A solution to the long-object problem in helical cone-beam tomography

2000 Phys. Med. Biol. 45 623

(http://iopscience.iop.org/0031-9155/45/3/305)

Phys. Med. Biol. **45** (2000) 623–643. Printed in the UK PII: S0031-9155(00)09045-X

A solution to the long-object problem in helical cone-beam tomography

Michel Defrise†‡, Frédéric Noo§ and Hiroyuki Kudo‖

† Division of Nuclear Medicine, AZ-VUB, Free University of Brussels, Belgium
§ Montefiore Institute of Electricity, University of Liège, Belgium
‖ Institute of Information Sciences and Electronics, University of Tsukuba, Japan

E-mail: mdefrise@minf.vub.ac.be

Received 27 October 1999

Abstract. This paper presents a new algorithm for the long-object problem in helical cone-beam (CB) computerized tomography (CT). This problem consists in reconstructing a region-of-interest (ROI) bounded by two given transaxial slices, using axially truncated CB projections corresponding to a helix segment long enough to cover the ROI, but not long enough to cover the whole axial extent of the object. The new algorithm is based on a previously published method, referred to as CB-FBP (Kudo *et al* 1998 *Phys. Med. Biol.* **43** 2885–909), which is suitable for quasi-exact reconstruction when the helix extends well beyond the support of the object. We first show that the CB-FBP algorithm simplifies dramatically, and furthermore constitutes a solution to the long-object problem, when the object under study has line integrals which vanish along all *PI-lines*. (A PI-line is a line which connects two points of the helix separated by less than one pitch.) Exploiting a geometric property of the helix, we then show how the image can be expressed as the sum of two images, where the first image can be reconstructed from the measured CB projections by a simple backprojection procedure, and the second image has zero PI-line integrals and hence can be reconstructed using the simplified CB-FBP algorithm. The resulting method is a quasi-exact solution to the long-object problem, called the *ZB method*. We present its implementation and illustrate its performance using simulated CB data of the 3D Shepp phantom and of a more challenging head-like phantom.

1. Introduction

The recent introduction of multi-row CT scanners generates numerous questions concerning three-dimensional (3D) image reconstruction in helical cone-beam (CB) tomography. In the current multi-row scanners, the axial aperture of the cone-beam is small because the detectors have only four rows. Therefore, approximate 2D filtered-backprojection (FBP) methods similar to those used in spiral CT are appropriate for reconstruction. But the ongoing technological developments of area detectors will soon lead to large cone-beam apertures for which these approximate 2D algorithms are no longer suitable. For large CB apertures, efficient, stable and highly accurate algorithms are needed. Designing such algorithms amounts to solving what people call the *long-object problem* in the literature.

By definition, the long-object problem is the problem of reconstructing a region of interest (ROI) bounded by two given transaxial slices using only axially truncated CB projections measured for a finite helical path. An optimal solution to this problem should be such that the

‡ Address for correspondence: Division of Nuclear Medicine, University Hospital AZ-VUB, Laarbeeklaan 101, B-1090 Brussels, Belgium.

finite extent of the helical path and the CB collimation (resulting in data truncation) required for reconstruction limit at best radiation exposure.

To solve the long-object problem, a first level of improvement with respect to the 2D FBP algorithms was obtained by backprojecting the data in 3D, along the actual measurement rays. The prototype of this approach is the algorithm of Wang *et al* (1993). This algorithm and other similar ones (Kudo and Saito 1992, Yan and Leahy 1992) can all be viewed as an adaptation to the helical path of the FBP algorithm, called FDK, designed by Feldkamp, Davis and Kress (1984) for a circular vertex path. These methods are derived empirically and generally produce good but approximate reconstructions.

Further improvement of the accuracy of the reconstruction requires *exact* algorithms that discretize exact analytical inversion formula for the long-object problem. This problem is complex, and it is only recently that two solutions have been described and implemented (Schaller *et al* 1999, Kudo *et al* 1999b). This paper presents a third algorithm for the long-object problem, which is based like the two previous ones on the generalization of Grangeat's formula to the case of truncated CB projections. The new algorithm, called the *zero-boundary* (ZB) algorithm, for reasons that will become apparent below, involves an approximation, which is identical to that in Kudo *et al* (1998) and in Kudo *et al* (1999b), and was shown to be highly accurate. In this sense, we speak of a *quasi-exact* algorithm.

The derivation of the ZB algorithm relies heavily on the FBP algorithm proposed in Kudo *et al* (1998) for the full reconstruction of a short object from truncated helical CB projections. Just like this algorithm, the ZB method requires the data to be measured in the region of the detector bounded by the CB projection of the upper and lower turns of the helix. This region, which we call region B, was introduced by Tam *et al* (1998), Danielsson *et al* (1997) and Kudo *et al* (1998), and defines the minimum data set allowing exact reconstruction.

The algorithm will be derived in two steps in section 3. First, we will show that the FBP algorithm in Kudo *et al* (1998) becomes simple for objects which have CB projections vanishing along the boundaries of B. More explicitly, we will show that this algorithm reduces in that case to the FDK algorithm of Yan and Leahy (1992), provided the data are first restricted to B. A key benefit of this approach is that the local character of the ramp filter in the FDK algorithm allows us to handle the long-object problem. Next, we will present a simple and numerically efficient method to modify the projections of an arbitrary object in such a way that they become zero along the boundary of the region B, and therefore can be reconstructed using the FDK method. The construction of the modified projections exploits a remarkable geometric property of the helix (Danielsson *et al* 1997), namely that each point within the cylindrical volume defined by the helix belongs to one and only one *PI-line* connecting two points of the helix separated by less than one pitch. These PI-lines define the boundary of the region B. In section 4, we describe the implementation of the ZB algorithm and present a preliminary study of its performance using simulated data.

2. Scanner description and definitions

Figure 1 illustrates the data acquisition geometry. First note that we use $f(\vec{r})$ with $\vec{r} = (x, y, z)$ to describe the density function to be reconstructed. We assume that this function is smooth and vanishes outside the cylinder

$$\Omega = \{(x, y, z) \in \mathbb{R}^3 \mid \quad x^2 + y^2 \leqslant R_{\mathrm{FOV}}^2\}. \tag{1}$$

In this definition, R_{FOV} is the radius of the field of view of the scanner.

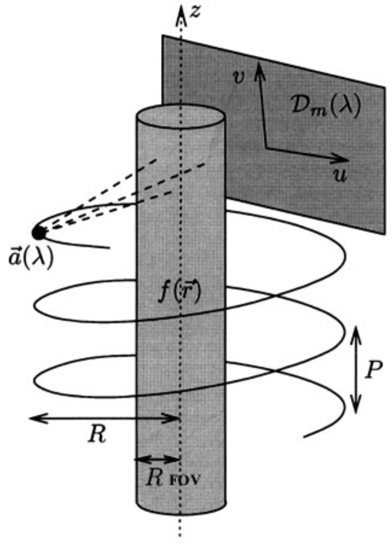

Figure 1. Scanner geometry. The field-of-view is a cylinder of radius $R_{\rm FOV}$, centred on the rotation axis of the scanner. Relative to the object, the source–detector assembly moves along a helix of pitch P and radius R. The u- and v-axes in the detector plane are orthogonal and oriented such that the u-axis is parallel to the vector tangent to the helical path at $\vec{a}(\lambda)$.

To reconstruct $f(\vec{r})$, we have CB projections available for cone vertices on the helical path $\vec{a}(\lambda)$ oriented along the z-axis. The equation of this path is

$$\vec{a}(\lambda) = (R\cos\lambda, R\sin\lambda, h\lambda) \qquad (2)$$

with $\lambda \in [\lambda_{\rm min}, \lambda_{\rm max}]$. In this equation, λ is the angle describing the rotation of the cone vertex, $R > R_{\rm FOV}$ is the radius of the helix and $P = 2\pi h$ is its pitch. When λ increases by 2π, the cone vertex moves a distance P in the z-direction. Note that, in medical CT, the cone vertex represents the position of the x-ray source relative to the patient.

As illustrated in figure 1, we consider that the area detector is flat and moves with the cone vertex. At angular position λ, it is parallel to the z-axis and to the vector $\vec{a}'(\lambda)$ tangent to the helix. The distance from $\vec{a}(\lambda)$ to the detector is D. To locate detector pixels in space, we introduce Cartesian coordinates u and v in the plane $\mathcal{D}(\lambda)$ containing the area detector. These coordinates are measured along unit vectors $\vec{1}_u$ and $\vec{1}_v$ orthogonal in $\mathcal{D}(\lambda)$, with $\vec{1}_u$ parallel to $\vec{a}'(\lambda)$. We have

$$\begin{aligned}\vec{1}_u &= (-\cos\eta\sin\lambda, \cos\eta\cos\lambda, \sin\eta)\\ \vec{1}_v &= (\sin\eta\sin\lambda, -\sin\eta\cos\lambda, \cos\eta)\end{aligned} \qquad (3)$$

with

$$\eta = \arctan(h/R). \qquad (4)$$

The origin $(u, v) = (0, 0)$ is at the orthogonal projection of $\vec{a}(\lambda)$ onto $\mathcal{D}(\lambda)$, and we use $\mathcal{D}_m(\lambda)$ to denote the (u, v)-region of $\mathcal{D}(\lambda)$ where the area detector is located.

The CB projection for vertex position λ is the set of line integrals $g(u, v, \lambda)$ which connect $\vec{a}(\lambda)$ to points (u, v) in $\mathcal{D}(\lambda)$. We have

$$g(u, v, \lambda) = \int_0^\infty {\rm d}l\, f(\vec{a}(\lambda) + l\vec{\beta}(u, v, \lambda)), \qquad (u, v) \in \mathcal{D}(\lambda) \qquad (5)$$

where $\vec{\beta}(u, v, \lambda)$ is the unit vector along the line joining $\vec{a}(\lambda)$ to the point (u, v). In practice, $g(u, v, \lambda)$ is measured only for $(u, v) \in \mathcal{D}_m(\lambda)$. The projection is *non-truncated* when $g(u, v, \lambda)$ is zero outside $\mathcal{D}_m(\lambda)$. When this condition does not hold, the projection is *truncated*. With our definition of $f(\vec{r})$, truncation is avoided when $\Omega_\lambda \subset \mathcal{D}_m(\lambda)$, where

$$\Omega_\lambda = \left\{(u, v) \mid \quad |u \cos\eta - v \sin\eta| \leqslant R_{\text{FOV}} D/\sqrt{R^2 - R_{\text{FOV}}^2}\right\} \tag{6}$$

is the support of the projection of Ω onto $\mathcal{D}(\lambda)$†. Note that this region is unbounded in the axial direction. Therefore the condition $\Omega_\lambda \subset \mathcal{D}_m(\lambda)$ cannot be satisfied with a finite detector.

