



Published in final edited form as:

Phys Med Biol. 2009 August 7; 54(15): 4625–4644. doi:10.1088/0031-9155/54/15/001.

Accurate helical cone-beam CT reconstruction with redundant data

Harald Schöndube^{1,2}, Karl Stierstorfer², and Frédéric Noo¹

¹UCAIR, Department of Radiology, University of Utah, Salt Lake City, UT, USA

²Siemens Healthcare, Siemensstr. 1, 91301 Forchheim, Germany

Abstract

We present a new image reconstruction algorithm for helical cone-beam computed tomography (CT). This algorithm is designed for data collected at or near maximum pitch, and provides a theoretically exact and stable reconstruction while beneficially using all measured data. The main operations involved are a differentiated backprojection and a finite-support Hilbert transform inversion. These operations are applied onto M-lines, and the beneficial use of all measured data is gained from averaging three volumes reconstructed each with a different choice of M-lines. The technique is overall similar to that presented by one of the authors in a previous publication, but operates volume-wise, instead of voxel-wise, which yields a significantly more efficient reconstruction procedure. The algorithm is presented in detail. Also, preliminary results from computer-simulated data are provided to demonstrate the numerical stability of the algorithm, the beneficial use of redundant data and the ability to process data collected with an angular flying focal spot.

1. Introduction

The ongoing increase in the number of detector rows has created a need for more and more sophisticated helical cone-beam (CB) reconstruction algorithms in x-ray computed tomography (CT). So far, for computational complexity reasons, preference has been given to analytical reconstruction algorithms. These can be divided into two classes: the approximate and the theoretically exact and stable (TES) algorithms. The approximate algorithms are currently the method of choice for commercial CT scanners; see, e.g., Feldkamp *et al* (1984), Stierstorfer *et al* (2002), Flohr *et al* (2003), Stierstorfer *et al* (2004), Heuscher *et al* (2004), Kudo *et al* (2004), Shechter *et al* (2004), Köhler *et al* (2006), Tang and Hsieh (2007), Zamyatin *et al* (2006), Katsevich *et al* (2007). They are attractive because they are able to use all measured data. But they result in reconstructions that are distorted by CB artifacts that can be severe for imaging with 64 detector rows or more.

In this paper, we focus on the class of TES algorithms, which due to their mathematically exact nature do not suffer from CB artifacts. We are particularly interested in the difficult problem of performing a TES reconstruction while beneficially using all measurements. Several TES algorithms have been published; we categorize them into $1-\pi$ and $n-\pi$ methods.

The $1-\pi$ methods, such as Katsevich's algorithm (Katsevich 2002) and the π -line two-step Hilbert algorithm (Zou and Pan 2004, Pack *et al* 2005, Ye *et al* 2005, Schöndube *et al* 2007a), essentially use a data angular coverage of 180° for each voxel, as viewed from the

location of the voxel. That is, they perform reconstruction by backprojecting only filtered data defined inside the Tam–Danielsson (TD) window (Tam *et al* 1998, Danielsson *et al* 1997). For a standard helical CB data acquisition geometry, this means that even in the best case scenario only 73% of the measured data are effectively used in terms of noise. The best case scenario is data acquisition at maximum pitch, which is the pitch value such that the TD window just fits within the detector area. Therefore, the $1-\pi$ methods significantly waste dose when applied to standard helical CB data. This issue could be solved by adapting the x-ray beam to the TD window to prevent the patient from being exposed to unnecessary dose, but such an adaptation is mechanically challenging to build and could potentially remove flexibility in pitch selection. A preferable solution is to find a way to efficaciously include the data measured outside the TD window into a TES algorithm such that no dose is wasted. Including these data then serves as a way for reducing image noise while maintaining resolution.

The $n-\pi$ methods, and in particular the $3-\pi$ methods, rely on the theoretical observations made in Proksa *et al* (2000). They extend the data set on which the reconstruction is based to the general case of a $n-\pi$ detector window (Bontus *et al* 2003, Katsevich 2004, Bontus *et al* 2005, Katsevich 2006), where n is an odd integer. Thus, they essentially perform reconstruction at any given location using a relative data angular coverage of n times 180° . This allows a good increase in noise-effective data usage, namely to 86% for $n = 3$ and 89% for $n = 5$ (Proksa *et al* 2000), but does not allow full data utilization. Also, it requires fixing the pitch to relatively small values, e.g., for $n = 3$, the pitch needs to be fixed at about one third of the maximum pitch that was defined above for the $1-\pi$ methods. This results in a significant increase of the data acquisition time for a given axial coverage and thus induces a higher risk of motion artifacts, especially for imaging applications where a high pitch is typically required.

Up to now, it has not been clear if efficient TES reconstruction can be achieved while using *all* data acquired at or near maximum pitch with a standard rectangular (flat or curved) detector. It is also unclear if a TES algorithm can be built such that it uses most or all data on the detector for arbitrary pitch values. This paper focuses on the first of these questions. We start our investigation from an observation that was made in Pack *et al* (2005), namely that in some circumstances, the x-ray linear attenuation coefficient at a given location can be reconstructed in three different ways that involve each a different amount of data and yield together full data utilization. Moreover, a reduction of image noise can be achieved by averaging the three different reconstructions. Unfortunately, this observation was based on a computationally demanding procedure, as reconstruction at any location \underline{x} required a differentiated backprojection (DBP) and a subsequent inverse Hilbert transform (HT) on three lines that contain \underline{x} .

We present here an efficient algorithm that allows TES reconstruction using all the data acquired at or near maximum pitch with a standard cylindrical detector. This algorithm is a volume-based implementation of the observation made in Pack *et al* (2005), which was found by extending our efficient π -line-based DBP–HT method (Schöndube *et al* 2007a, 2007b) to the general case of M-lines (i.e., lines that connect a source position with a point on the detector; see section 2.2) (‘M-line’ is an acronym for ‘measured line’). The extension provides a means to efficiently reconstruct the attenuation function on any desired stack of nutating slices that are made of M-lines and cover the volume of interest (VOI). We apply it to reconstruct the VOI three times, using stacks of nutating slices that require different amounts of data and that involve, altogether, all measurements. Averaging of the multiple VOI reconstructions then yields a result that effectively utilizes all measured data. Reconstruction on the different stacks of nutating slices can be performed in a sequential manner, or as a whole, within a single loop on the projections, which offers some advantages in terms of computational effort and data management, and is our preferred approach.

The paper is organized as follows. First, a mathematical description of the data acquisition geometry and of the DBP on M-lines is given in section 2. Then, in section 3, we explain how data outside the TD window can effectively be involved using the DBP, and describe our new algorithm in detail. Last, preliminary reconstruction results are shown in section 4, demonstrating high accuracy, the ability to make significant use of redundant data and the ability to reduce noise through this use. A discussion-and-conclusion section is given after these results.

2. Mathematical background

2.1. Data acquisition geometry

To describe the data acquisition geometry we use a system of Cartesian coordinates, x , y and z , that is attached to the object under examination. This system is laid out such that the z -axis corresponds to the direction in which the object is translated. A vector in this coordinate system is denoted as $\underline{x} = [x, y, z]^T$, with $[\cdot]^T$ denoting the transpose of a matrix or vector. The 3D distribution of x-ray attenuation coefficients to be reconstructed is $f(\underline{x})$.

The motion of the vertex point (i.e., the radiation source) relative to the object is described as

$$\underline{a}(\lambda) = [R_0 \cos(\lambda + \lambda_0), R_0 \sin(\lambda + \lambda_0), z_0 + h\lambda]^T, \quad (1)$$

where R_0 is the helix radius, and $2\pi h$ is the helix pitch. The position of the vertex point is therefore defined by a single parameter, $\lambda \in [\lambda_{\text{start}}, \lambda_{\text{end}}]$, where λ_{start} denotes the first and λ_{end} denotes the last position where measurements are taken. The vertex path is adjusted by λ_0 and z_0 such that at $\lambda = 0$ the source is located in the plane $z = z_0$ at a polar angle λ_0 .

