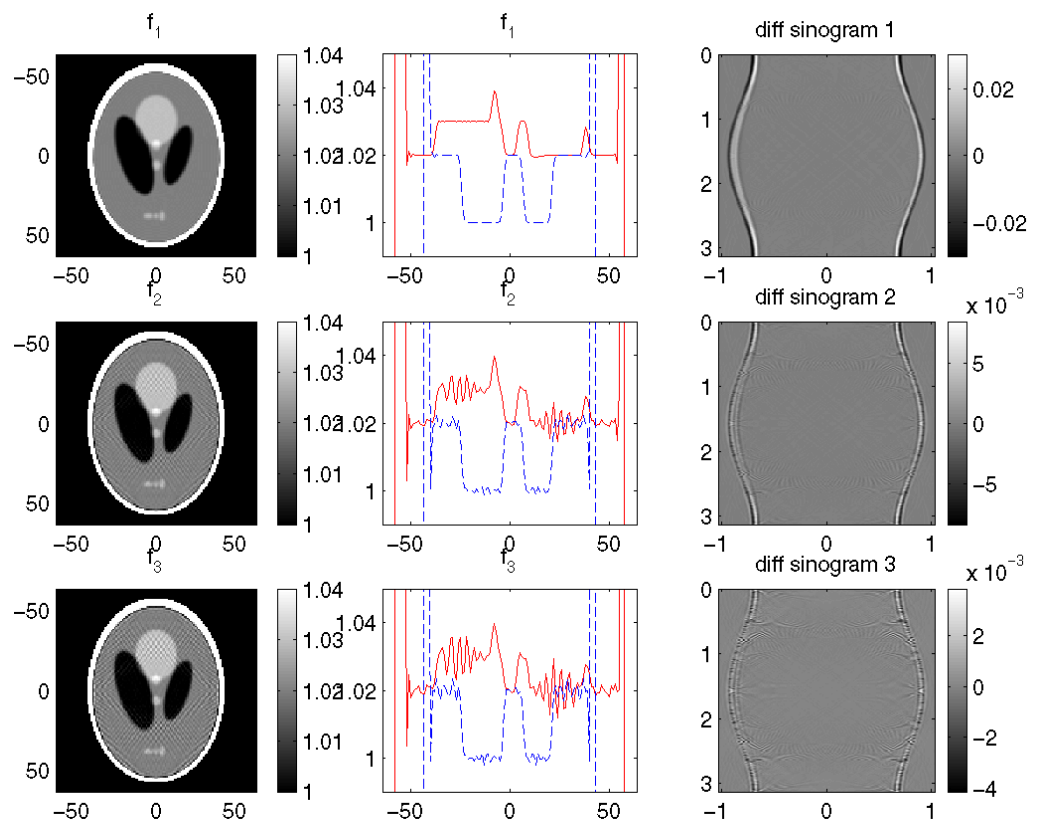
Experiment 3.

$$\frac{\Delta_x}{\Delta_t} = 1.92$$

Linear interp.