The long-object problem to be solved in this paper can be formulated as follows:

Definition 1. *Long-object problem: given the geometric characteristics of the detector and of the helix, design an exact reconstruction algorithm, which, for reconstruction of a given slice* $z = z_0$*, only requires CB data to be known in a finite region* $\lambda \in [z_0/h - \Delta\lambda, z_0/h + \Delta\lambda]$ *with* $\Delta\lambda$ *independent of the axial extent of the support of* $f(\vec{r})$.

Exact solutions to this problem have already been proposed by Schaller *et al* (1999) and Kudo *et al* (1999b), based on previous works by Tam *et al* (1998), Danielsson *et al* (1997) and Kudo *et al* (1998). All these methods assume that the detector satisfies the condition $\Omega_\lambda \bigcap B \subset \mathcal{D}_m(\lambda)$, where B is the region defined below. The algorithm developed in this paper relies on the same assumption.

Definition 2. *We call* B *the region of the detector plane* $\mathcal{D}(\lambda)$ *that is bounded by the CB projections onto* $\mathcal{D}(\lambda)$ *of the helix turns* $\{\vec{a}(\lambda'), \lambda' \in [\lambda, \lambda + 2\pi]\}$ *and* $\{\vec{a}(\lambda'), \lambda' \in [\lambda - 2\pi, \lambda]\}$ *located above and below the cone vertex* $\vec{a}(\lambda)$.

An illustration of region B is given in figure 2. Note that B depends on λ as it is defined as a subset of points of $\mathcal{D}(\lambda) \subset \mathbb{R}^3$. However, as all vertices along the helix are equivalent, all $B(\lambda)$ are congruent. Therefore, when it is seen as a subset of the (u, v)-plane, the region B is independent of λ. To simplify the exposition, we will omit the λ dependency of B.

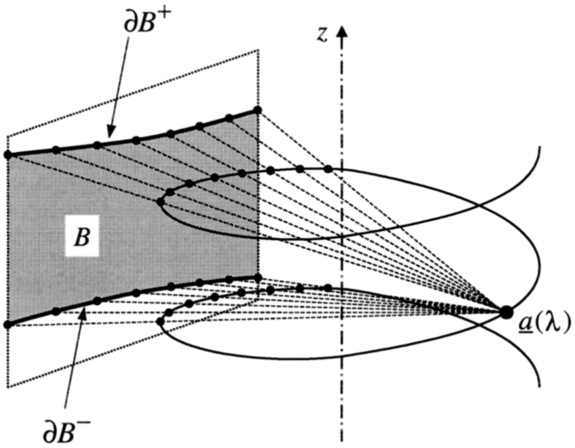

Figure 2. Illustration of region B. The two curves ∂B^+ and ∂B^- are the CB projections of the helix turns above and below $\vec{a}(\lambda)$. Any point on ∂B^+ and ∂B^- corresponds to a PI-line which links $\vec{a}(\lambda)$ to another vertex.

† In this paper, we use ‘projection onto $\mathcal{D}(\lambda)$’ as shorthand for ‘CB projection onto $\mathcal{D}(\lambda)$ with $\vec{a}(\lambda)$ as vertex’.

The upper and lower boundaries of region B play an important role in this paper. We denote them as ∂B^+ and ∂B^- respectively. Points on ∂B^+ and ∂B^- are strongly related to the notion of *PI-line*.

Definition 3. *A PI-line is a segment of line connecting two points of the helix separated by less than one pitch in the z-direction.*

Consider a point (u, v) on ∂B^+ in $\mathcal{D}(\lambda)$. By definition, (u, v) is the CB projection onto $\mathcal{D}(\lambda)$ of some cone vertex $\vec{a}(\lambda')$, with $\lambda < \lambda' \leqslant \lambda + 2\pi$. Therefore, the line connecting $\vec{a}(\lambda)$ to the point $(u, v) \in \partial B^+$ corresponds to a PI-line. Each point (u, v) on ∂B^+ and ∂B^- in $\mathcal{D}(\lambda)$ defines a PI-line in the (x, y, z)-space.

Note that each PI-line appears twice in the helical data set, once on ∂B^+ and once on ∂B^-. To understand this point, consider the PI-line $\mathcal{L}$ which contains $\vec{a}(\lambda)$ and $\vec{a}(\lambda')$, with $\lambda < \lambda' \leqslant \lambda + 2\pi$. Let (u, v) be the point of ∂B^+ where $\vec{a}(\lambda')$ is projected onto $\mathcal{D}(\lambda)$, and let (u', v') be the point of ∂B^- where $\vec{a}(\lambda)$ is projected onto $\mathcal{D}(\lambda')$. By construction, the PI-line $\mathcal{L}$ is equivalently defined by the triplets (u, v, λ) and (u', v', λ').

Owing to the symmetry of the helix, the coordinates of the CB projection of $\vec{a}(\lambda')$ onto $\mathcal{D}(\lambda)$ only depend on the difference $\lambda' - \lambda$. They are given by two odd functions, $u = U(\lambda' - \lambda)$ and $v = V(\lambda' - \lambda)$, which define the parametric equation of the boundaries of B. (See appendix A for the explicit expression of these functions.)

3. ZB algorithm

This section is dedicated to the derivation of the ZB algorithm. We begin the discussion with a summary of the CB-FBP algorithm developed by Kudo *et al* (1998). This algorithm plays a key role in the derivation of the ZB method. Next, we show in section 3.2 that dramatic simplifications of CB-FBP are possible for reconstruction of objects with CB projections vanishing on the boundaries of B. Particularly, we observe that CB-FBP automatically solves the long-object problem for such objects. Section 3.3 shows how these simplifications can be providentially used for reconstruction of $f(\vec{r})$, thanks to geometrical properties of the helix. As a result, we present then in section 3.4 a new solution to the long-object problem, called the ZB method.

3.1. The CB-FBP algorithm

This subsection describes the CB-FBP algorithm derived in Kudo *et al* (1998). This algorithm is suitable for quasi-exact reconstruction of $f(\vec{r})$ provided the CB projection $g(u, v, \lambda)$ are known for each $(u, v) \in B$ and for each $\lambda \in \mathbb{R}$. In practice data are only measured for a finite helix segment $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, and this condition is then equivalent to the two following conditions:

- The projections not measured for $\lambda \notin [\lambda_{\min}, \lambda_{\max}]$ must vanish within region B, i.e.

$$g(u, v, \lambda) = 0 \text{ for } (u, v) \in B \text{ when } \lambda \notin [\lambda_{\min}, \lambda_{\max}]. \tag{7}$$

- For each vertex position $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, $\Omega_\lambda \cap B$ must be in $\mathcal{D}_m(\lambda)$, since we know $g(u, v, \lambda)$ is zero outside Ω_λ (see equation (6)).

Note that the first condition comes down to stating that the exact CB-FBP reconstruction of a given z-slice is not possible when the helical path does not extend well beyond the limits of $f(\vec{r})$. Thus CB-FBP is not a solution to the long-object problem. In the next sections, we will show, however, that CB-FBP can be modified to overcome this constraint and provide a quasi-exact solution to the long-object problem.

The CB-FBP algorithm consists of the following steps, applied sequentially to each CB projection λ. We refer the reader to Kudo *et al* (1998) for the derivation of CB-FBP and details on the very slight approximation involved in this derivation†.

- Step 1: Apply the usual cone-angle weight to the data to get

$$\overline{g}(u,v,\lambda) = \frac{D}{\sqrt{D^2+u^2+v^2}} g(u,v,\lambda). \tag{8}$$

- Step 2: Calculate the partial derivatives of $\overline{g}(u,v,\lambda)$ with respect to u and v, and truncate to region B to obtain

$$\begin{aligned} g_u^{tr}(u,v,\lambda) &= \chi_B(u,v)\frac{\partial}{\partial u}\overline{g}(u,v,\lambda) \\ g_v^{tr}(u,v,\lambda) &= \chi_B(u,v)\frac{\partial}{\partial v}\overline{g}(u,v,\lambda) \end{aligned} \tag{9}$$

where

$$\chi_B(u,v) = \begin{cases} 1 & \text{if } (u,v) \in B \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

- Step 3: Apply a 2D convolution filter to $g_u^{tr}(u,v,\lambda)$ and $g_v^{tr}(u,v,\lambda)$ to get

$$\begin{aligned} g_u^F(u,v,\lambda) &= \iint_B \mathrm{d}u'\,\mathrm{d}v'\, g_u^{tr}(u',v',\lambda) q_u(u-u',v-v') \\ g_v^F(u,v,\lambda) &= \iint_B \mathrm{d}u'\,\mathrm{d}v'\, g_v^{tr}(u',v',\lambda) q_v(u-u',v-v') \end{aligned} \tag{11}$$

with kernels $q_u(u,v)$ and $q_v(u,v)$ obtained as the inverse 2D Fourier transform of the filters

$$Q_u(\nu_u,\nu_v) = -\frac{\mathrm{i}}{2\pi}\frac{\nu_u|\nu_u|}{\nu_u^2+\nu_v^2}, \qquad Q_v(\nu_u,\nu_v) = -\frac{\mathrm{i}}{2\pi}\frac{\nu_v|\nu_u|}{\nu_u^2+\nu_v^2} \tag{12}$$

where ν_u and ν_v are the frequencies corresponding to u and v, respectively.