The algorithm is designed for a third generation CT scanner, where the x-ray data are measured using a curved-area detector; see figure 1. The curvature of the detector is such that the detector sits on the cylindrical surface of radius D that is centered on the line parallel to the z -axis through the vertex point. Any point P on the area detector can be specified using two coordinates, w and γ , which are defined relative to a midplane and a midline within this plane, as explained below. The area detector is pixelated into N_{rows} rows of N_{cols} pixels that are equidistantly spaced in w and γ ; the sampling steps are denoted as Δw and $\Delta\gamma$, so that each pixel is of size $(D\Delta\gamma) \times \Delta w$, with $\Delta\gamma$ expressed in radians.

We call the plane that contains the vertex point and is orthogonal to the z -axis the *midplane*, and we call the line within the midplane that diverges from the vertex point and intersects the z -axis the *midline*. Coordinate w specifies the signed distance between P and the midplane; it is measured in the direction of z . Coordinate γ is the angle between the midline and the line that connects the vertex point with the orthogonal projection of P onto the midplane. This angle is measured positively in the clockwise direction, relative to the z -axis. By construction, $(\gamma, w) = (0, 0)$ defines the point where the midline intersects the area detector.

The detector is centered on the origin, $(\gamma, w) = (0, 0)$, except for a quarter-detector-pixel offset in γ , as in current CT scanners. The maximum fan angle is denoted as γ_{max} , and implicitly defines the radius of the field of view (FOV) within which the object is contained. The first detector row is at position $w = w_{\text{min}}$ and the last detector row is at position $w = w_{\text{max}}$, with $w_{\text{min}} = -w_{\text{max}}$ and $w_{\text{max}} = (N_{\text{rows}} - 1)\Delta w/2$. Note that these definitions will be extensively used later.

To mathematically describe the data measured at a given (γ, w) location, we introduce three orthogonal unit vectors:

$$\underline{e}_u(\lambda) = [-\sin(\lambda + \lambda_0), \cos(\lambda + \lambda_0), 0], \quad (2)$$

$$\underline{e}_v(\lambda) = [-\cos(\lambda + \lambda_0), -\sin(\lambda + \lambda_0), 0], \quad (3)$$

$$\underline{e}_w = [0, 0, 1]. \quad (4)$$

These vectors are such that $\underline{e}_v(\lambda)$ is parallel to the midline, and $\underline{e}_u(\lambda) = \underline{e}_v(\lambda) \times \underline{e}_w$. Thus, $\underline{e}_u(\lambda)$, $\underline{e}_v(\lambda)$ and \underline{e}_w form a 3D orthonormal basis rotating with the vertex point.

The line that connects the vertex point $\underline{a}(\lambda)$ to the (γ, w) location on the area detector is of direction

$$\underline{\alpha}(\lambda, \gamma, w) = \frac{(D \sin \gamma \underline{e}_u(\lambda) + D \cos \gamma \underline{e}_v(\lambda) + w \underline{e}_w)}{\sqrt{D^2 + w^2}}, \quad (5)$$

and the data measured along this line are

$$g(\lambda, \gamma, w) = \int_0^\infty f(\underline{a}(\lambda) + t \underline{\alpha}(\lambda, \gamma, w)) dt. \quad (6)$$

The TD window plays a significant role in our algorithm. It is defined as the region of the detector inside which the data are measured that are necessary (Tuy 1983, Danielsson *et al* 1997) and sufficient (Zou and Pan 2004) for TES reconstruction. When using only data inside this window, each voxel is reconstructed from a 180° range of data, as viewed from the voxel perspective. The upper and lower boundaries of the TD window are the CB projections of the helix turns above and below the current source position onto the detector; in our notation, they are given by (Noo *et al* 2003)

$$w_{\text{top}} = \frac{Dh \pi/2 - \gamma}{R_0 \cos \gamma}, \quad w_{\text{bottom}} = -\frac{Dh \pi/2 + \gamma}{R_0 \cos \gamma}. \quad (7)$$

We call the pitch value such that the TD window just fits within the detector area *maximum pitch*. This value, denoted by $2\pi h_{\text{max}}$, corresponds to a pitch factor,

$p_{\text{max}} = 2\pi h_{\text{max}} \cdot \frac{D}{N_{\text{rows}} \Delta w R_0}$, that depends on γ_{max} and the number of detector rows, N_{rows} , according to the formula

$$p_{\text{max}} = \pi \frac{N_{\text{rows}} - 1}{N_{\text{rows}}} \frac{\cos \gamma_{\text{max}}}{\pi/2 + \gamma_{\text{max}}}. \quad (8)$$

For a maximum fan angle of 26° , we have $p_{\text{max}} \approx 1.4$.

Our algorithm is not applicable for arbitrary values of γ_{max} and of the pitch factor, p . The following conditions must be met:

$$\begin{matrix} \gamma_{\max} < \gamma_{\text{crit}}, \\ P_{\min} < P < P_{\max}, \end{matrix} \quad (9)$$

where γ_{crit} is the smallest positive root of the equation $(\pi/2 + \delta) \tan \delta = 1$, namely $\gamma_{\text{crit}} \approx 26.24^\circ$, and

$$p_{\min} = \pi \frac{N_{\text{rows}} - 1}{N_{\text{rows}}} \sin \gamma_{\max}. \quad (10)$$

The condition on γ_{\max} is only introduced to guarantee $p_{\min} < p_{\max}$, whereas the constraint on p is derived from the algorithm, as will be discussed later.

2.2. Differentiated backprojection on M-lines

The backprojection part of the algorithm presented in this paper is essentially a generalization of the DBP-0 variant of our DBP-HT algorithm (Schöndube *et al* 2007a) from π -lines to ‘M-lines’. A π -line is a line that connects two source positions separated by less than one helix turn. In contrast, an M-line is any line that connects a source position with a point on the detector. By construction, any point within the helix cylinder belongs to a multitude of M-lines, and any π -line is an M-line, whereas the converse is not true.

In the same way as for the DBP on π -lines, the outcome of the DBP on M-lines is not the function $f(\underline{x})$. It is the HT of $f(\underline{x})$ along an M-line. To achieve the reconstruction of $f(\underline{x})$, an inverse HT has to be applied after the DBP is performed. For this task, we use an implementation of the formula first presented by Söhngen (1937) (Noo *et al* 2004), as it requires the HT of $f(\underline{x})$ only to be known over the support of $f(\underline{x})$. A detailed description of how we implement Söhngen’s formula can be found in Schöndube *et al* (2007a equations (12) and (13)).

The link between the DBP and the HT along M-lines can be described as follows. Let $(\mathcal{H}f)(\underline{x}, \underline{\omega})$ be the HT of f at \underline{x} along the unit vector $\underline{\omega}$, i.e.,

$$(\mathcal{H}f)(\underline{x}, \underline{\omega}) = - \int_{-\infty}^{\infty} \frac{f(\underline{x} - t\underline{\omega})}{\pi t} dt, \quad (11)$$

where the integral has to be understood in the sense of a Cauchy principal value. Let $\underline{\omega}(\lambda, \underline{x})$ denote the unit vector pointing from $\underline{a}(\lambda)$ toward \underline{x} :

$$\underline{\omega}(\lambda, \underline{x}) = \frac{\underline{x} - \underline{a}(\lambda)}{\|\underline{x} - \underline{a}(\lambda)\|}. \quad (12)$$

Furthermore, let $\mathcal{DBP}\{\lambda_a, \lambda_b, \underline{x}\}$ be an operator that denotes the DBP of $g(\lambda, \gamma, w)$ over the interval $[\lambda_a, \lambda_b]$ at \underline{x} . Using this notation we can formulate the relation that allows the reconstruction of $f(\underline{x})$ along M-lines using the DBP, as introduced in Pack *et al* (2005), as

$$(\mathcal{H}f)(\underline{x}, \underline{\omega}(\lambda_M, \underline{x})) = \mathcal{DBP}\{\lambda_M, \lambda_2(\underline{x}), \underline{x}\} - \mathcal{DBP}\{\lambda_1(\underline{x}), \lambda_M, \underline{x}\}. \quad (13)$$

In this relation, $\lambda_1(\underline{x})$ and $\lambda_2(\underline{x})$ correspond to the source positions where the projection of \underline{x} just enters (respectively, leaves) the TD window on the detector, and λ_M specifies the direction

of the HT, which is along the M-line that connects $\underline{a}(\lambda_M)$ to \underline{x} , i.e., along $\underline{\omega}(\lambda_M, \underline{x})$. Relation (13) is defined for any point \underline{x} inside the helix cylinder, and can be implemented whenever \underline{x} is illuminated with no interruptions over the intervals³ $[\lambda_M, \lambda_2(\underline{x})]$ and $[\lambda_1(\underline{x}), \lambda_M]$. A point \underline{x} is said to be illuminated over the interval $[a, b]$ if the line connecting $\underline{a}(\lambda)$ to \underline{x} is covered by the measurements for all $\lambda \in [a, b]$.