Joseph: $\Delta_w = \Delta_x \cos \theta$

DC-aliasing is eliminated. Some aliasing still remains



Experiment 4

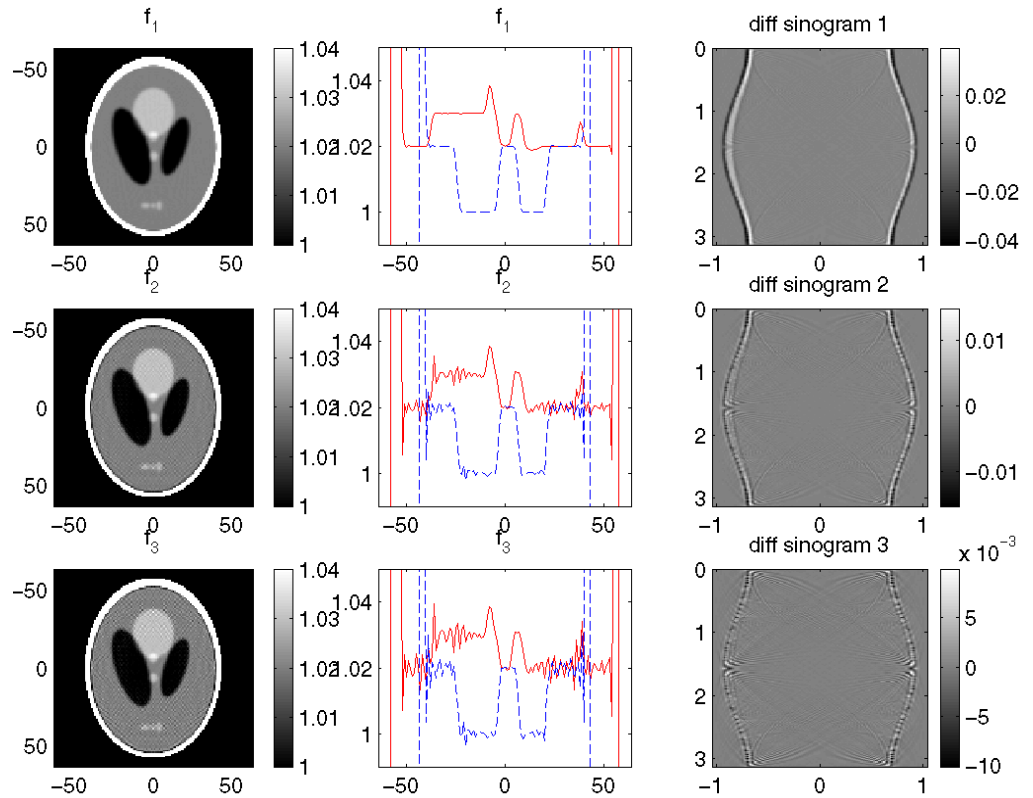
$$\frac{\Delta_x}{\Delta_t} = 1$$

Linear interp.

Joseph: $\Delta_w = \Delta_x \cos \theta$

Surprise: This case seems less prone to aliasing than Exp. 3. Additional shoulder aliasing should degrade the result.

This is the common version of Joseph's method, employed by Turbell in TAKE and Bruno de Mann in his talk in St Malo. Works well for high contrast objects but is not satisfactory in our case.



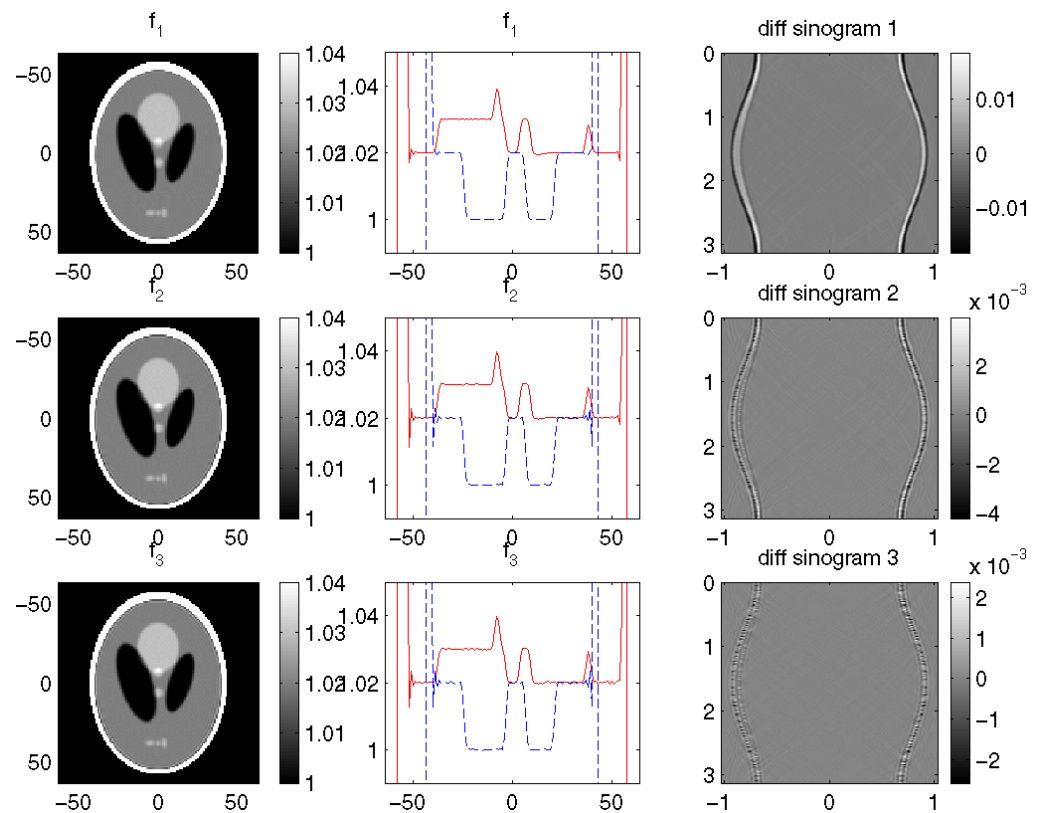
Experiment 5.