- Step 4: Calculate a boundary term $g_{\text{bound}}^F(u,v,\lambda)$ which involves only values of $g(u,v,\lambda)$ on the curves ∂B^+ and ∂B^-.
- Step 5: Add the results of steps 3 and 4 to obtain the filtered projection:

$$g^F(u,v,\lambda) = g_u^F(u,v,\lambda) + g_v^F(u,v,\lambda) + g_{\text{bound}}^F(u,v,\lambda). \tag{13}$$

- Step 6: Backproject the filtered projection onto the image volume to yield:

$$f(\vec{r}) = \int_{\lambda_{\min}}^{\lambda_{\max}} \mathrm{d}\lambda\, \frac{D\sqrt{R^2+h^2}}{|(\vec{r}-\vec{a}(\lambda))\cdot\vec{1}_w|^2} g^F(u_0,v_0,\lambda) \tag{14}$$

where (u_0,v_0) are the coordinates of the CB projection of $\vec{r}$ onto $\mathcal{D}(\lambda)$, and $\vec{1}_w = \vec{1}_u \times \vec{1}_v$.

Some comments are warranted about steps 3 and 4. In step 3 and in the rest of the paper, the Fourier transform of a given function $p(u,v)$ acting in the (u,v)-plane is defined as

$$(\mathcal{F}p)(\nu_u,\nu_v) = \int_{\mathbb{R}} \mathrm{d}u \int_{\mathbb{R}} \mathrm{d}v\, p(u,v)\exp[-2\pi\mathrm{i}(u\nu_u + v\nu_v)]. \tag{15}$$

† Note that the equations in Kudo *et al* (1998) were written for a detector containing the z-axis, i.e. in the particular case where $D = R$.

In step 4, we have omitted the definition of the boundary term $g^F_{\mathrm{bound}}(u, v, \lambda)$ which can be found in Kudo *et al* (1998). For this paper, we only need to know that $g^F_{\mathrm{bound}}(u, v, \lambda)$ vanishes when $g(u, v, \lambda)$ is zero on ∂B^+ and ∂B^-. This property will be used in section 3.2.

Thanks to the multiplication by χ_B in equation (9), CB-FBP only requires measuring the data in the finite region $\Omega_\lambda \cap B$. However, this is not sufficient to solve the long-object problem as condition (7) is required for CB-FBP reconstruction. To understand why this condition is required, observe that the convolution kernels $q_u(u, v)$ and $q_v(u, v)$ have unbounded supports in u and v. Even though the support of $g^{tr}_u(u, v, \lambda)$ and $g^{tr}_v(u, v, \lambda)$ in equation (9) is within B by construction, the support of the filtered data $g^F_u(u, v, \lambda)$ and $g^F_v(u, v, \lambda)$ extends to the whole detector plane. Therefore, the CB projection $g(u, v, \lambda)$ contributes to the reconstruction of $f(x, y, z)$ at all values of z, as long as $g(u, v, \lambda)$ is non-identically zero within $\Omega_\lambda \cap B$.

For completeness, some remark on the approximation involved in the CB-FBP algorithm is in order because this approximation will be carried over to the ZB algorithm derived below. As shown in Kudo *et al* (1998), the CB-FBP algorithm is mathematically equivalent to building the 3D Radon transform of $f(x, y, z)$ and then inverting the Radon transform to recover $f(x, y, z)$. The approximation in CB-FBP arises from a non-exact handling of a very small number of Radon planes, namely those that have three intersections with a 2π segment of the helix and are such that the PI-line connecting the first intersection to the last intersection intersects Ω. In all numerical tests performed so far, the influence of this approximation was negligible compared with the discretization errors.

3.2. Reconstruction from data vanishing along the boundaries of B

In this section, we show that most of the complexity of CB-FBP stems from the data along ∂B^+ and ∂B^-, i.e. from the ray-sums corresponding to PI-lines. More specifically, we show that CB-FBP reduces to a simple FDK-like method when the CB data vanish along ∂B^+ and ∂B^- for all λ.

Consider a smooth hypothetical object $f_2(\vec{r})$ with CB projections $g_2(u, v, \lambda)$ zero along the boundaries of B for all λ, i.e. such that

$$g_2(u, v, \lambda) = \int_0^\infty \mathrm{d}l\, f_2(\vec{a}(\lambda) + l\vec{\beta}(u, v, \lambda)) = 0, \qquad \text{if } (u, v) \in \partial B^+ \bigcup \partial B^-. \tag{16}$$

We investigate below simplifications of CB-FBP that can be made for reconstruction of such objects.

The first simplification concerns equation (9). Since $f_2(\vec{r})$ is smooth, $g_2(u, v, \lambda)$ tends smoothly to zero when approaching the boundaries of B. Therefore, we can rewrite (9) in the form

$$\begin{aligned} g^{tr}_u(u, v, \lambda) &= \frac{\partial}{\partial u}\overline{g}^{tr}_2(u, v, \lambda) \\ g^{tr}_v(u, v, \lambda) &= \frac{\partial}{\partial v}\overline{g}^{tr}_2(u, v, \lambda) \end{aligned} \tag{17}$$

with

$$\overline{g}^{tr}_2(u, v, \lambda) = \frac{D}{\sqrt{D^2 + u^2 + v^2}}\chi_B(u, v) g_2(u, v, \lambda). \tag{18}$$

This simplification is possible because (a) the partial derivatives of $\chi_B(u, v)$ vanish everywhere except along ∂B^+ and ∂B^-, (b) $g_2(u, v, \lambda)$ is smooth and equals zero on ∂B^+ and ∂B^-.

Now, consider the 2D filtering step of equation (11). Take the 2D Fourier transform of $\overline{g}^{tr}_2(u, v, \lambda)$ with respect to u and v to get

$$(\mathcal{F}\overline{g}^{tr}_2)(\nu_u, \nu_v, \lambda) = \iint_{\mathbb{R}^2} \mathrm{d}u\, \mathrm{d}v \exp[-2\pi \mathrm{i}(u\nu_u + v\nu_v)]\overline{g}^{tr}_2(u, v, \lambda). \tag{19}$$

Next, recall that the derivatives in u and v in equation (17) correspond to multiplications by $2\pi \mathrm{i}\nu_u$ and $2\pi \mathrm{i}\nu_v$ in frequency space. Using these properties with the explicit expression (12) of the filters $Q_u(\nu_u, \nu_v)$ and $Q_v(\nu_u, \nu_v)$ in equation (11), we find

$$(\mathcal{F}g_u^F)(\nu_u, \nu_v, \lambda) = \frac{\nu_u^2|\nu_u|}{\nu_u^2 + \nu_v^2}(\mathcal{F}\overline{g}_2^{tr})(\nu_u, \nu_v, \lambda)$$
$$(\mathcal{F}g_v^F)(\nu_u, \nu_v, \lambda) = \frac{\nu_v^2|\nu_u|}{\nu_u^2 + \nu_v^2}(\mathcal{F}\overline{g}_2^{tr})(\nu_u, \nu_v, \lambda). \tag{20}$$

Therefore, the Fourier transform of the filtered projection in equation (13) is

$$(\mathcal{F}g_2^F)(\nu_u, \nu_v, \lambda) = (\mathcal{F}g_u^F)(\nu_u, \nu_v, \lambda) + (\mathcal{F}g_v^F)(\nu_u, \nu_v, \lambda) = |\nu_u|(\mathcal{F}\overline{g}_2^{tr})(\nu_u, \nu_v, \lambda) \tag{21}$$

because $g_{\mathrm{bound}}^F(u, v, \lambda)$ is zero under hypothesis (16). This result is important. It shows that the CB-FBP filtering step reduces to applying a 1D ramp filter along the u-axis when the CB data vanish along the boundaries of B. CB-FBP reconstruction of objects such as $f_2(\vec{r})$ can thus be achieved as follows:

- B-FDK1: Weight and truncate data $g_2(u, v, \lambda)$ as in equation (18)
- B-FDK2: Apply a ramp filter along the u-direction to get
$$g_2^F(u, v, \lambda) = \int_{-\infty}^{\infty} \mathrm{d}u'\, \overline{g}_2^{tr}(u', v, \lambda)h(u - u') \tag{22}$$
where $h(u)$ is the kernel of the ramp filter, i.e. the inverse 1D Fourier transform of $|\nu_u|$.
- B-FDK3: Backproject the filtered projection $g_2^F(u, v, \lambda)$ onto the image volume according to equation (14).

We use the acronym B-FDK for this modified version of CB-FBP because it is identical to the FDK-like algorithm described in Yan and Leahy (1992), except for the truncation to region B in equation (18).

Hypothesis (16) not only leads to considerable simplifications of CB-FBP, but also indicates a way to solve the long-object problem. To understand this point, see figure 3 which shows the support of the filtered projection $g_2^F(u, v, \lambda)$. This support is bounded in v because $\overline{g}_2^{tr}(u, v, \lambda)$ is zero outside $\Omega_\lambda \cap B$ and the 1D ramp filter only extends the support of $\overline{g}_2^{tr}(u, v, \lambda)$ in the u-direction. Therefore, the CB backprojection of $g_2^F(u, v, \lambda)$ only contributes to z-slices within a finite range $h\lambda - \Delta z \leqslant z \leqslant h\lambda + \Delta z$ with Δz independent of the axial extent of the support of $f(\vec{r})$. The explicit expression of Δz is given in appendix C. The existence of such a Δz shows that B-FDK solves the long-object problem for $f_2(\vec{r})$ because the reconstruction of the slice $z = z_0$ only requires us to know the CB data over the range $h\lambda \in [z_0 - \Delta z, z_0 + \Delta z]$.