The value at a given point \underline{x} on the left-hand side of equation (13) depends on λ_M , as, for different choices of λ_M , the value of $\underline{\omega}(\lambda_M, \underline{x})$ indicating the direction of the HT through \underline{x} will be different. It is only after performing the inverse HT that the reconstruction result at \underline{x} is mathematically independent from the choice of λ_M . It is also important to note that λ_M is *not* required to be within the interval $[\lambda_1(\underline{x}), \lambda_2(\underline{x})]$. Let $\lambda_i(\underline{x})$ and $\lambda_o(\underline{x})$ be the first (respectively, last) source position at which the projection of \underline{x} just enters (respectively, leaves) *the area detector*. If there is no interrupted illumination over $[\lambda_i(\underline{x}), \lambda_o(\underline{x})]$, then λ_M can be selected anywhere between $\lambda_i(\underline{x})$ and $\lambda_o(\underline{x})$ (Pack *et al* 2005). Conditions (9) guarantee that there is no such interrupted illumination for all points \underline{x} within the FOV, as will be explained later. The possibility of choosing λ_M freely from a large number of values provides us with a degree of freedom in selecting the set of M-lines used for reconstruction. In the following section, we will discuss how this degree of freedom can be used to take redundant data measured outside the TD window into account.

The DBP can be implemented in various ways; see, e.g., Pack *et al* (2005), Zou and Pan (2004), Schöndube *et al* (2007a, 2007b), Yu *et al* (2007). In Schöndube *et al* (2007a, 2007b), Yu *et al* (2007), it was shown that using a rebinning step for the DBP significantly improves efficiency and noise properties. Using a rebinning step also provides an easy way of processing flying focal spot (FFS) data, since this feature affects then only the rebinning step, not the actual backprojection. In our implementation, we thus carry out the backprojection in a rebinned pseudo-parallel ‘wedge’ geometry (Flohr *et al* 2005, Heuscher *et al* 2004).

The rebinning is performed according to the equation

$$g_r(\vartheta(\lambda, \gamma), s_r(\lambda, \gamma), w) = g(\lambda, \gamma, w), \quad (14)$$

with

$$\vartheta(\lambda, \gamma) = \lambda + \frac{\pi}{2} - \gamma, \quad s_r(\lambda, \gamma) = R_0 \sin \gamma, \quad (15)$$

and w remaining untouched. In this rebinned geometry, the boundaries of the TD window are given by

$$w_{\text{top}} = \frac{Dh \pi/2 - \arcsin(s_r/R_0)}{R_0 \sqrt{1 - s_r^2/R_0^2}}, \quad w_{\text{bottom}} = -\frac{Dh \pi/2 + \arcsin(s_r/R_0)}{R_0 \sqrt{1 - s_r^2/R_0^2}}, \quad (16)$$

and the DBP is expressed with an operator, $\mathcal{D}\mathcal{R}\mathcal{P}_r\{\vartheta_a, \vartheta_b, \underline{x}\}$, which designates the DBP of $g_r(\vartheta, s_r, w)$ over the interval $[\vartheta_a, \vartheta_b]$ at \underline{x} .

The link between the wedge definition of the DBP and the expression used in equation (13) is

³In this context, $[a, b]$ means $[b, a]$ when $b < a$.

$$\mathcal{DBP}\{\lambda_a, \lambda_b, \underline{x}\} = \mathcal{DBP}_r\{\vartheta^*(\lambda_a, \underline{x}), \vartheta^*(\lambda_b, \underline{x}), \underline{x}\}, \quad (17)$$

where

$$\vartheta^*(\lambda, \underline{x}) = \lambda + \frac{\pi}{2} - \gamma^*(\lambda, \underline{x}), \quad (18)$$

with $\gamma^*(\lambda, \underline{x})$ being the value of γ that corresponds to the line through $\underline{a}(\lambda)$ and \underline{x} , i.e.,

$$\gamma^*(\lambda, \underline{x}) = \arctan\left(\frac{y \cos(\lambda + \lambda_0) - x \sin(\lambda + \lambda_0)}{R_0 - x \cos(\lambda + \lambda_0) - y \sin(\lambda + \lambda_0)}\right). \quad (19)$$

Therefore, equation (13) can be rewritten in the form

$$(\mathcal{H}f)(\underline{x}, \underline{\omega}(\lambda_M, \underline{x})) = \mathcal{DBP}_r\{\vartheta^*(\lambda_M, \underline{x}), \vartheta_2(\underline{x}), \underline{x}\} - \mathcal{DBP}_r\{\vartheta_1(\underline{x}), \vartheta^*(\lambda_M, \underline{x}), \underline{x}\}, \quad (20)$$

where $\vartheta_1(\underline{x}) = \vartheta^*(\lambda_1(\underline{x}), \underline{x})$ and $\vartheta_2(\underline{x}) = \vartheta^*(\lambda_2(\underline{x}), \underline{x})$ denote the values of ϑ where the projection of \underline{x} just enters (respectively leaves) the TD window in the rebinned geometry.

The precise expression for $\mathcal{DBP}_r\{\vartheta_a, \vartheta_b, \underline{x}\}$ is

$$\mathcal{DBP}_r\{\vartheta_a, \vartheta_b, \underline{x}\} = -\frac{1}{2\pi} \int_{\vartheta_a}^{\vartheta_b} g_d(\vartheta, s_r^*(\vartheta, \underline{x}), w^*(\vartheta, \underline{x})) d\vartheta, \quad (21)$$

with

$$g_d(\vartheta, s_r, w) = \frac{D}{\sqrt{D^2 + w^2}} \frac{\partial}{\partial s_r} g_r(\vartheta, s_r, w), \quad (22)$$

$$s_r^*(\vartheta, \underline{x}) = x \cos(\vartheta + \vartheta_0) + y \sin(\vartheta + \vartheta_0), \quad (23)$$

$$w^*(\vartheta, \underline{x}) = \frac{D(z - z_0 - h(\vartheta - \pi/2 + \arcsin(s_r^*/R_0)))}{y \cos(\vartheta + \vartheta_0) - x \sin(\vartheta + \vartheta_0) + \sqrt{R_0^2 - s_r^{*2}}}, \quad (24)$$

where $\vartheta_0 = \lambda_0 + \pi/2$.