$$\frac{\Delta_x}{\Delta_t} = 1.92$$

4p-SinCot

No Joseph

Probably the best image quality using 4p-SinCot



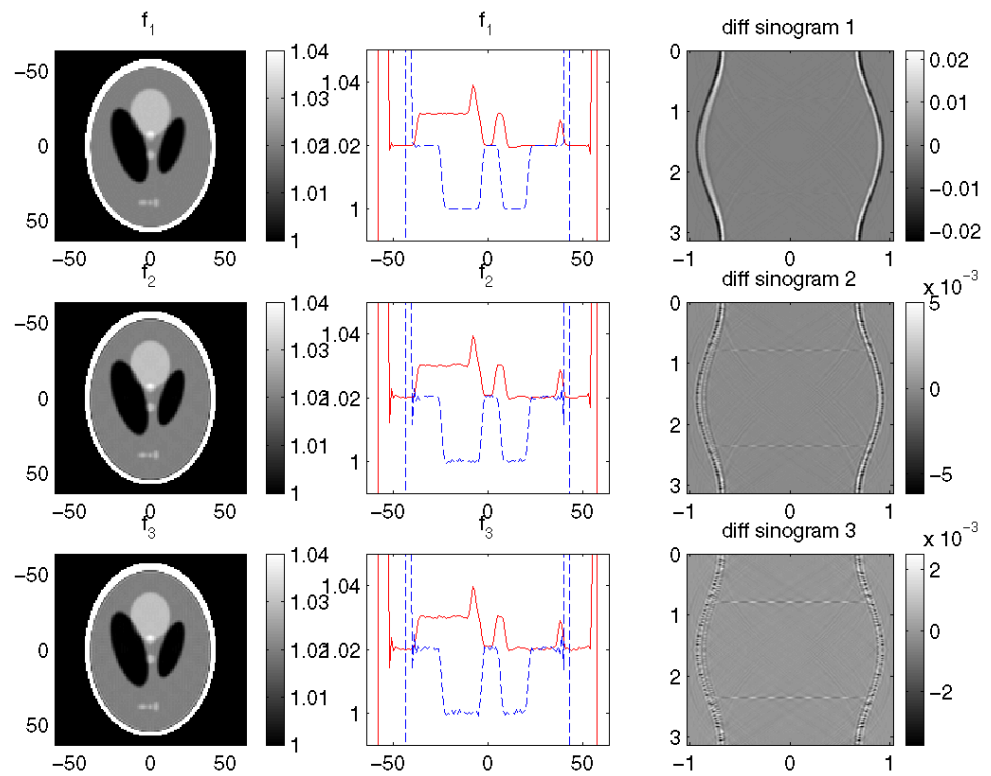
Experiment 6

$$\frac{\Delta_x}{\Delta_r} = 1.5$$

4p-SinCot

No Joseph

Image quality is not quite as good as in Exp.1.5. Note that a slight DC-aliasing occurs in 45-degree directions.