3.3. Zeroing the data along the boundaries of B

In this section, we investigate the possibility of using the results of section 3.2 for reconstruction of any object $f(\vec{r})$. The idea we have in mind is the following. Assume that we can build a smooth function $f_1(\vec{r})$ which satisfies the following condition

$$g_1(u, v, \lambda) = \int_0^{\infty} \mathrm{d}l\, f_1(\vec{a}(\lambda) + l\vec{\beta}(u, v, \lambda)) = g(u, v, \lambda), \qquad \text{if } (u, v) \in \partial B^+ \bigcup \partial B^-. \tag{23}$$

Consider then the difference $g_2(u, v, \lambda) = g(u, v, \lambda) - g_1(u, v, \lambda)$ between the measured data and the CB projections of $f_1(\vec{r})$. By construction, the data $g_2(u, v, \lambda)$ correspond to a hypothetical object $f_2(\vec{r}) = f(\vec{r}) - f_1(\vec{r})$ which has CB projections equal to zero on ∂B^+ and ∂B^-. The hypothesis of equation (16) is thus satisfied, $f_2(\vec{r})$ can be reconstructed using the B-FDK algorithm, and from $f_2(\vec{r})$ one readily obtains $f(\vec{r}) = f_1(\vec{r}) + f_2(\vec{r})$. We have just

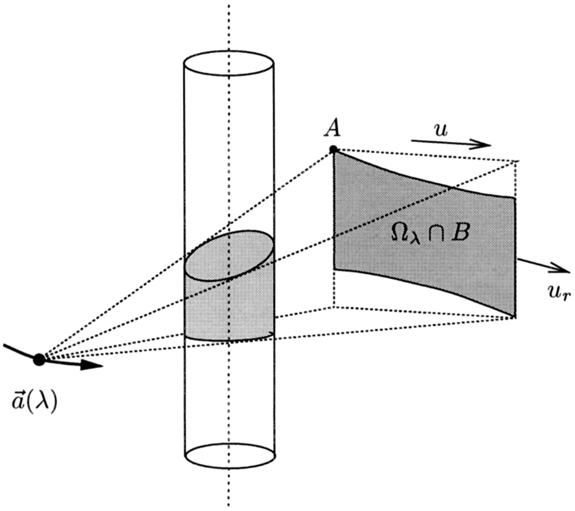

Figure 3. Contribution of $g(u, v, \lambda)$ to the reconstruction of $f(\vec{r})$ in the B-FDK algorithm. A mask is first applied to $g(u, v, \lambda)$ which limits non-zero data to the region $\Omega_\lambda \cap B$. Next, the ramp filter is applied in the u-direction, which extends the support of the masked projection to a parallelogram. The CB backprojection of this parallelogram defines a finite volume in the FOV, outside which $g(u, v, \lambda)$ brings no contribution.

sketched a new method for reconstruction of $f(\vec{r})$. As B-FDK is a solution to the long-object problem for $f_2(\vec{r})$, one can expect this method to be a solution to the long-object problem for $f(\vec{r})$, provided $f_1(\vec{r})$ can be defined at any z-position using only data on a finite segment of the helix.

Everything in the above method relies on the possibility of building a function $f_1(\vec{r})$ which satisfies (23), i.e. a function $f_1(\vec{r})$ which has the same line integrals as $f(\vec{r})$ on all PI-lines. We show below that this is possible thanks to a particular property of the helix which, to our knowledge, was first introduced by Danielsson *et al* (1997). This property states that there exists always a PI-line passing through a point $\vec{r}$ such that $x^2 + y^2 < R^2$. Furthermore, this PI-line is unique:

Helix property. *Any point $\vec{r}$ within the region $x^2 + y^2 < R^2$, where R is the radius of the helix, belongs to one and only one PI-line.*

A geometrical proof of this property can be found in Danielsson *et al* (1997). For completeness, we give an alternative analytic proof in appendix B.

We will denote the unique PI-line containing $\vec{r}$ by $\mathcal{L}(\vec{r})$ and will use $\lambda_L(\vec{r})$ and $\lambda_U(\vec{r})$, with $\lambda_U(\vec{r}) > \lambda_L(\vec{r})$ for the angular positions of the two helix points at the extremities of $\mathcal{L}(\vec{r})$. Define now an arbitrary smooth profile function $\rho(\vec{r})$ which satisfies the following conditions:

- The support of $\rho(\vec{r})$ is contained within the cylinder of axis z and of radius R_ρ with $R_{\text{FOV}} < R_\rho < R$.
- The integrals of $\rho(\vec{r})$ are non-zero on all PI-lines which intersect Ω.

With such a function, one can construct a function $f_1(\vec{r})$ satisfying equation (23) as

$$f_1(\vec{r}) = \frac{\rho(\vec{r})}{\int_{\mathcal{L}(\vec{r})} \mathrm{d}\vec{s}\, \rho(\vec{s})} g(\mathcal{L}(\vec{r})) \tag{24}$$

where $g(\mathcal{L}(\vec{r}))$ is a shorthand notation for the value of the ray-sum measured on the PI-line containing $\vec{r}$. By definition, $\mathcal{L}(\vec{r})$ is constant when $\vec{r}$ moves along a PI-line. Therefore, integrating $f_1(\vec{r})$ along a PI-line, all factors except $\rho(\vec{r})$ in equation (24) can be factored out of the integral and it is easily seen that $f_1(\vec{r})$ satisfies equation (23). As we have seen in section 2, the PI-line $\mathcal{L}(\vec{r})$ can be found twice in the data set, once on ∂B^+ and once on ∂B^-. By averaging the two available values, which are equal for consistent data, one gets

$$g(\mathcal{L}(\vec{r})) = \int_{\mathcal{L}(\vec{r})} \mathrm{d}\vec{s}\, f(\vec{s}) = \frac{1}{2}\{g(U(\Delta\lambda), V(\Delta\lambda), \lambda_L(\vec{r})) + g(U(-\Delta\lambda), V(-\Delta\lambda), \lambda_U(\vec{r}))\} \tag{25}$$

where $\Delta\lambda = \lambda_U(\vec{r}) - \lambda_L(\vec{r})$ and the functions $U(\lambda)$ and $V(\lambda)$ are used to describe the boundaries of B, as explained in section 2. When the signal-to-noise ratio is sufficient, one may instead of the average in equation (25) use only one of the two estimates of $g(\mathcal{L}(\vec{r}))$ to reduce the data set required for reconstruction (see appendix C).

The construction of $f_1(\vec{r})$ according to (24) is very simple as it amounts to a weighted backprojection of the small subset of projection data located on the boundaries of B. Besides, due to the definition of PI-lines with the helix property above, data within the range $h\lambda \in [z_0 - 2\pi h, z_0 + 2\pi h]$ are more than sufficient for calculation of $f_1(\vec{r})$ at slice $z = z_0$ (see appendix C). Therefore, the method outlined in the first paragraph of this subsection defines a solution to the long-object problem for $f(\vec{r})$. This method, called the ZB-method, is summarized in the next section.

3.4. Pseudo-code description of the ZB method

Using the results in the previous subsections, the ZB algorithm can now be defined as a five-step procedure:

- Step 1: Using equations (24) and (25), calculate the image $f_1(\vec{r})$ that has the same PI-line integrals as $f(\vec{r})$.
- Step 2: Calculate the CB projections $g_1(u, v, \lambda)$ of $f_1(\vec{r})$ for $(u, v) \in B$.
- Step 3: Subtract $g_1(u, v, \lambda)$ from the original data to obtain data $g_2(u, v, \lambda) = g(u, v, \lambda) - g_1(u, v, \lambda)$ which vanish for all PI-lines.
- Step 4: Reconstruct $f_2(\vec{r})$ from $g_2(u, v, \lambda)$, using the B-FDK algorithm.
- Step 5: Add $f_2(\vec{r})$ to $f_1(\vec{r})$ to get $f(\vec{r}) = f_1(\vec{r}) + f_2(\vec{r})$.

To carefully analyse how the ZB algorithm solves the long-object problem, we need to calculate the helix segment $\lambda \in [z_0/h - \Delta\lambda_{\mathrm{ZB}}, z_0/h + \Delta\lambda_{\mathrm{ZB}}]$ required for calculation of $f(\vec{r})$ at a given slice $z = z_0$. As already stated this segment is finite and does not depend on $f(\vec{r})$, thanks to the local character of the 1D ramp filter used to reconstruct $f_2(\vec{r})$, and also thanks to the geometrical properties of the PI-lines used to define $f_1(\vec{r})$. The explicit calculation of $\Delta\lambda_{\mathrm{ZB}}$ is given in appendix C where it is shown that $\Delta\lambda_{\mathrm{ZB}}$ only depends on R_ρ/R. See figure 4 for a plot of $(2\Delta\lambda_{\mathrm{ZB}})$ as a function of R_ρ/R.

4. Implementation and performance of the ZB algorithm

4.1. Implementation

The implementation of the ZB algorithm in section 3.4 is straightforward, but we have found that a number of precautions help to reduce the discretization errors. These empirical

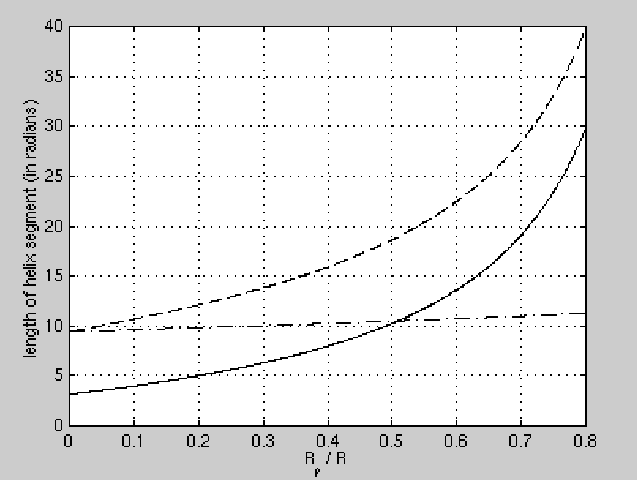

Figure 4. Helix segment required for reconstruction of a z-slice with the ZB algorithm. This helix segment only depends on the ratio R_ρ/R where R_ρ defines the support of $f_1(\vec{r})$. The full curve is the minimum helix segment $2\Delta\lambda_{\mathrm{ZB}}$, obtained when appropriately switching between data from ∂B^+ and ∂B^- to evaluate PI-line integrals. The broken curve is the helix segment required when one always uses the two estimates available for each PI-line integral (see equation (25)). The dot-dashed curve, given for comparison purposes, is the helix segment required by the exact method of Schaller *et al* (1999).

implementation details are described below, though they are not meant to be optimal. The variables Δ_x, Δ_y, Δ_z, Δ_u and Δ_v used for this description denote the sampling distances in x, y, z, u and v.