Inserting equation (21) into the right-hand side of equation (20) and combining the resulting terms together into a single weighted backprojection integral offers a more practical expression for implementing equation (20). This expression given below is used in our algorithm as described later:

$$(\mathcal{H}f)(\underline{x}, \underline{\omega}(\lambda_M, \underline{x})) = -\frac{1}{2\pi} \int_{\vartheta_{\min}}^{\vartheta_{\max}} \alpha(\vartheta, \underline{x}) g_d(\vartheta, s_r^*(\vartheta, \underline{x}), w^*(\vartheta, \underline{x})) d\vartheta, \quad (25)$$

where ϑ_{\min} and ϑ_{\max} are the minimum and maximum values among the three angles, $\vartheta_1(\underline{x})$, $\vartheta_2(\underline{x})$ and $\vartheta^*(\lambda_M, \underline{x})$, and where

$$\alpha(\vartheta, \underline{x}) = \text{sign}(\vartheta - \vartheta^*(\lambda_M, \underline{x})) - \frac{1}{2} \text{sign}(\vartheta - \vartheta_1(\underline{x})) - \frac{1}{2} \text{sign}(\vartheta - \vartheta_2(\underline{x})). \quad (26)$$

Note that $\alpha(\vartheta, \underline{x})$ is zero for $\vartheta \notin [\vartheta_{\min}, \vartheta_{\max}]$, so the integration bounds in equation (25) can be extended to the nearest view samples outside $[\vartheta_{\min}, \vartheta_{\max}]$ to simplify the implementation.

3. DBP–HT with redundant data

3.1. Principle

As discussed in the previous section, λ_M can be chosen freely over a wide range of values when implementing equation (20) for a given point \underline{x} . By construction, the selection of λ_M plays a significant role in the range of projections involved in the DBP. In order to illustrate this, figure 2 shows three different choices of λ_M with the corresponding M-lines. The part of the vertex path marked in bold indicates the source positions that contribute to the backprojection terms on the right-hand side of equation (13). The backprojection range spans over the interval $[\lambda_1(\underline{x}), \lambda_2(\underline{x})]$ for $\lambda_1(\underline{x}) \leq \lambda_M \leq \lambda_2(\underline{x})$, whereas it is extended to $[\lambda_M, \lambda_2(\underline{x})]$ if $\lambda_M < \lambda_1(\underline{x})$ and to $[\lambda_1(\underline{x}), \lambda_M]$ if $\lambda_M > \lambda_2(\underline{x})$. The backprojection interval in the case $\lambda_M \in [\lambda_1(\underline{x}), \lambda_2(\underline{x})]$ is precisely the same as in the 1- π methods for TES reconstruction (Katsevich 2002, Noo *et al* 2003, Zou and Pan 2004, Schöndube *et al* 2007a). This interval is exactly covered by the TD window and thus does not allow using the redundant data. On the other hand, the backprojection range extends beyond the boundaries of the TD window when $\lambda_M \notin [\lambda_1(\underline{x}), \lambda_2(\underline{x})]$. Thus, in this case, redundant data are actually used for reconstruction at \underline{x} . This interesting result was first noted in Pack *et al* (2005).

Unfortunately, when evaluating their method, Pack *et al* (2005) found that the amount of noise in the reconstruction at \underline{x} increases for a choice of $\lambda_M \notin [\lambda_1(\underline{x}), \lambda_2(\underline{x})]$, compared to $\lambda_M \in [\lambda_1(\underline{x}), \lambda_2(\underline{x})]$. This somewhat surprising result can be understood by considering the simplified 2D case of $h = 0$ and $w = 0$. In this case, the relation $g_d(\vartheta, s_r, 0) = -g_d(\vartheta + \pi, -s_r, 0)$ holds, which shows that, in contrast to the standard filtered backprojection (FBP) case, there is a minus sign between opposite filtered projections. If the differentiated backprojection is extended over more than a π -interval, the contributions from data that are 180° apart begin to cancel out each other, whereas they are averaged in the standard FBP. Because there is no noise correlation between measurements that are 180° apart, noise contributions add up regardless of whether we perform a DBP or an FBP reconstruction. Therefore, in the case of the DBP, extending the backprojection range yields an increase in the noise level. Pack *et al* (2005) concluded that a similar behavior can be assumed for the 3D case. In summary, when using the DBP–HT reconstruction method on M-lines it is not possible to reduce the noise level in reconstructed images just by naively extending the backprojection range over more than the π -interval.

Nonetheless, Pack *et al* (2005) found that noise can be reduced by averaging the three different reconstructions of $f(\underline{x})$ that correspond to the following choices for λ_M : $\lambda_M = \lambda_i(\underline{x})$, $\lambda_M \in [\lambda_1(\underline{x}), \lambda_2(\underline{x})]$ and $\lambda_M = \lambda_o(\underline{x})$. Unfortunately, this procedure requires that a backprojection on all points along the intersections of three specific M-lines with the support of f must be performed to obtain the final outcome at any given point \underline{x} ; it is thus inefficient and not suitable

for practical use. Specifically, the required number of operations is of order $O(N^5)$, which can be seen as follows. Let $N_x = N_y = N_z = N$ be the number of voxels that is desired in x , y and z . Let $N_M \simeq N$ be the number of samples on one M-line. Let $N_p \simeq N$ be the number of source positions per half-turn of the helix. To reconstruct one voxel value, we need to backproject onto all N_M samples of one M-line through the center of the voxel, so that the inverse HT can subsequently be applied. Moreover, to perform this backprojection, we need about one half helix turn of projections. Thus, the required number of operations is of order $O(N_M \cdot N_p)$ for one voxel value and one M-line. Since the procedure proposed in Pack *et al* (2005) requires three M-lines per voxel, and we wish for $N_x \cdot N_y \cdot N_z = N^3$ voxels, we end up with a number of operations being $O(N_M \cdot N_p \cdot 3 \cdot N^3)$, which is essentially $O(N^5)$.

In the remainder of this paper, the ideas from Pack *et al* (2005), Schöndube *et al* (2007a, 2007b) are combined to obtain a new TES reconstruction algorithm that effectively uses the redundant data while requiring a number of operations of order $O(N^4)$, instead of $O(N^5)$.

3.2. Our method

In contrast to the inefficient voxel-based method outlined in section 3.1, our approach performs a volume-based reconstruction similar to our π -line method presented in Schöndube *et al* (2007a). More precisely, we perform three volumetric reconstructions of the desired volume V , and then compute their average to obtain a result that beneficially involves the redundant data.

Each volumetric reconstruction is obtained by partitioning V into a stack of nutating surfaces of M-lines, performing reconstruction on those surfaces and then interpolating to a Cartesian grid. The partitioning of V is based on a single parameter, denoted by w_{surf} . This parameter specifies a w -coordinate on the detector toward which all M-lines of all surfaces are required to point to. The three partitionings of V that we consider correspond to the following choices for w_{surf} : $w_{\text{surf}} = 0$, $w_{\text{surf}} = w_{\text{min}}$ and $w_{\text{surf}} = w_{\text{max}}$, with w_{min} and w_{max} being the w -coordinate for the first and the last detector row, respectively, as described in section 2.1. The reconstructions of V using these three values of w_{surf} are denoted as $V^*(0)$, $V^*(w_{\text{min}})$ and $V^*(w_{\text{max}})$, and their average as $\bar{V}^* = (V^*(0) + V^*(w_{\text{min}}) + V^*(w_{\text{max}}))/3$. These four volumes differ from V only due to data noise and discretization errors.

In addition to pointing toward $w = w_{\text{surf}}$, the M-lines forming each surface in any given partitioning of V are chosen to be equidistant and parallel to each other when projected onto the (x, y) -plane. There is thus one and only one M-line on each surface that intersects the z -axis. We use the angle that specifies the source position of this M-line to index the surfaces; this angle is called λ_{surf} . Figure 3 depicts the surface of index λ_{surf} for partitioning of V based on a given value of w_{surf} . We denote this surface as $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$ with a semicolon between the two parameters, λ_{surf} and w_{surf} , to emphasize that the first parameter is a variable, whereas the second is a constant for a given partitioning of V .