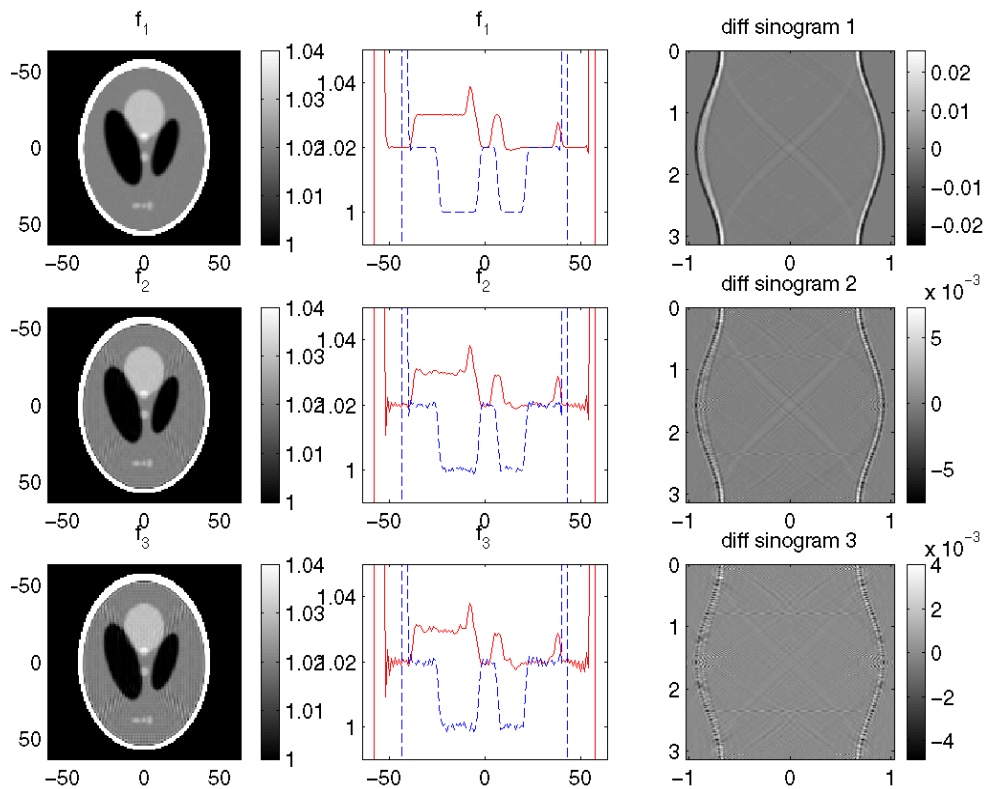
Experiment 7

$$\frac{\Delta_x}{\Delta_r} = 1.3$$

4p-SinCot

No Joseph

Degraded image quality due to shoulder aliasing.