4.1.1. Choice of the profile function. The function $\rho(\vec{r})$ can be chosen arbitrarily as long as it is smooth and satisfies the conditions given in section 3.3. We use a smooth bell-shaped radial profile:

$$\rho(x,y,z) = \begin{cases} \cos^2\left(\dfrac{\pi}{2}\dfrac{\sqrt{x^2+y^2}}{R_\rho}\right) & \text{if } \sqrt{x^2+y^2} \leqslant R_\rho \\ 0 & \text{otherwise} \end{cases} \qquad (26)$$

with some $R_\rho > R_{\mathrm{FOV}}$. In general larger values of R_ρ lead to better reconstructions, but this unfortunately requires larger matrix sizes for $f_1(\vec{r})$ and consequently for $g_1(u, v, \lambda)$ and $g_2(u, v, \lambda)$. We use values $R_\rho \simeq 1.1 R_{\mathrm{FOV}}$.

4.1.2. Calculation of the indicator function. For discrete implementation, the function $\chi_B(u, v)$ is smoothed in the v direction. Instead of using $\chi_B(u, v)$, we use the function

$$\chi_B^{\mathrm{sm}}(u,v) = \int_{-w}^{w} \mathrm{d}v'\, \chi_B(u, v-v') \frac{\pi}{4w} \cos\left(\frac{\pi v'}{2w}\right) \qquad (27)$$

with typically $w = 2\Delta_v$ so that the smoothed boundary extends over about five pixels. The value of the projection data along ∂B^+ or ∂B^- in equation (25) is also obtained using the same smoothing profile so that each value is estimated as the weighted average of about five data points.

4.1.3. Calculation of f_1. The image $f_1(\vec{r})$ is obtained by backprojecting the data along ∂B^+ and ∂B^- in each projection, with a weight defined by the profile $\rho(\vec{s})$. The backprojection is ray-driven with normalization for the variable number of PI-lines which contribute to each image voxel. After the backprojection, $f_1(\vec{r})$ is smoothed by a 3D Hamming filter.

4.1.4. Calculation of g_1. The reprojection step is implemented using the method of Joseph (1982) with linear interpolation. Note that one only needs to calculate $g_1(u, v, \lambda)$ within the region B.

4.1.5. Reconstruction of f_2. We apply a standard implementation of the FDK algorithm, with a voxel driven backprojection and a ramp filter apodized by a Hamming window, cut-off at Nyquist's frequency.

4.1.6. Addition of the two images. Though the last step of the ZB algorithm appears trivial, it is not because the two discrete images $f_1(\vec{r})$ and $f_2(\vec{r})$ are obtained in different ways and therefore have unmatched spatial resolution. Specifically, $f_2(\vec{r})$ has a lower resolution owing to the interpolations involved in the reprojection step (step 2 in section 3.4) and in the B-FDK reconstruction. We approximate this loss of resolution by a 3D spatial invariant filter, with the following frequency representation:

$$\begin{aligned} S(\nu_x, \nu_y, \nu_z) = {} & \mathrm{sinc}^2(\nu_t \Delta_x)\, \mathrm{sinc}^2(\nu_z \Delta_z) W_{\mathrm{Ham}}(\nu_t R \Delta_u / D)\, \mathrm{sinc}^2(\nu_t R \Delta_u / D) \\ & \times \mathrm{sinc}^2(\nu_z R \Delta_v / D) \end{aligned} \tag{28}$$

where $\nu_t = \sqrt{\nu_x^2 + \nu_y^2}$, $\mathrm{sinc}(x) \equiv \sin(\pi x)/(\pi x)$, and the Hamming window is

$$W_{\mathrm{Ham}}(x) = \begin{cases} 0.50\ (1 + \cos(2\pi x)) & \text{if } |x| \leqslant 1/2 \\ 0 & \text{otherwise.} \end{cases} \tag{29}$$

The first two factors in equation (28) model the effect of the linear interpolations in the reprojection step (calculation of $g_1(u, v, \lambda)$), the third one models the apodization of the ramp filter, and the two last factors model the effect of the linear interpolations in the CB backprojection step of the B-FDK algorithm.

Using the filter of equation (28), step 5 of the ZB algorithm is replaced by $f(\vec{r}) = s * f_1(\vec{r}) + f_2(\vec{r})$, where $*$ represents 3D convolution and $s = \mathcal{F}^{-1} S$ is the convolution kernel corresponding to the filter S. The support of this kernel is to a good approximation local and therefore the 3D convolution is implemented via the FFT. Though empirical, this trick was found to improve the reconstruction of objects with low-contrast internal structures, such as the Shepp phantom.

4.2. Simulation studies

In this section, we show results obtained with the ZB algorithm for two phantoms, the classical 3D Shepp phantom and a more complicated head phantom designed by Schaller (1998). Both phantoms were scaled to just fit in the sphere of radius 100 mm, centred at the origin $\vec{r} = 0$. The radius of the field-of-view was thus $R_{\mathrm{FOV}} = 100$ mm.

Two helical paths were used for the simulations, one with a pitch $P = 27$ mm and the other one with $P = 54$ mm. In both cases, the radius of the helix was $R = 400$ mm and there were 1000 vertices per helix turn. With these parameters the helix segment required to reconstruct a slice $z = z_0$ is given by $\lambda \in [z_0/h - \Delta\lambda_{\mathrm{ZB}}, z_0/h + \Delta\lambda_{\mathrm{ZB}}]$ with $\Delta\lambda_{\mathrm{ZB}} = 2.8$ rad (see equations (48) and (50) in appendix C). The first vertex of the helix was always at

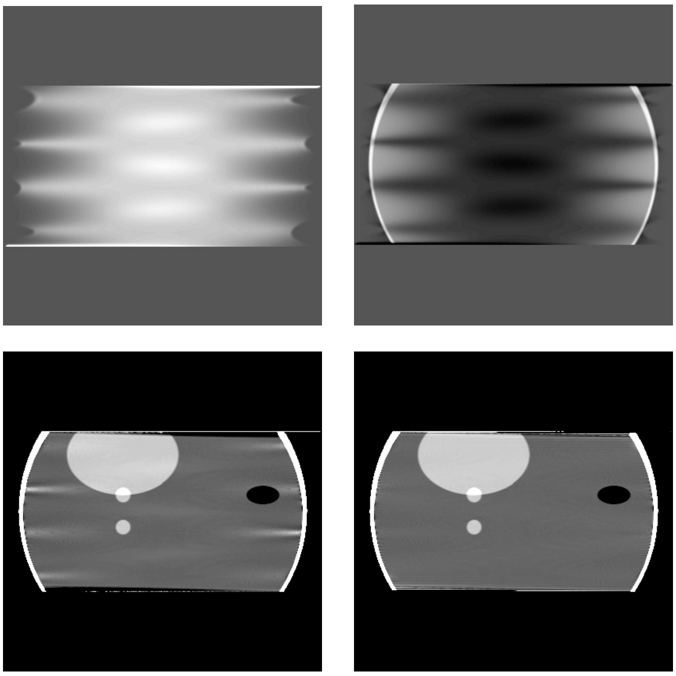

Figure 5. Performance of the ZB algorithm. Central vertical slice $x = 0$ through the Shepp phantom: (top left) reconstruction of $f_1(\vec{r})$; (top right) reconstruction of $f_2(\vec{r})$; (bottom left) sum of $f_1(\vec{r})$ and $f_2(\vec{r})$ (grey-scale window [1.01, 1.035]); (bottom right) sum of $f_1(\vec{r})$ and $f_2(\vec{r})$ after applying the filter of equation (28) to $f_1(\vec{r})$ (grey-scale window [1.01, 1.035]).

location $z = -81$ mm while the last one was at $z = 81$ mm. Six (three) helix turns were needed to cover this region with $P = 27$ mm ($P = 54$ mm). As defined, the helix did not extend over the whole axial extent of the phantom. Imaging a central ROI in the two phantoms thus corresponded to solving the long-object problem.

Data were simulated analytically on a rectangular area detector located at the distance $D = 800$ mm from the cone vertex, with its centre at $(u, v) = (0, 0)$. The detector pixel size was 1×1 mm^2. There were 410 pixels in the u-direction, while the number of detector rows was 43 for $P = 27$ mm and 86 for $P = 54$ mm. This number was defined in each case so as to fully cover the region $\Omega_\lambda \cap B$.

The images displayed in figures 5, 6, 7 and 8 correspond all to reconstructions on a grid of $400 \times 400 \times 200$ cubic voxels of side 0.5 mm.

In figure 5, we first show the results obtained for the Shepp phantom with the pitch $P = 54$ mm. With such a high pitch value, approximate algorithms have been shown to introduce low-frequency artefacts and also to modify the shape of small internal objects (see e.g. Kudo *et al* (1999a) and Noo *et al* (1999)). As can be seen, the ZB algorithm does not exhibit such problems. The ZB reconstruction obtained after smoothing $f_1(\vec{r})$ with the filter S of equation (28) is very good, even though a highly compressed grey-scale window was used for display. Note that streak artefacts appear in the reconstruction of $f(\vec{r})$ when the filter S is not used. These artefacts are clearly due to unmatched spatial resolution between the reconstructions of $f_1(\vec{r})$ and $f_2(\vec{r})$.

In figures 6, 7 and 8, we show the results obtained for the head phantom of Schaller, using different grey-scale windows and both pitch values, $P = 27$ mm and $P = 54$ mm. This head

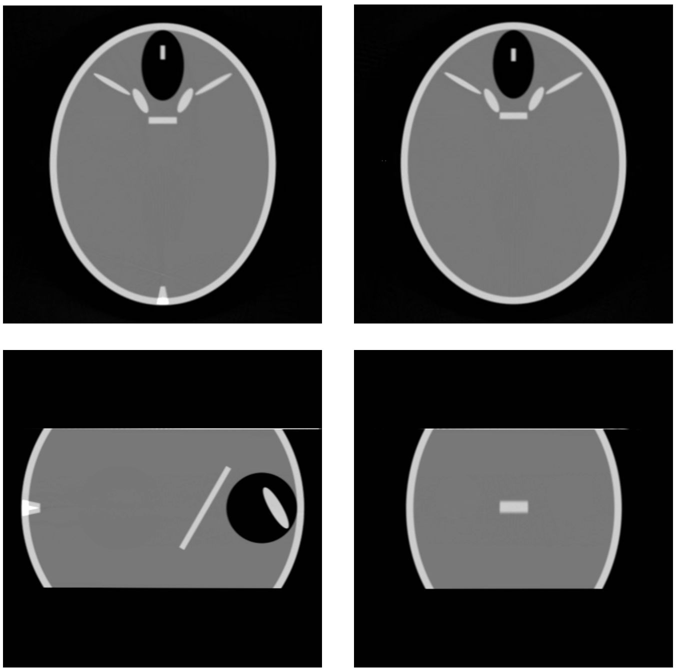

Figure 6. ZB reconstruction of the head-like phantom of Schaller (1998). Pitch $P = 54$ mm. Grey-scale window [0, 2.5]. The filter of equation (28) was used: (top left) slice $z = 1.5$ mm; (top right) slice $z = 5.5$ mm; (bottom left) slice $x = 0$ mm; (bottom right) slice $y = 26$ mm.

phantom contains high-density sharp internal structures and is thus much more sensitive to discretization errors than the classical Shepp phantom. Using a large grey-scale window, as in figure 6, the ZB reconstruction seems perfect. On the contrary, when using a compressed greyscale window, artefacts resulting from discretization errors are visible. They illustrate the limitations of the ZB method. In particular, we have observed that some streak artefacts (such as those crossing the oblique white cylinder in the bottom-left image in figure 8) appear when the pitch decreases.