To specify locations onto $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$, we use two Cartesian coordinates, s and τ , that are defined within the (x, y) -plane, as shown in figure 3. These coordinates are measured along the following orthonormal axes, which rotate with the direction of the M-lines:

$$\underline{e}_s(\lambda_{\text{surf}}) = [-\sin(\lambda_{\text{surf}} + \lambda_0), \cos(\lambda_{\text{surf}} + \lambda_0), 0], \quad (27)$$

$$\underline{e}_\tau(\lambda_{\text{surf}}) = [-\cos(\lambda_{\text{surf}} + \lambda_0), -\sin(\lambda_{\text{surf}} + \lambda_0), 0]. \quad (28)$$

Vector $\underline{e}_\tau(\lambda_{\text{surf}})$ is along the projection of each M-line onto the (x, y) -plane and $\underline{e}_s(\lambda_{\text{surf}})$ is defined such that $\underline{e}_s(\lambda_{\text{surf}}) \times \underline{e}_\tau(\lambda_{\text{surf}}) = \underline{e}_z$ for any λ_{surf} . Coordinate s identifies the M-lines on the surface, whereas τ is the positioning parameter on the M-line at the location s . The meaning of s is that of the signed distance from the origin to the projection of any M-line onto the (x, y) -plane. Location $(s, \tau) = (0, 0)$ corresponds to the intersection of $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$ with the z -axis.

For any given w_{surf} , the variables s , τ and λ_{surf} parameterize together the three-dimensional space within the helix cylinder. The Cartesian coordinates of the point, $\underline{x}(s, \tau, \lambda_{\text{surf}}; w_{\text{surf}})$, specified by these three variables are

$$x = -s \sin(\lambda_{\text{surf}} + \lambda_0) - \tau \cos(\lambda_{\text{surf}} + \lambda_0), \quad (29)$$

$$y = s \cos(\lambda_{\text{surf}} + \lambda_0) - \tau \sin(\lambda_{\text{surf}} + \lambda_0), \quad (30)$$

$$z = z_0 + h \lambda_M^*(\lambda_{\text{surf}}, s) + (\tau + \sqrt{R_0^2 - s^2}) \cdot \frac{w_{\text{surf}}}{D}, \quad (31)$$

where $\lambda_M^*(\lambda_{\text{surf}}, s)$ is the value of λ_M that corresponds to the M-line at coordinate s on the surface $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$, that is

$$\lambda_M^*(\lambda_{\text{surf}}, s) = \lambda_{\text{surf}} + \arcsin(s/R_0). \quad (32)$$

Note that, by construction, the quantity $\vartheta^*(\lambda_M^*(\lambda_{\text{surf}}, s), \underline{x}(s, \tau, \lambda_{\text{surf}}; w_{\text{surf}}))$ based on equation (18) is the same for all (s, τ) -locations on $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$. It is equal to

$$\vartheta_{\text{surf}} = \lambda_{\text{surf}} + \pi/2. \quad (33)$$

Note also that the z -coordinate of $\underline{x}(s, \tau, \lambda_{\text{surf}}; w_{\text{surf}})$, given by equation (31), is a continuous function of λ_{surf} that tends toward $+\infty$ (respectively, $-\infty$) when λ_{surf} tends to $+\infty$ (respectively, $-\infty$). Hence, any point Q inside the FOV can be associated with at least one value of λ_{surf} such that Q belongs to $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$.

To accommodate the geometry of nutating slices, all reconstructions of V are performed with equal, equidistant sampling in x and y ; s and τ are sampled using the same grid as that used for x and y . The number of samples in x , y , s and τ is denoted as N_{lat} and the sampling step as Δ_{lat} , where 'lat' stands for lateral. The main steps to compute $V^*(w_{\text{surf}})$ on a Cartesian grid of $N_{\text{lat}} \times N_{\text{lat}} \times N_z$ voxels of size $(\Delta_{\text{lat}}, \Delta_{\text{lat}}, \Delta z)$ are as follows for any relevant value of w_{surf} , namely $w_{\text{surf}} \in \{0, w_{\text{min}}, w_{\text{max}}\}$:

1. Calculate the range of λ_{surf} needed to fully cover V by the M-line surfaces corresponding to the fixed value of w_{surf} .
2. Create a stack of M-line surfaces, $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$, spaced by $\Delta \lambda_{\text{surf}} = \Delta z/h$ over the calculated range of λ_{surf} .
3. Rebin the data to the wedge geometry using the method described in Flohr *et al* (2005), which is applicable to data collected both without and with FFS.

4. Differentiate the rebinned data in s_r using a simple two-point scheme, i.e., for a generic function $k(s_r)$:

$$k'(s_r + \Delta s_r / 2) \simeq (k(s_r + \Delta s_r) - k(s_r)) / (\Delta s_r). \quad (34)$$

5. Weight the data in w according to equation (22).
6. Compute the DBP onto the stack of M-line surfaces according to equation (25).
7. Apply a finite HT inversion on each M-line.
8. Linearly interpolate from the M-line surfaces to Cartesian coordinates. We first interpolate from $(s, \tau, \lambda_{\text{surf}})$ to an intermediate $(x, y, \lambda_{\text{surf}})$ grid using equations (29) and (30) and then from these intermediate coordinates to (x, y, z) using equation (31).

Step 6 is most demanding in terms of managing computational effort and discretization errors. We provide further details on our implementation of this step in section 3.3. Before going into these details, two observations are, however, in order.

First, the steps outlined above are only applicable when there is no interrupted illumination for all points within the FOV. We show in the appendix that this condition is satisfied as long as the pitch factor, p , and the maximum fan angle, γ_{max} , are selected as in conditions (9).

Interestingly, the condition of no interrupted illumination is closely linked to the requirement that the M-line surfaces have no intersection with each other within the FOV when $|w_{\text{surf}}| = w_{\text{max}}$. In fact, under conditions (9), relation (31) with s and τ defined from equations (29) and (30) establishes a one-to-one relationship between z and λ_{surf} for any (x, y) in the FOV and any $|w_{\text{surf}}| \leq w_{\text{max}}$. This property facilitates the implementation of step 8 and is discussed in the appendix along with the issue of interrupted illumination.

Second, the three targeted reconstructions of V involve different data as they are based on different values of w_{surf} . On one hand, reconstruction $V^*(0)$ involves no data outside the TD window. On the other hand, reconstructions $V^*(w_{\text{min}})$ and $V^*(w_{\text{max}})$ involve together all measured data, with $V^*(w_{\text{min}})$ using the data below and within the TD window, and $V^*(w_{\text{max}})$ using the data within and above the TD window. This is illustrated in figure 4. Following our discussion in section 3.1, the noise level in $V^*(w_{\text{min}})$ and in $V^*(w_{\text{max}})$ may be expected to be higher than in $V^*(0)$, whereas the average \bar{V}^* may be expected to be significantly less noisy than any of these three reconstructions of V , as is demonstrated in the following section.

3.3. Details on the backprojection step

Given that the backprojection step involves many arithmetic operations that do not depend on the value of w_{surf} , it is more efficient to perform the backprojection for the three volumes, $V^*(0)$, $V^*(w_{\text{min}})$ and $V^*(w_{\text{max}})$, altogether. We describe the steps for such a joint backprojection below. These steps compute a numerically stable version of equation (25) using the simple rectangular rule for numerical integration. The numerical stabilization amounts to replacing $\alpha(\vartheta, \underline{x})$ in (25) by

$$\alpha_a(\vartheta, \underline{x}) = \text{sign}_a(\vartheta - \vartheta^*(\lambda_M, \underline{x})) - \frac{1}{2} \text{sign}(\vartheta - \vartheta_1(\underline{x})) - \frac{1}{2} \text{sign}(\vartheta - \vartheta_2(\underline{x})), \quad (35)$$

where $\text{sign}_a(\cdot)$ is a smooth approximation of the signum function, namely,

$$\text{sign}_a(\vartheta) = \begin{cases} -1 & \text{if } \vartheta \leq -\Delta\vartheta \\ 2\frac{\vartheta}{\Delta\vartheta} + \left(\frac{\vartheta}{\Delta\vartheta}\right)^2 & \text{if } -\Delta\vartheta < \vartheta < 0 \\ 2\frac{\vartheta}{\Delta\vartheta} - \left(\frac{\vartheta}{\Delta\vartheta}\right)^2 & \text{if } 0 \leq \vartheta < \Delta\vartheta \\ 1 & \text{if } \vartheta \geq \Delta\vartheta. \end{cases} \quad (36)$$

Note that only the first appearance of the signum function is smoothed; the other two signum functions are left untouched. These two functions are sufficiently randomly sampled over ϑ , x , y and z that no smoothing is needed, whereas the first signum function is used on a fairly deterministic basis that requires numerical stabilization.