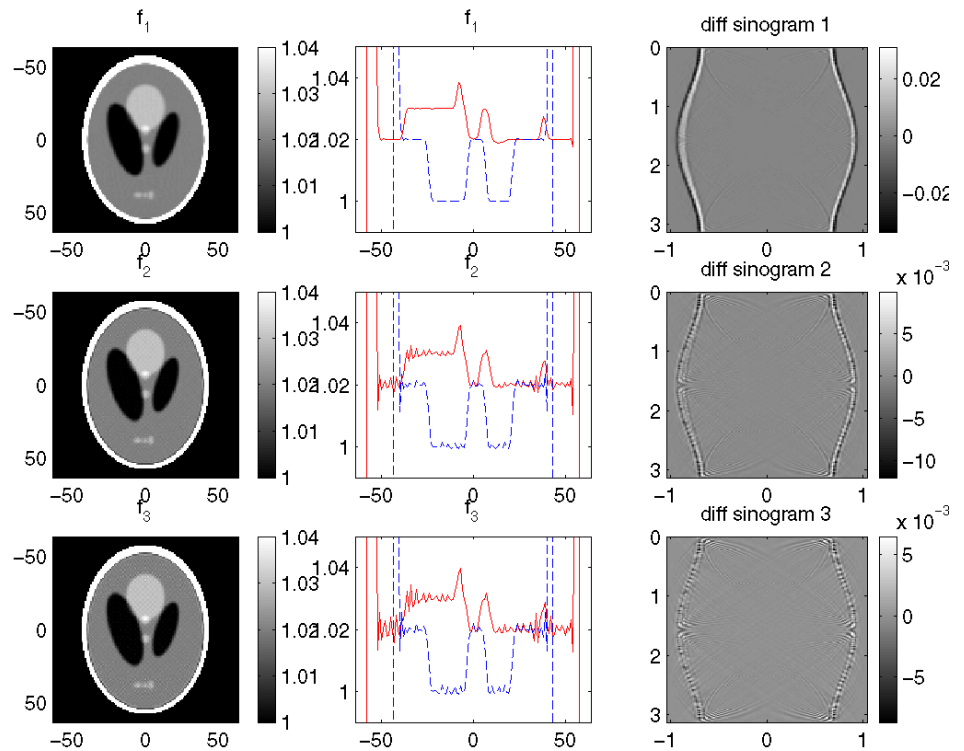
Experiment 8

$$\frac{\Delta_x}{\Delta_t} = 1$$

4p-SinCot

No Joseph

*Degraded image quality.
Full shoulder aliasing*



Experiment 9

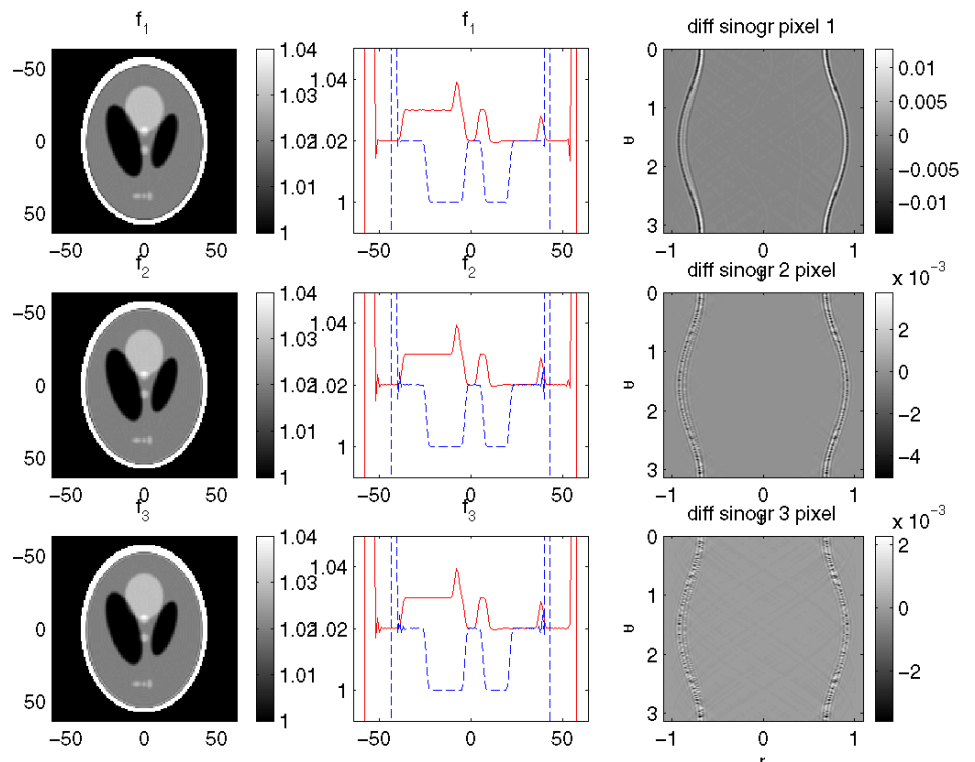
$$\frac{\Delta_x}{\Delta_t} = 1.5$$

6p-SinCot

No Joseph

Compare Exp. 5 and 6.

Conclusion: *In some circumstances, Image quality can be preserved by trading filter size for detector resolution.*