The reconstructions times for both pitch values, $P = 27$ mm and $P = 54$ mm, were similar. On a SUN SPARC Station (300 MHz), the computation of $f_1(\vec{r})$, $g_1(u, v, \lambda)$ and $f_2(\vec{r})$ took respectively 364 s, 7304 s, and 11 500 s CPU time. The total CPU time for the computation of $f(\vec{r})$ was thus only 1.7 times longer than the time for the simple B-FDK reconstruction. Little effort was made at optimizing code.

Stability in the presence of noise is an important feature of an algorithm, which should certainly be checked with the ZB method as the final image is obtained as the sum of two images, one of which, $f_2(\vec{r})$, may have negative values. We have simulated data for a long uniform cylinder of radius 80 mm, and the same data acquisition parameters as above with the pitch $P = 54$ mm. Pseudo-random Poisson noise was then added to the data, corresponding to a maximum number of photons per detector pixel of 12 269. The variance of the reconstruction was estimated for each image by taking the difference with the corresponding noise-free reconstruction and by calculating the mean square of the pixel values in a square region of interest of size 100 mm in the central transaxial slice. For comparison, we also performed a 2D short-scan fan-beam reconstruction of one slice of the cylinder, using the same maximum count per detector pixel and the same parameters for data acquisition and reconstruction. The

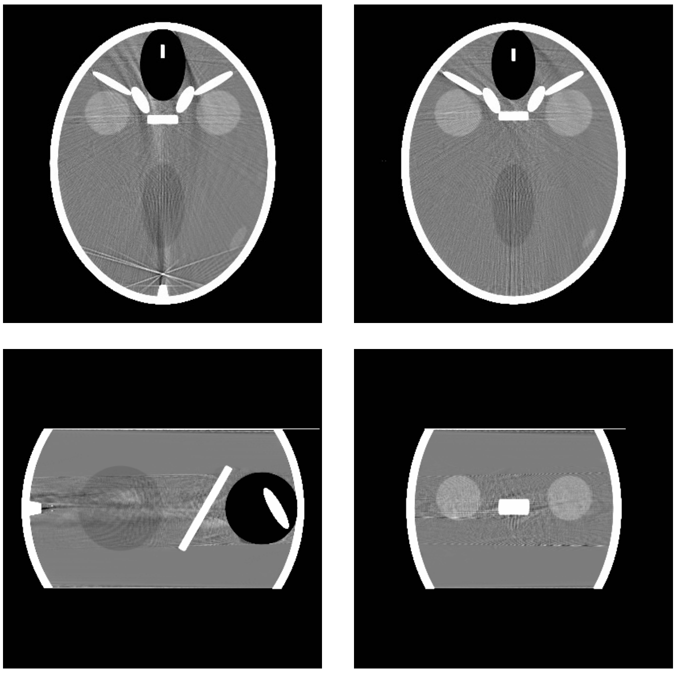

Figure 7. ZB reconstruction of the head-like phantom of Schaller (1998). Pitch $P = 54$ mm. Greyscale window [1.01, 1.09]. The filter of equation (28) was used: (top left) slice $z = 1.5$ mm; (top right) slice $z = 5.5$ mm; (bottom left) slice $x = 0$ mm; (bottom right) slice $y = 26$ mm.

standard deviations estimated that way were respectively 0.000 45, 0.0171 and 0.0172 for the images $f_1(\vec{r})$, $f_2(\vec{r})$ and $f(\vec{r}) = f_1(\vec{r}) + f_2(\vec{r})$, and 0.021 for the 2D reconstruction. For this test, the smoothing filter S of equation (28) was not applied to f_1. This result indicates that (a) the variance in the final ZB reconstruction almost exclusively arises from the $f_2(\vec{r})$ image, as can be expected since $f_1(\vec{r})$ is the backprojection of projection data which have not undergone any high-pass filtering, and (b) the variance in the ZB reconstruction is comparable to the variance in 2D fan-beam reconstruction. Note that the higher standard deviation observed with the 2D fan-beam reconstruction is explained by the absence of axial interpolation in that reconstruction.

5. Conclusion

We have described and implemented a quasi-exact algorithm for the long-object problem in helical cone-beam CT: the ZB algorithm. Like two other methods recently proposed for this problem (Schaller *et al* 1999, Kudo *et al* 1999b), the ZB method is based on the generalized formula of Grangeat linking truncated CB projections to the 3D Radon transform. Except for this foundation, however, the three algorithms have little in common. Comparing their properties and performance could therefore improve our insight into the long-object problem. An important open issue, in particular, is to determine the minimum length of helix segment needed to reconstruct a z-slice when the CB data are measured within the region B. The different minimum segments required by the ZB and PHI methods, shown in figure 4, is a good illustration of our yet incomplete understanding of the problem.

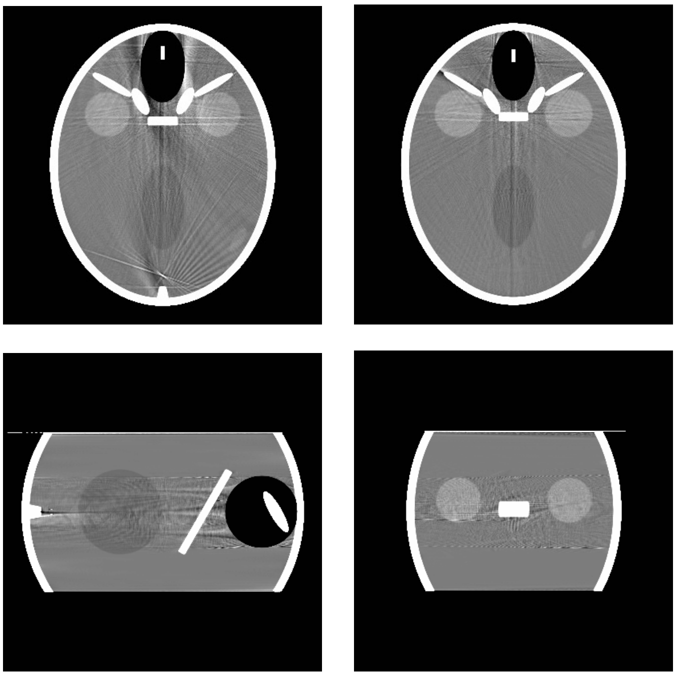

Figure 8. ZB reconstruction of the head-like phantom of Schaller (1998). Pitch $P = 27$ mm. Greyscale window [1.01, 1.09]. The filter of equation (28) was used: (top left) slice $z = 1.5$ mm; (top right) slice $z = 5.5$ mm; (bottom left) slice $x = 0$ mm; (bottom right) slice $y = 26$ mm.

In its present implementation, the ZB method does not have the form of a filtered backprojection because the image $f_1(\vec{r})$ obtained as a weighted backprojection of the PI-line data must be calculated and reprojected before proceeding with the FBP reconstruction of the modified data $g_2(u, v, \lambda)$. The algorithm is nevertheless efficient numerically, the reconstruction time being only at most two times longer than for the helical FDK algorithms. Together with the conceptual simplicity and ease of implementation, this numerical efficiency is an attractive feature of the ZB method.

Future work will also focus on further improving the algorithm. A key parameter here is the profile function $\rho(\vec{r})$ involved in the definition of $f_1(\vec{r})$. A simple bell-shaped profile was used in this paper, but alternative choices could accelerate reconstruction or improve image quality. One possibility we have in mind is to use a profile $\rho(\vec{r})$ which has its support in a cylindrical ring exterior to the support Ω of $f(\vec{r})$. With such a choice, the reconstruction of $f(\vec{r})$ within its support would no longer require the explicit calculation of the intermediate image $f_1(\vec{r})$ provided the CB projections $g_2(u, v, \lambda)$ can be evaluated directly from the measured data. We hope that this approach, or a similar one, could lead to a true FBP-like version of the ZB method.

Acknowledgments

MD is *onderzoeksdirecteur* with the Fund for Scientific Research (FWO, Belgium). FN is *chargé de recherches* with the National Fund for Scientific Research (FNRS, Belgium).