Before starting the backprojection, we compute a range of λ_{surf} that allows covering V for all three values of w_{surf} . This range is defined as $[\lambda_{\text{surf,min}}, \lambda_{\text{surf,max}}]$, where $\lambda_{\text{surf,min}}$ is the minimum value of λ_{surf} for which $\mathcal{M}(\lambda_{\text{surf}}, w_{\text{max}})$ just intersects V , and $\lambda_{\text{surf,max}}$ is the maximum value of λ_{surf} for which $\mathcal{M}(\lambda_{\text{surf}}, w_{\text{min}})$ just intersects V . Since the M-line surfaces are sampled with a step $\Delta\lambda_{\text{surf}} = \Delta z/h$, the number of surfaces for each partition of V is then selected as

$$N_{\text{surf}} = \text{ceil} \left(\frac{\lambda_{\text{surf,max}} - \lambda_{\text{surf,min}}}{\Delta\lambda_{\text{surf}}} \right), \quad (37)$$

where $\text{ceil}(\cdot)$ is the operation that rounds any number to the nearest larger integer. By construction, N_{surf} differs from N_z only by some overhead resulting from the obliqueness of the surfaces of M-line relative to the z -axis.

The backprojection is performed over the range $\vartheta \in [\lambda_{\text{surf,min}} + \pi/2, \lambda_{\text{surf,max}} + \pi/2]$ and fills the entries of three arrays of size $N_{\text{lat}} \times N_{\text{lat}} \times N_{\text{surf}}$, where each array corresponds to one of the desired values of w_{surf} . The steps for any filtered projection of angle ϑ that is within the backprojection range are as follows:

1. For every $\lambda_{\text{surf}} \in [\lambda_{\text{surf,min}}, \lambda_{\text{surf,max}}]$:
 - Test if the current projection contributes to samples that are within the FOV on $\mathcal{M}(\lambda_{\text{surf}}, w_{\text{surf}})$ for any $w_{\text{surf}} \in \{0, w_{\text{min}}, w_{\text{max}}\}$. If not, increment the value of λ_{surf} ; otherwise, proceed with the steps below. The test is always (conservatively) positive when $\hat{\vartheta} = \vartheta - \vartheta_{\text{surf}}$ satisfies the following condition, with $\vartheta_{\text{surf}} = \lambda_{\text{surf}} + \pi/2$:

$$|\hat{\vartheta}| < \gamma_{\text{max}} + (N_{\text{rows}} \Delta w) \frac{R_0(1 + \sin \gamma_{\text{max}})}{Dh}. \quad (38)$$

This condition can be used to speed up this first step.

- For every s on the grid perform the following:
 - Determine the range of τ which is inside the FOV for the current value of s .
 - For every τ within that interval:
 - ◆ Compute the value of s_r^* for the current value of (s, τ) according to equation (23).

- ◆ For this value of s_r^* , compute w_{top} and w_{bottom} , the upper and lower boundaries of the TD window, using linear interpolation through tabulated values of equation (16).
- ◆ For each $w_{\text{surf}} \in \{0, w_{\text{min}}, w_{\text{max}}\}$:
 - Compute the value of w^* for the current $(s, \tau, w_{\text{surf}})$ triple according to equations (24) and (31), using again interpolation through tabulated values, this time of the functions $\arcsin(\cdot)$ and $\sqrt{\cdot}$.
 - Compute $\alpha_a(\vartheta, \underline{x})$ of equation (35) using the formula

$$\alpha_a(\vartheta, \underline{x}) = \begin{cases} (\text{sign}_a(\widehat{\vartheta}) - 1) & \text{if } w^* < w_{\text{bottom}} \\ (\text{sign}_a(\widehat{\vartheta})) & \text{if } w_{\text{bottom}} < w^* < w_{\text{top}} \\ (\text{sign}_a(\widehat{\vartheta}) + 1) & \text{if } w^* > w_{\text{top}}. \end{cases}$$

(39)

- Compute $\alpha_a(\vartheta, \underline{x}) \cdot g_d(\vartheta, s_r^*, w^*)$, with bilinear interpolation as the backprojection contribution to the location \underline{x} associated with the current values of w_{surf} , τ , s and λ_{surf} .

The required number of operations of this algorithm is of order $O(N^4)$. Let N be the number of voxels that is desired in x , y and z . Thus, $N_{\text{lat}} = N$ and $N_z = N$. Let $N_p = N$ be the number of source positions per half-turn of the helix. To reconstruct one M-line surface, we need to backproject about one half helix turn of projections onto $N_{\text{lat}} \cdot N_{\text{lat}} = N^2$ locations. This means that the required number of operations is of order $O(N^2 \cdot N_p)$. The procedure outlined above requires N_{surf} M-line surfaces per volume, and we need three volumes for the combined reconstruction, thus we end up with a number of operations being $O(3 \cdot N_{\text{surf}} \cdot N^2 \cdot N_p)$, which is essentially $O(N^4)$, since $N_{\text{surf}} \approx N_z = N$.

4. Algorithm validation

We present here preliminary results on the performance of our algorithm. These results were obtained from computer-simulated data defined using parameters that are representative of state-of-the-art commercial CT scanners; see table 1. Two configurations were considered for the data simulation, namely, without and with a lateral FFS. Moreover, the effect of the finite size of the focal spot and of the detector elements was consistently included, using 3×3 detector-lets and 2×2 source-lets at each source position, with exponential averaging of all line integrals connecting a source-let to a detector-let.

Three experiments were performed. In the first experiment, we used the FORBILD head phantom to visually assess robustness in terms of CB artifacts, discretization errors and effective handling of the lateral FFS feature. Figure 5 displays the outcome of this experiment for the slice $z = 0$ cm, as obtained on a grid of 401×401 square pixels of side 0.75 mm. The top row of this figure shows the results of our algorithm without (left) and with (right) lateral FFS, whereas the bottom row shows the ground truth and the difference image between the two images of the first row. These results are complemented by figure 6, where the profile through the ear at $y = 0$ cm is given for the reconstructions without and with FFS, and for the ground truth. It can be observed that the new algorithm yields accurate results, and that the FFS feature is effectively accounted for by the algorithm; in that it yields a clear improvement in resolution.

The second experiment aimed at demonstrating the effective use of the data outside the TD window, and at assessing the magnitude of the contribution of these data to the reconstruction. To achieve this goal, we performed two reconstructions using the FORBILD head phantom data of the first experiment. One reconstruction was obtained with a binary weight applied to the data that set all values inside the TD window to zero and left the other values unchanged. The other reconstruction was obtained with all detector values outside the TD window being set to zero. Naturally, none of these two reconstructions is theoretically exact, but their sum is due to the linearity of all operations in our algorithm. This data splitting technique allows singling out the impact of the data outside the TD window. The two reconstructions and their sum are displayed in figure 7. It can be observed that the data outside the TD window significantly impact the whole reconstruction process.

The third and last experiment aimed at verifying that noise performance behaves as expected. For this experiment, we used the FORBILD thorax phantom with addition of Poisson noise corresponding to an emission of 150 000 photons per ray. Figure 8 displays the reconstructions of slice $z = 2.18$ cm through each of the three sub-volumes, $V^*(0)$, $V^*(w_{\min})$ and $V^*(w_{\max})$, and also through the combined volume \bar{V}^* . These reconstructions were performed on a grid of 512×512 square pixels of side 0.9mm. We also included the difference image between the reconstructions of $V^*(0)$ and \bar{V}^* from noise-free data. This image shows that the difference in resolution between $V^*(0)$ and \bar{V}^* is very small, as the edges in the phantom are hardly visible, whereas the noise level is significantly lower in \bar{V}^* . Quantitative measurements of the standard deviation within seven regions of interest, labeled as A through G, are given in table 2 and support the visual assessment.