Experiment 10

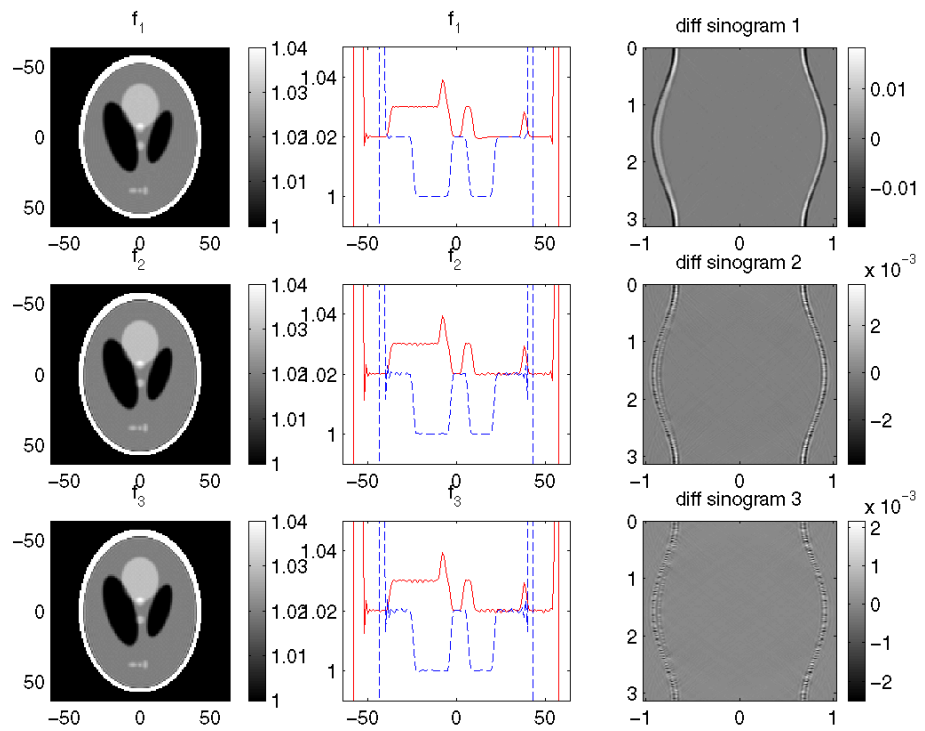
$$\frac{\Delta_x}{\Delta_t} = 1.92$$

4p-Sincot

Joseph: $\Delta_w = \Delta_x \cos \theta$

Joseph with 4p-SinCot should be overkill.

Is image quality really better than in Exp. 5 ?



Experiment 11

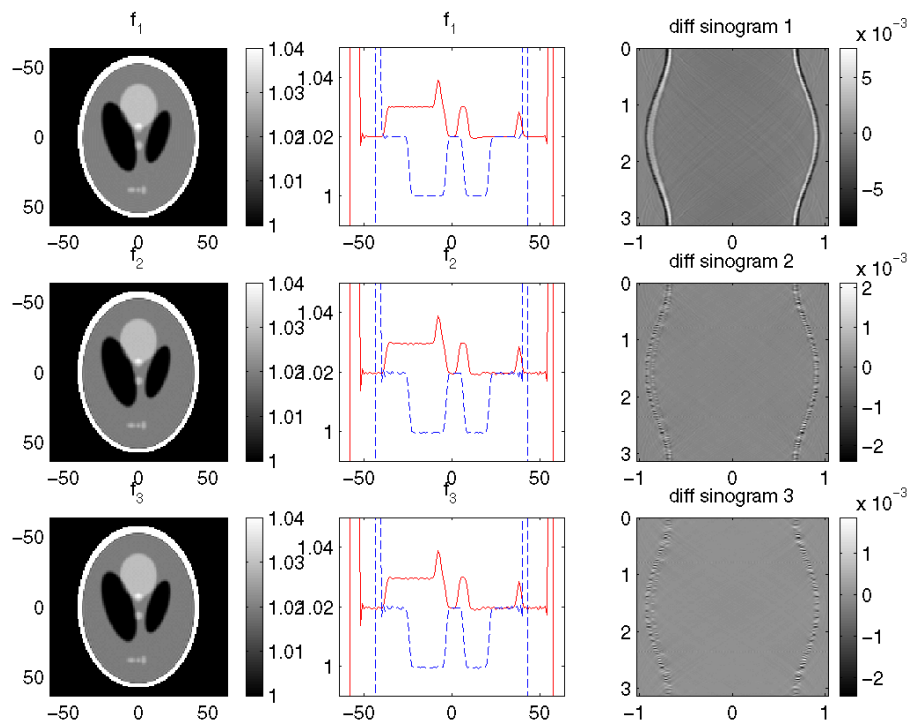
Inverse filtering

$$\frac{\Delta_x}{\Delta_t} = 1.92$$

4p-SinCot interp.
and
(SinCot 4p)⁻¹-filtering of
the projection results

No Joseph

Slightly better result than
in Exp.1.5



14. Discussions and conclusions

Although this report only includes 2D-experiments, we believe that we have verified that the Fig.1.1 iteration scheme works. It might be a way to get exact reconstruction when only approximate reconstruction algorithms are available. Exact and efficient reconstruction algorithms have been published, for instance by Schaller et al [19], by Katsevich [20], and by Zou and Pan [21]. However, just as for the PI-method [13],[18], to fully utilize the given detector, these methods require either a fixed translation speed of the patient, or alternatively, that the relation between the gantry rotation and table feed velocities are kept constant. The last alternative is not technically feasible. Still, we want to arbitrarily slow down the table feed translation to gain higher photon flux and better signal-to-noise ratio. Unfortunately, none of the above-mentioned methods allows for this. Cutting the speed to 60% of the maximum also means that only 60% of the available detector area will be utilized.