Appendix A

To define the region B we use coordinates (u_r, v_r) rotated by the angle η of equation (4). In this rotated system, the u_r-axis is lying in the (x, y) plane:

$$\begin{aligned} u_r &= u\cos\eta - v\sin\eta \\ v_r &= u\sin\eta + v\cos\eta. \end{aligned} \tag{30}$$

The CB projection of a point (x, y, z) onto the detector $\mathcal{D}(\lambda_1)$ is:

$$\begin{aligned} u_r &= \frac{D(-x\sin\lambda_1 + y\cos\lambda_1)}{R - x\cos\lambda_1 - y\sin\lambda_1} \\ v_r &= \frac{D(z - h\lambda_1)}{R - x\cos\lambda_1 - y\sin\lambda_1}. \end{aligned} \tag{31}$$

Replacing (x, y, z) by the coordinates of a vertex point $\vec{a}(\lambda_2)$ (equation (2)), we obtain the parametric equation of ∂B^+ and ∂B^- in the rotated coordinate system. Owing to the symmetry of the helix, this equation only depends on $\lambda = \lambda_2 - \lambda_1$:

$$\begin{aligned} U_r(\lambda) &= D\sin\lambda/(1-\cos\lambda) \\ V_r(\lambda) &= hD\lambda/R(1-\cos\lambda) \end{aligned} \tag{32}$$

with $\lambda \in [0, 2\pi]$ for ∂B^+ and $\lambda \in [-2\pi, 0]$ for ∂B^-. The corresponding expressions in the original coordinate system used in the paper are obtained by rotation:

$$\begin{aligned} U(\lambda) &= U_r(\lambda)\cos\eta + V_r(\lambda)\sin\eta \\ V(\lambda) &= -U_r(\lambda)\sin\eta + V_r(\lambda)\cos\eta. \end{aligned} \tag{33}$$

Eliminating $\lambda \in [0, 2\pi]$ from the equations (32), one finds the explicit equation of ∂B^+ as given by Tam *et al* (1998):

$$v_r = V_r(u_r) = \frac{Dh}{R}\left(1 + \frac{u_r^2}{D^2}\right)\left[\frac{\pi}{2} - \arctan\left(\frac{u_r}{D}\right)\right] \tag{34}$$

where the arctan is defined with its range in $[-\pi/2, +\pi/2]$. By symmetry, the explicit equation of ∂B^- is $v_r = -V_r(-u_r)$. An important property of these two curves is that they are respectively convex and concave, and hence each have a maximum of two intersections with any straight line in the detector plane.

Appendix B

We demonstrate below the property of the helix used in section 3.3: *any point $\vec{r}_0 = (x_0, y_0, z_0)$ within the region $x_0^2 + y_0^2 < R^2$, where R is the radius of the helix, belongs to one and only one PI-line.*

To prove this property, we use the notation $\vec{r}_0 = (r_0\cos\mu_0, r_0\sin\mu_0, z_0)$ with $r_0 < R$. PI-lines are identified by two vertices λ_1 and λ_2, and a variable t denotes the position along the PI-line, with $t = 0$ and $t = 1$ corresponding respectively to $\vec{a}(\lambda_2)$ and $\vec{a}(\lambda_1)$.

Our aim is to show that $\vec{r}_0$ belongs to one and only one PI-line. This means that there is a unique triplet λ_1, λ_2, t, with $t \in (0, 1)$ and $0 < \lambda_2 - \lambda_1 < 2\pi$, such that:

$$\begin{aligned} x_0 &= Rt\cos\lambda_1 + R(1-t)\cos\lambda_2 \\ y_0 &= Rt\sin\lambda_1 + R(1-t)\sin\lambda_2 \\ z_0 &= ht\lambda_1 + h(1-t)\lambda_2. \end{aligned} \tag{35}$$

First, we consider the family of all PI-lines which cross the line V_0 that contains $\vec{r}_0$ and is parallel to the z-axis. This is a one-dimensional family which can be parametrized by λ_1. For each $\lambda_1 \in \mathbb{R}$, we solve the two first equations in (35) for λ_2 and t. Thus the family of PI-lines is determined by two functions $\lambda_2(x_0, y_0, \lambda_1)$ and $t(x_0, y_0, \lambda_1)$ such that

$$\begin{aligned} r_0 \cos \mu_0 &= Rt(x_0, y_0, \lambda_1) \cos \lambda_1 + R(1 - t(x_0, y_0, \lambda_1)) \cos \lambda_2(x_0, y_0, \lambda_1) \\ r_0 \sin \mu_0 &= Rt(x_0, y_0, \lambda_1) \sin \lambda_1 + R(1 - t(x_0, y_0, \lambda_1)) \sin \lambda_2(x_0, y_0, \lambda_1) \\ 0 &< \lambda_2(x_0, y_0, \lambda_1) - \lambda_1 < 2\pi. \end{aligned} \tag{36}$$

The solution of (36) is unique and is given by:

$$t(x_0, y_0, \lambda_1) = \frac{R^2 - r_0^2}{2R[R - r_0 \cos(\mu_0 - \lambda_1)]} \tag{37}$$

and λ_2 is determined by:

$$\cos\left(\frac{\lambda_2(x_0, y_0, \lambda_1) - \lambda_1}{2}\right) = \frac{r_0 \sin(\mu_0 - \lambda_1)}{\sqrt{R^2 + r_0^2 - 2Rr_0 \cos(\mu_0 - \lambda_1)}}. \tag{38}$$

Note that for $r_0 < R$, the denominator in (37) or (38) never vanishes, $0 < t(x_0, y_0, \lambda_1) < 1$, and $\lambda_2(x_0, y_0, \lambda_1)$ and $t(x_0, y_0, \lambda_1)$ are C^1 functions of λ_1.

Each PI-line in the family (38) intersects the vertical line V_0 through $\vec{r}_0$ in a point with coordinates (x_0, y_0, z) where:

$$z(x_0, y_0, \lambda_1) = h\lambda_1 t(x_0, y_0, \lambda_1) + h\lambda_2(x_0, y_0, \lambda_1)(1 - t(x_0, y_0, \lambda_1)). \tag{39}$$

Therefore the PI-line contains $\vec{r}_0$ if and only if

$$z(x_0, y_0, \lambda_1) = z_0. \tag{40}$$

The theorem is proven by showing that this equation has one and only one solution for λ_1. This is seen by calculating

$$\frac{\mathrm{d}z(x_0, y_0, \lambda_1)}{\mathrm{d}\lambda_1} = h\left((\lambda_1 - \lambda_2)\frac{\mathrm{d}t}{\mathrm{d}\lambda_1} + t + (1 - t)\frac{\mathrm{d}\lambda_2}{\mathrm{d}\lambda_1}\right) \tag{41}$$

where we have omitted the arguments of t and of λ_2. The two derivatives are most easily calculated by differentiating equation (36) with respect to λ_1. This leads to a pair of linear equations in $\mathrm{d}t/\mathrm{d}\lambda_1$ and $\mathrm{d}\lambda_2/\mathrm{d}\lambda_1$, which are solved to yield:

$$\begin{aligned} \frac{\mathrm{d}t(x_0, y_0, \lambda_1)}{\mathrm{d}\lambda_1} &= t \cot \frac{\lambda_2 - \lambda_1}{2} \\ \frac{\mathrm{d}\lambda_2(x_0, y_0, \lambda_1)}{\mathrm{d}\lambda_1} &= \frac{t}{1 - t}. \end{aligned} \tag{42}$$

Putting these expressions into (41) we get

$$\frac{\mathrm{d}z(x_0, y_0, \lambda_1)}{\mathrm{d}\lambda_1} = 2ht\left(1 - \frac{\lambda_2 - \lambda_1}{2} \cot \frac{\lambda_2 - \lambda_1}{2}\right) \tag{43}$$

which is easily shown to be strictly positive when $0 < \lambda_2 - \lambda_1 < 2\pi$. We thus conclude that $\mathrm{d}z/\mathrm{d}\lambda_1 > 0$. Furthermore one has $\lim_{\lambda_1 \to -\infty} z = -\infty$ and $\lim_{\lambda_1 \to \infty} z = \infty$. Therefore, as the function $z(x_0, y_0, \lambda_1)$ is continuous, the equation $z(x_0, y_0, \lambda_1) = z_0$ has one and only one solution λ_1, which together with equation (38) defines the unique PI-line containing $\vec{r}_0$. This concludes the proof.

Appendix C

We study in this appendix the length of the helix segment required to reconstruct a z-slice in the image $f(\vec{r})$, using the ZB algorithm. We will successively determine: (a) the helix segment required to reconstruct a z-slice in $f_2(\vec{r})$ using the B-FDK algorithm, (b) the segment required to calculate the data $g_1(u, v, \lambda)$ for a given vertex λ, and (c) the segment required to calculate a z-slice in $f_1(\vec{r})$. All results will then be put together in a concluding paragraph. In the whole appendix we use the system of rotated coordinates (u_r, v_r) defined in appendix A, with the u_r-axis in the (x, y)-plane. To simplify the notations we omit the subscript r for these coordinates, except in figure 3.

C.1. Helix segment required by the B-FDK method

Let $\lambda_0 = 0$. We determine to which z-slices the CB projection $g_2(u, v, \lambda_0)$ contributes when processed using the B-FDK algorithm. Consider the detector plane $\mathcal{D}(\lambda_0)$ in figure 3. The region B^F where the filtered projection is non-zero is easily seen in this figure. It is obtained by extending the support of B in the direction where the 1D ramp filter is applied, i.e. in the direction of slope $\mathrm{d}v/\mathrm{d}u = h/R$, since we use rotated (u, v)-axes. Let A be the point where ∂B^+ intersects the left limit of the projected FOV of $f_2(\vec{r})$, i.e. the point where ∂B^+ meets the vertical line

$$u = u_A = -R_\rho D/\sqrt{R^2 - R_\rho^2}. \tag{44}$$

By construction, B^F is limited above (in v) by the straight line $R(v - v_A) = h(u - u_A)$. Being on a boundary of B, A is the CB projection onto $\mathcal{D}(\lambda_0)$ of some vertex $\vec{a}(\lambda_A)$. Thus, from results in appendix A, we have

$$\begin{aligned} u_A &= D\sin\lambda_A/(1 - \cos\lambda_A) \\ v_A &= hD\lambda_A/R(1 - \cos\lambda_A). \end{aligned} \tag{45}$$

Using (44) and (45), one finds:

$$\lambda_A = \pi + 2\arcsin(R_\rho/R) \tag{46}$$

which is nothing but the well-known orbit range required for the 2D short-scan fan-beam reconstruction of a disc of radius R_ρ. From the expression of u_A and v_A above, we calculate now the equation of the plane Π defined by the straight line $R(v - v_A) = h(u - u_A)$ and the vertex $\vec{a}(\lambda_0)$. We obtain

$$\Pi = \left\{(x, y, z) \in \mathbb{R}^3 | \frac{z}{h} = \left(\frac{\lambda_A - \sin\lambda_A}{1 - \cos\lambda_A}\right)\left(1 - \frac{x}{R}\right) + \frac{y}{R}\right\}. \tag{47}$$

This plane represents the upper limit of the (x, y, z)-region to which the CB projection $g_2(u, v, \lambda_0)$ contributes. The highest z-slice in this region is found by calculating the intersection of Π with the surface of the cylinder of radius R_ρ. One easily gets:

$$z_{\max} = h\Delta\lambda_{\mathrm{FDK}} = h\left(\frac{\lambda_A - \sin\lambda_A}{1 - \cos\lambda_A}\right)\left\{1 + \frac{R_\rho}{R}\sqrt{1 + \left(\frac{1 - \cos\lambda_A}{\lambda_A - \sin\lambda_A}\right)^2}\right\} \tag{48}$$

where $\Delta\lambda_{\mathrm{FDK}}$ is a quantity depending only on R_ρ/R. We can thus conclude that the CB projection $g_2(u, v, \lambda_0)$ only contributes to z-slices in the range $z \in [-z_{\max}, z_{\max}]$. Equivalently, we note that the B-FDK reconstruction of a given slice $z = z_0$ only requires data to be known on the helix segment $\lambda \in [z_0/h - \Delta\lambda_{\mathrm{FDK}}, z_0/h + \Delta\lambda_{\mathrm{FDK}}]$.