5. Discussion and conclusion

We have presented a new algorithm for helical CB CT that allows TES reconstruction while beneficially using all measured data. This algorithm was designed as an extension of the π -line DBP-HT algorithm presented in Schöndube *et al* (2007a) to the more general case of M-lines. The noise improvement that this algorithm offers is achieved using a principle that was first described in Pack *et al* (2005). Specifically, three different reconstructions of the VOI based

on data sets that are only partially overlapping are averaged together to obtain a noise-improved image. Each individual reconstruction is obtained using a differentiated backprojection onto a specific type of M-lines. To ensure the usage of all redundant data on the detector, our proposed method uses one set of M-lines directed to the upper end of the detector, one set directed to the lower end of the detector and one set directed to the central detector row. Unlike the technique presented in Pack *et al* (2005) our algorithm performs the reconstruction volume-wise, instead of voxel-wise; this allows a significant decrease in computational effort, which we have shown to be of the order of the cubic root of the number of voxels.

Our algorithm only applies when there is no interrupted illumination for all points within the FOV, and we have proven that conditions (9) guarantee this to be the case. We did not discuss so far the optimality of these conditions. Regarding this issue, we have observed that the M-line surfaces defined with $|w_{\text{surf}}| = w_{\text{max}}$ intersect each other non-tangentially within the FOV when p is decreased below p_{min} . Such intersections appear first near the edge of the FOV, and can be shown to imply the appearance of interrupted illumination. For this reason, we believe that conditions (9) are optimal.

Conditions (9) are not very restrictive for a small FOV, such as that corresponding to a head scan. But they are very stringent for thorax imaging, especially with wide patients. We are actively investigating new methodologies allowing the handling of interrupted illumination, so that lower pitch values may be considered for an arbitrary FOV size.

Preliminary results from computer-simulated data were provided to show effective use of redundant data and also of the angular FFS. Further quantification of gains in image quality, especially for real data, is important. Noise comparison with other approximate and TES algorithms is also of high interest. However, these two topics are beyond the scope of this paper, particularly because a careful resolution analysis needs to be performed at the same time. We will report on such evaluations in the near future.

An important issue when evaluating noise is whether uniform weighting of independent M-line results is optimal or not. Also, reconstruction using more than three sets of M-lines may be preferred to mitigate patient-motion artifacts along with noise. An early investigation of these issues was reported in Köhler *et al* (2008) at the 2008 IEEE Medical Imaging Conference, and yielded very encouraging results regarding patient-motion handling. No algorithm details were provided in Köhler *et al* (2008), but it is likely that a methodology that is very similar to ours was used. A first description of our algorithm was also given at the 2008 IEEE Medical Imaging Conference (Schöndube *et al* 2008).

Acknowledgments

This work was partially supported by NIH grant R01 EB007236 and by Siemens Healthcare. Its contents are solely the responsibility of the authors.

Appendix

In this appendix, we discuss how conditions (9) guarantee that there is no interrupted illumination for the points within the FOV. Our proof involves a close examination of intersections between the surfaces of M-lines defined with $|w_{\text{surf}}| = w_{\text{max}}$. More precisely, we demonstrate that: (i) under conditions (9) and the assumption that $|w_{\text{surf}}| \leq w_{\text{max}}$, the surfaces $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$ do not intersect each other within the FOV when varying λ_{surf} , (ii) there is no interrupted illumination when the surfaces $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$ do not intersect each other within the FOV for $|w_{\text{surf}}| = w_{\text{max}}$. Taken together, these two statements prove that there is no interrupted illumination under conditions (9).

To best understand our argument, recall that the condition $p < p_{\max}$ is required for any TES reconstruction, and is thus independent from our algorithm. Recall also that the object is always within the FOV, so the CB projection of a point that is within the FOV can only leave the detector through its top and bottom rows. Last, note that the rebinning step of equation (14) does not change the data properties with respect to interrupted illumination: if there is no interrupted illumination in the initial data geometry, there is also none after rebinning, and conversely.

To prove statement (i) above, we rewrite equation (31) into an expression that yields the z -coordinate of the point (x, y, z) that belongs to $\mathcal{M}(\lambda_{\text{surf}}, w_{\text{surf}})$ at given x and y :

$$z(x, y, \lambda_{\text{surf}}; w_{\text{surf}}) = z_0 + h\lambda_{\text{surf}} + h \arcsin\left(\frac{s(x, y, \lambda_{\text{surf}})}{R_0}\right) + \frac{w_{\text{surf}}}{D} (\tau(x, y, \lambda_{\text{surf}}) + \sqrt{R_0^2 - s(x, y, \lambda_{\text{surf}})^2}) \quad (\text{A.1})$$

with

$$s(x, y, \lambda_{\text{surf}}) = -x \sin(\lambda_{\text{surf}} + \lambda_0) + y \cos(\lambda_{\text{surf}} + \lambda_0), \quad (\text{A.2})$$

$$\tau(x, y, \lambda_{\text{surf}}) = -x \cos(\lambda_{\text{surf}} + \lambda_0) - y \sin(\lambda_{\text{surf}} + \lambda_0). \quad (\text{A.3})$$

Then, we examine the behavior of $\partial z / \partial \lambda_{\text{surf}}$ for any (x, y) within the FOV. Straightforward calculations yield

$$\frac{\partial z}{\partial \lambda_{\text{surf}}} = \left(h - \frac{w_{\text{surf}} s(x, y, \lambda_{\text{surf}})}{D} \right) \left(1 + \frac{\tau(x, y, \lambda_{\text{surf}})}{\sqrt{R_0^2 - s(x, y, \lambda_{\text{surf}})^2}} \right) \quad (\text{A.4})$$

$$= \left(h + \frac{w_{\text{surf}}}{D} \cdot r \sin(\lambda_{\text{surf}} - \varphi) \right) \left(1 - \frac{r \cos(\lambda_{\text{surf}} - \varphi)}{\sqrt{R_0^2 - r^2 + r^2 \cos^2(\lambda_{\text{surf}} - \varphi)}} \right) \quad (\text{A.5})$$

$$= \left(h + \frac{w_{\text{surf}}}{D} \cdot r \sin(\lambda_{\text{surf}} - \varphi) \right) k(\cos(\lambda_{\text{surf}} - \varphi)), \quad (\text{A.6})$$

where $r \geq 0$ and $\phi \in [0, 2\pi)$ are the polar coordinates of x and y , and where

$k(t) = 1 - rt / \sqrt{R_0^2 - r^2 + r^2 t^2}$. It turns out that $k(t) > 0$ for all t when $r < R_0$ because

- $k'(t) = -r(R_0^2 - r^2) / (R_0^2 - r^2 + r^2 t^2)^{3/2} < 0$ for any t if $r < R_0$,
- $\lim_{t \rightarrow -\infty} k(t) = 2, \lim_{t \rightarrow +\infty} k(t) = 0$.

Therefore, $\partial z / \partial \lambda_{\text{surf}} > 0$ whenever $r < R_0$ and $|w_{\text{surf}}| r < hD$. However, under conditions (9), we have $p_{\min} < p$, which can easily be rewritten in the form $w_{\max} R_0 \sin \gamma_{\max} < hD$ from the definition of p , w_{\max} and p_{\min} . Given that $R_0 \sin \gamma_{\max}$ is the radius of the FOV, we can conclude at this stage that $\partial z / \partial \lambda_{\text{surf}} > 0$ for any (x, y) within the FOV. This proves statement (i).