In the AMPR method, see Stierstorfer et al [22] and Flohr et al [23], employed by Siemens, and in the “Wedge” method, see Tuy [24], employed by Philips, the dose utilization is 90 % or better within a wide range of translation speeds. Obviously, non-exact reconstruction methods such as these could benefit of the iterative scheme proposed in this report. Iterative methods are as old as CT itself, which is around 35 years. Still, iterative methods of today are mostly employed for other imaging modalities and hardly at all for CT. It may then seem pretentious to nourish a hope that a different fate for the scheme of Fig.1.1. The reasons to be hopeful can be found in the following list of achievements.

1. The proposal of Fig.1.1 has never been tested, at least not publicized before. Our guess is that unpublicized experiments have turned out to be very discouraging, maybe as discouraging as the simultaneous ART experiments found in the well-known textbook by Kak and Slaney [17], pp285-295.
2. The matrix representation QP of the recursive loop is possible to analyze. It lends itself to define the convergence rate in terms of the largest point-spread function error as in (3.16).
3. For the benevolent case, when the matrix QP embeds one single space-invariant point-spread function h_{QP} the eigenvalues of QP are nothing but the frequency components of h_{QP} . Although our goal is to handle the space-variant cases as well, we assume that filtered back-projection is delivering a good approximation of the final result. Therefore, our modeling of the projection-rampfiltering-back-projection loop in terms of frequencies should bear enough similarity to the eigenvalues of QP to analyze potential breaks with the convergence condition (3.4) and finding optimal gain using (3.15) and (3.18).
4. We have indeed managed to construct a consistent Fourier model for the QP -loop, based on a thorough understanding of the relation between basis-functions, window functions, and the sampling processes during projection and back-projection.
5. Aliasing is constant threat to high image quality, a threat that becomes more obvious for each additional iteration step. In [17], p 292, the authors suspect that the problems are due to “inconsistencies in the model used for the forward projection process”. We believe we have come to terms with these “inconsistencies” in both forward and back-projection up to a point where we can make both operations “aliasing-free”.
6. Better understanding of aliasing gives better chances to avoid unnecessary low-pass filtering in the reconstruction process.

Among the shortcomings of this paper, postponed to a second report on the same subject, are the following.

1. More 2D-experiments using different phantoms
2. 2D-experiments with fewer projections
3. 2D-experiments with noisy input projection data
4. Experiments with 3D phantoms and helical cone beam algorithms

Acknowledgements

Many of the experiments that are referred to are conducted by Jonas Sunnegårdh. The authors are grateful to Björn Johansson and Tommy Elfving, Linköping University for valuable contributions, discussions, and literature guidance on the convergence issue of this paper. Financial support from Siemens Medical Solutions is gratefully acknowledged.