C.2. Helix segment required for g_1

We consider the calculation of the CB projection of $f_1(\vec{r})$ at some point $(u, v) \in B$ in $\mathcal{D}(\lambda)$. From the definition of $f_1(\vec{r})$ in equation (24), we have

$$\begin{aligned} g_1(u, v, \lambda) &= \int_0^\infty \mathrm{d}l\, f_1(\vec{a}(\lambda) + l\vec{\beta}(u, v, \lambda)) \\ &= \int_I \mathrm{d}\vec{r} \frac{\rho(\vec{r})}{\int_{\mathcal{L}(\vec{r})} ds \rho(\vec{s})} g(\mathcal{L}(\vec{r})) \end{aligned} \qquad (49)$$

where I is the line connecting $\vec{a}(\lambda)$ to the point $(u, v) \in B$. For any point $\vec{r}$ along this line, we denote as in section 3.3 the PI-line containing $\vec{r}$ by $\mathcal{L}(\vec{r})$ and the corresponding vertices at its extremities by $\lambda_L(\vec{r})$ and $\lambda_U(\vec{r})$.

We first show that $\lambda_L(\vec{r}) \leqslant \lambda \leqslant \lambda_U(\vec{r})$ when $(u, v) \in B$, where (u, v) is the projection of $\vec{r}$. This means that the projection of the vertices $\vec{a}(\lambda_L(\vec{r}))$ and $\vec{a}(\lambda_U(\vec{r}))$ onto $\mathcal{D}(\lambda)$ respectively belong to ∂B^- and ∂B^+. This can be proven by contradiction. Assume for example that the projections of $\vec{a}(\lambda_L(\vec{r}))$ and $\vec{a}(\lambda_U(\vec{r}))$ both belong to ∂B^+. As $\vec{r} \in I$ is a point on the PI-line linking these two vertices, (u, v) should be between the projections of $\vec{a}(\lambda_L(\vec{r}))$ and $\vec{a}(\lambda_U(\vec{r}))$. That is (u, v) should be on a segment of line linking two points of ∂B^+. But ∂B^+ is convex. Thus (u, v) should be outside region B, which is in contradiction with the hypothesis that $(u, v) \in B$.

Now, observe that $g(\mathcal{L}(\vec{r}))$ is always zero when $\mathcal{L}(\vec{r})$ does not intersect the support of $f_1(\vec{r})$. Therefore, the contribution of point $\vec{r}$ to $g_1(u, v, \lambda)$ is non-zero only if $\lambda_U(\vec{r}) - \lambda_L(\vec{r}) \leqslant \lambda_A$, where λ_A is given by equation (46). Combining this result with the previous one, we conclude that $g_1(u, v, \lambda)$ can be calculated from the helix segment $[\lambda, \lambda + \lambda_A]$ if one evaluates $g(\mathcal{L}(\vec{r}))$ using only data from $\mathcal{D}(\lambda_U(\vec{r}))$, or $[\lambda - \lambda_A, \lambda]$ if one only uses $\mathcal{D}(\lambda_L(\vec{r}))$. More generally, $g_1(u, v, \lambda)$ can be obtained from any helix segment of length λ_A which contains λ, that is, from any segment $[\lambda - \delta, \lambda - \delta + \lambda_A]$ where $0 \leqslant \delta \leqslant \lambda_A$, if one switches between data from $\mathcal{D}(\lambda_L(\vec{r}))$ and $\mathcal{D}(\lambda_U(\vec{r}))$ to evaluate $g(\mathcal{L}(\vec{r}))$.

C.3. Helix segment required for f_1

To calculate $f_1(\vec{r})$ in the slice $z = 0$, we only need to know the line integrals of $f(\vec{r})$ for all PI-lines which intersect this slice within the field-of-view of radius R_ρ. Since these data are needed to evaluate $g_1(u, v = 0, \lambda = 0)$, we observe that the helix segment required here is nothing but a subset of the segment needed for calculation of $g_1(u, v, \lambda = 0)$.

C.4. Conclusion

The previous results can now be integrated, using the property that $\Delta\lambda_{\mathrm{FDK}} \geqslant \lambda_A/2$ (as can be easily seen from equation (48) and the fact that $\lambda_A > \pi$). To reconstruct the slice $z = 0$ in the image $f_2(\vec{r})$, we need $g_2(u, v, \lambda)$, and therefore $g(u, v, \lambda)$ and $g_1(u, v, \lambda)$, in the interval $[-\Delta\lambda_{\mathrm{FDK}}, \Delta\lambda_{\mathrm{FDK}}]$. To estimate $g_1(u, v, \lambda)$ in this interval with the smallest possible helix segment, we use for each λ the helix segment of length λ_A that has its centre at the shortest distance from $\vec{a}(\lambda = 0)$ while containing $\vec{a}(\lambda)$, that is

- the segment $[\lambda, \lambda + \lambda_A]$ for $-\Delta\lambda_{\mathrm{FDK}} \leqslant \lambda \leqslant -\lambda_A/2$,
- the segment $[-\lambda_A/2, \lambda_A/2]$ for $-\lambda_A/2 \leqslant \lambda \leqslant \lambda_A/2$,
- the segment $[\lambda - \lambda_A, \lambda]$ for $+\lambda_A \leqslant \lambda \leqslant \Delta\lambda_{\mathrm{FDK}}$.

All these segments are contained within $[-\Delta\lambda_{\mathrm{FDK}}, +\Delta\lambda_{\mathrm{FDK}}]$. Since the calculation of $f_1(\vec{r})$ does not need more data than those used to get $g_1(u, v, \lambda)$, we conclude that the helix segment required to reconstruct the slice $z = 0$ with the ZB algorithm is given by $[-\Delta\lambda_{\mathrm{ZB}}, \Delta\lambda_{\mathrm{ZB}}]$, with

$$\Delta\lambda_{\mathrm{ZB}} = \Delta\lambda_{\mathrm{FDK}}. \tag{50}$$

Similarly, to reconstruct the slice $z = z_0$, one needs the helix segment $[z_0/h - \Delta\lambda_{\mathrm{ZB}}, z_0/h + \Delta\lambda_{\mathrm{ZB}}]$.

Note that the above interval is the minimum interval needed for the ZB method. When one averages the data on ∂B^+ and ∂B^- as in equation (25), one must use $\Delta\lambda_{\mathrm{ZB}} = \Delta\lambda_{\mathrm{FDK}} + \lambda_A$.

References

Danielsson P E, Edholm P, Eriksson J and Magnusson Seger M 1997 Towards exact reconstruction for helical cone-beam scanning of long objects. A new detector arrangement and a new completeness condition *Proc. 1997 Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine (Pittsburgh, PA)* ed D W Townsend and P E Kinahan, pp 141–4

Feldkamp L A, Davis L C and Kress J W 1984 Practical cone-beam algorithm *J. Opt. Soc. Am.* A **A6** 612–19

Grangeat P 1991 Mathematical framework of cone-beam 3D reconstruction via the first derivative of the Radon transform *Mathematical Methods in Tomography (Lecture Notes in Mathematics 1497)* ed G T Herman, A K Louis and F Natterer (Berlin: Springer) pp 66–97

Joseph T M 1982 An improved algorithm for reprojecting rays through pixel images *IEEE Trans. Med. Imag.* **1** 192–201

Kudo H, Noo F and Defrise M 1998 Cone-beam filtered-backprojection algorithm for truncated helical data *Phys. Med. Biol.* **43** 2885–909

——1999b Quasi-exact filtered-backprojection algorithm for long-object problem in helical cone-beam tomography *IEEE Trans. Med. Imaging* submitted

Kudo H, Park S, Noo F and Defrise M 1999a Performance of quasi-exact cone-beam filtered-backprojection algorithm for axially truncated data *IEEE Trans. Nucl. Sci.* **46** 608–17

Kudo H and Saito T 1992 Helical-scan computed tomography using cone-beam projections *Conf. Rec. 1991 IEEE Med. Imag. Conf. (Santa Fe, NM)* pp 1958–62

Noo F, Defrise M and Clackdoyle R 1999 Single-slice rebinning method for helical cone-beam CT *Phys. Med. Biol.* **44** 561–70

Schaller S, Flohr T and Steffen P 1997 New, efficient Fourier-reconstruction method for approximate image reconstruction in spiral cone-beam CT at small cone angles *Proc. SPIE* **3032** 213–24

Schaller S 1998 Practical image reconstruction for cone-beam tomography *PhD Thesis* University of Erlangen

Schaller S, Noo F, Sauer F, Tam K C, Lauritsch G and Flohr T 1999 Exact Radon rebinning algorithm for the long object problem in helical cone-beam CT *IEEE Trans. Med. Imaging* submitted

Tam K C, Samarasekera S and Sauer F 1998 Exact cone-beam CT with a spiral scan *Phys. Med. Biol.* **43** 1015–24

Wang G, Lin T H, Cheng P and Shinozaki D M 1993 A general cone-beam reconstruction algorithm *IEEE Trans. Med. Imaging* **12** 486–96

Yan X and Leahy R M 1992 Cone-beam tomography with circular, elliptical and spiral orbits *Phys. Med. Biol.* **37** 493–506