References

- [1] J.Nuyts, B.DeMan, P.Dupont, M.Defrise, P.Suetens, and L.Mortelmans, *Iterative reconstruction for helical CT: a simulation study*, Phys. Med. Biol. Vol 43, pp 729-737, 1998
- [2] W.H.Press, S.A.Teukolsky, W.T.Vetterling, and B.P.Flannery, *Numerical Recipes in C*, Second edition, Cambridge University Press, 1992
- [3] J.Nuyts, personal communication, 2004
- [4] T-S Pan, A.E.Yagle, N.H.Clinthorne, and W.L. Rogers, *Acceleration and filtering in the general Landweber iteration using a variable shaping matrix*, IEEE Medical imaging, vol 12, pp 278-286, 1993
- [5] S.Kaczmarz, *Angenaherte auflösung von systemen linearer gleichungen*, Bull. Acad. Pol. Sci. Lett. A **6-8A** (1937) 355-357
- [6] P.E- Danielsson *Metod för bildförbättring i datortomografi*, Swedish patent, C2 520708, filed Dec.3, 2001, approved, Aug. 12, 2003
- [7] P-E. Danielsson, Maria Magnusson Seger, *Aliasing in forward projection of sampled images*, Abstract book, 2003 Meeting on Fully Three-Dimensional Medical Image Reconstruction, St Malo, France
- [8] L.Råde and B.Westergren, *Mathematics Handbook for Science and Engineering*, (BETA), Studentlitteratur, Third edition, 1995
- [9] P.M. Joseph, An improved algorithm for reprojecting rays through pixel images. *IEEE Trans. Med. Im.* **1** (1982) 192-196
- [10] R. Lewitt, Multidimensional digital image representations using generalized Kaiser-Bessel window functions. *J.Opt. Soc. Am. A.*, **7** (1990) 1834 -1846
- [11] M. Magnusson (Seger), Linogram and other direct Fourier methods for tomographic reconstruction, Linköping Studies in Science and Technology, Dissertation No.320, Linköping University (1993)
- [12] M. Magnusson Seger, Three-dimensional reconstruction from cone-beam data using an efficient Fourier technique combined with a special interpolation filter, *Physics in Medicine and Biology* **43** (1998) 951-959
- [13] H. Turbell, Cone-beam reconstruction using filtered backprojection, Linköping Studies in Science and Technology, Dissertation No.672, Linköping University, 2001
- [14] R. L. Siddon, Fast calculation of the exact radiological path length for a three-dimensional CT array. *Medical Physics* **12** (1985), 252-255
- [15] T. Köhler, H. Turbell, and M. Grass, Efficient forward projection through discrete data sets using trilinear interpolation. *IEEE Medical Imaging Conf. Abstracts*, Oct 16-20, (2000), Lyon, France
- [16] K.C.Tam, G. Lauritsch, and K. Sourbelle, Filtering point-spread function in back-projection cone-beam CT and its applications in long object imaging, *Physics in Medicine and Biology* **47** (2002) 2685-2703
- [17] A.C. Kak and M. Slaney, *Principles of computerized tomographic imaging*, IEEE Press, New York (1988)
- [18] P.E.Danielsson, P. Edholm, J. Eriksson, M. Magnusson Seger, H. Turbell, *Technique and arrangement for tomographic imaging*, International application published under the Patent Cooperation Treaty (PCT), WO 98/30980, July 16, 1988
- [19] S. Schaller, F. Noo, F. Sauer, K.C. Tam, G. Lauritsch, and T. Flohr, *Exact Radon rebinning algorithms using local regions-of-interest for helical cone-beam CT*, IEEE Trans. Med. Imaging, **19**, pp 361-375, 2000

- [20] Katsevich, A., *Analysis of an exact inversion algorithm for spiral cone-beam CT*. Phys. Med. Biol., **47** pp 2583-2597, 2002.
- [21] Y. Zou, and X. Pan, Exact image reconstruction on PI-lines in helical cone-beam CT, Dec 16, 2003 (to be published)
- [22] K.Strierhofer, T.Flohr, and H.Bruder, Segmented multiple pale reconstruction: a novel approximate reconstruction scheme for multi-slice spiral CT, Phys. Med. Biol. **47**, pp 2571-2581, 2002
- [23] T. Flohr, K. Strierhofer, H. Bruder, J. Simon, A. Polacin, and S. Schaller, *Image reconstruction and image quality evaluation for a 16-slice CT scanner*, Medical Physics, **30**, pp 823- 831, 2003
- [24] H. Tuy, *Image reconstruction from helical partial cone-beam scanners using wedge beam*, U.S. Patent 6104775, filed 8/15/2000
- [25] M. Unser, *Splines. A perfect fit for signal and image processing*, IEEE Signal Processing Magazine, pp 22-38, Nov. 1999
- [26] T. Nielsen, T. Köhler, R. Proksa, U. Von Stevendaal, *Spatial resolution of iterative cone-beam CT reconstruction*, in 2003 IEEE Nuclear Science Symposium Conference Record", (S. Metzler, ed.)
- [27] P.R.Edholm, G.T.Herman, *Linograms in image reconstruction from projections*, IEEE Trans. on Med. Im. **6**, pp 301-307, 1987