The proof of statement (ii) goes as follows. If there was interrupted illumination, there would exist a point within the FOV that is projected at least twice onto $w = w_{\max}$ (or $w = w_{\min}$). But any point that is on the M-line surface $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$ sits on an M-line that represents the course of a ray projecting it onto the detector row $w = w_{\text{surf}}$ with the direction $\vartheta = \lambda_{\text{surf}} + \pi/2$ in the rebinned geometry. Thus, if a point is projected onto the detector row $\hat{w} = w_{\max}$ (or $\hat{w} = w_{\min}$) twice, say under the rebinned projection angles ϑ_1 and ϑ_2 , it must also be located on two M-line surfaces, namely $\mathcal{M}(\vartheta_1 - \pi/2; \hat{w})$ and $\mathcal{M}(\vartheta_2 - \pi/2; \hat{w})$. This is only possible if those two M-line surfaces intersect. Hence, there is no interrupted illumination if the surfaces $\mathcal{M}(\lambda_{\text{surf}}; w_{\text{surf}})$ do not intersect each other within the FOV for $|w_{\text{surf}}| = w_{\max}$.

References

- Bontus C, Köhler T, Proksa R. A quasixact reconstruction algorithm for helical CT using a 3-pi acquisition. *Med. Phys* 2003;30:2493–2502. [PubMed: 14528971]
- Bontus C, Köhler T, Proksa R. EnPiT: filtered back-projection algorithm for helical CT using an n-pi acquisition. *IEEE Trans. Med. Imaging* 2005;24:977–986. [PubMed: 16092330]
- Danielsson, PE.; Edholm, P.; Eriksson, J.; Magnusson Seger, M. Towards exact reconstruction for helical cone-beam scanning of long objects. a new detector arrangement and a new completeness condition. In: Townsend, DW.; Kinahan, PE., editors. *Proc. 1997 Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*; Pittsburgh, PA. 1997. p. 141-144.
- Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. *J. Opt. Soc. Am. A* 1984;1:612–619.
- Flohr T, Stierstorfer K, Bruder H, Simon J, Polacin A, Schaller S. Image reconstruction and image quality evaluation for a 16-slice CT scanner. *Med. Phys* 2003;30:832–845. [PubMed: 12772991]
- Flohr TG, Stierstorfer K, Ulzheimer S, Bruder H, Primak AN, McCollough CH. Image reconstruction and image quality evaluation for a 64-slice CT scanner with z-flying focal spot. *Med. Phys* 2005;32:2536–2547. [PubMed: 16193784]
- Heuscher D, Brown K, Noo F. Redundant data and exact helical cone-beam reconstruction. *Phys. Med. Biol* 2004;49:2219–2238. [PubMed: 15248574]
- Katsevich A. On two versions of a 3π algorithm for spiral CT. *Phys. Med. Biol* 2004;49:2129–2143. [PubMed: 15248568]
- Katsevich A. 3PI algorithms for helical computer tomography. *Adv. Appl. Math* 2006;36:213–250.
- Katsevich, A.; Zamyatin, AA.; Silver, MD. Optimized reconstruction algorithm for helical CT with fractional pitch between 1PI and 3PI. In: Beekman, F.; Kachelrieß, M., editors. *Proc. 2007 Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*; Lindau, Germany. 2007. p. 333-336.
- Katsevich AI. Analysis of an exact inversion algorithm for spiral cone-beam CT. *Phys. Med. Biol* 2002;47:2583–2597. [PubMed: 12200926]
- Köhler T, Bontus C, Koken P. The radon-split method for helical cone-beam CT and its application to nongated reconstruction. *IEEE Trans. Med. Imaging* 2006;25:882–897. [PubMed: 16827489]
- Köhler, T.; Bontus, C.; Proksa, R. BPF reconstruction for helical CT using all data; *IEEE Nuclear Science Symp. Conf. Record NSS'08*; 2008. p. 4154-4156.
- Kudo H, Rodet T, Noo F, Defrise M. Exact and approximate algorithms for helical cone-beam CT. *Phys. Med. Biol* 2004;49:2913–2931. [PubMed: 15285256]
- Noo F, Clackdoyle R, Pack J. A two-step Hilbert transform method for 2D image reconstruction. *Phys. Med. Biol* 2004;49:3903–3923. [PubMed: 15470913]
- Noo F, Pack J, Heuscher D. Exact helical reconstruction using native cone-beam geometries. *Phys. Med. Biol* 2003;48:3787–3818. [PubMed: 14703159]
- Pack J, Noo F, Clackdoyle R. Cone-beam reconstruction using the backprojection of locally filtered projections. *IEEE Trans. Med. Imaging* 2005;24:70–85. [PubMed: 15638187]
- Proksa R, Köhler T, Grass M, Timmer J. The n-PI-method for helical cone-beam CT. *IEEE Trans. Med. Imaging* 2000;19:848–863. [PubMed: 11127600]

- Schöndube, H.; Stierstorfer, K.; Dennerlein, F.; Noo, F. Comparative evaluation of two analytical methods for helical cone-beam tomography; IEEE Nuclear Science Symp. Conf. Record NSS'07; 2007b. p. 4467-4471.
- Schöndube, H.; Stierstorfer, K.; Dennerlein, F.; White, TA.; Noo, F. Towards an efficient two-step Hilbert algorithm for helical cone-beam CT. In: Beekman, F.; Kachelrieß, M., editors. Proc. 2007 Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine; Lindau, Germany. 2007a. p. 120-123.
- Schöndube, H.; Stierstorfer, K.; Noo, F. Accurate helical CT reconstruction with redundant data using nutating slices; IEEE Nuclear Science Symp. Conf. Record NSS'08; 2008. p. 5158-5160.
- Shechter G, Köhler Th, Altman A, Proksa R. The frequency split method for helical cone-beam reconstruction. *Med. Phys* 2004;31:2230–2236. [PubMed: 15377089]
- Söhngen H. Die Lösungen der Integralgleichung $g(x)=12\pi\int_{-\infty}^{\infty} f(\xi)x-\xi d\xi$ und deren Anwendung in der Tragflügeltheorie. *Math. Z* 1937;45:245–264.
- Stierstorfer K, Flohr T, Bruder H. Segmented multiple plane reconstruction: a novel approximate reconstruction scheme for multi-slice spiral CT. *Phys. Med. Biol* 2002;47:2571–2581. [PubMed: 12200925]
- Stierstorfer K, Rauscher A, Boese J, Bruder H, Schaller S, Flohr T. Weighted—a simple approximate 3D FBP algorithm for multislice spiral CT with good dose usage for arbitrary pitch. *Phys. Med. Biol* 2004;49:2209–2218. [PubMed: 15248573]
- Tam KC, Samarasekera S, Sauer F. Exact cone-beam CT with a spiral scan. *Phys. Med. Biol* 1998;43:1015–1024. [PubMed: 9572524]
- Tang X, Hsieh J. Handling data redundancy in helical cone beam reconstruction with a cone-angle-based window function and its asymptotic approximation. *Med. Phys* 2007;34:1989–1998. [PubMed: 17654902]
- Tuy HK. An inversion formula for cone-beam reconstruction. *SIAM J. Appl. Math* 1983;43:546–552.
- Ye Y, Zhao S, Yu H, Wang G. A general exact reconstruction for cone-beam CT via backprojection-filtration. *IEEE Trans. Med. Imaging* 2005;24:1190–1198. [PubMed: 16156356]
- Yu L, Xia D, Zou Y, Sidky E, Bian J, Pan X. A rebinned backprojection-filtration algorithm for image reconstruction in helical cone-beam CT. *Phys. Med. Biol* 2007;52:5497–5508. [PubMed: 17804878]
- Zamyatin, AA.; Katsevich, A.; Silver, MD.; Nakanishi, S. Helical CT reconstruction with large cone angle; IEEE Nuclear Science Symp. Conf. Record NSS'06; 2006. p. 2264-2267.
- Zou Y, Pan X. Exact image reconstruction on PI-line from minimum data in helical cone-beam CT. *Phys. Med. Biol* 2004;49:941–959. [PubMed: 15104318]

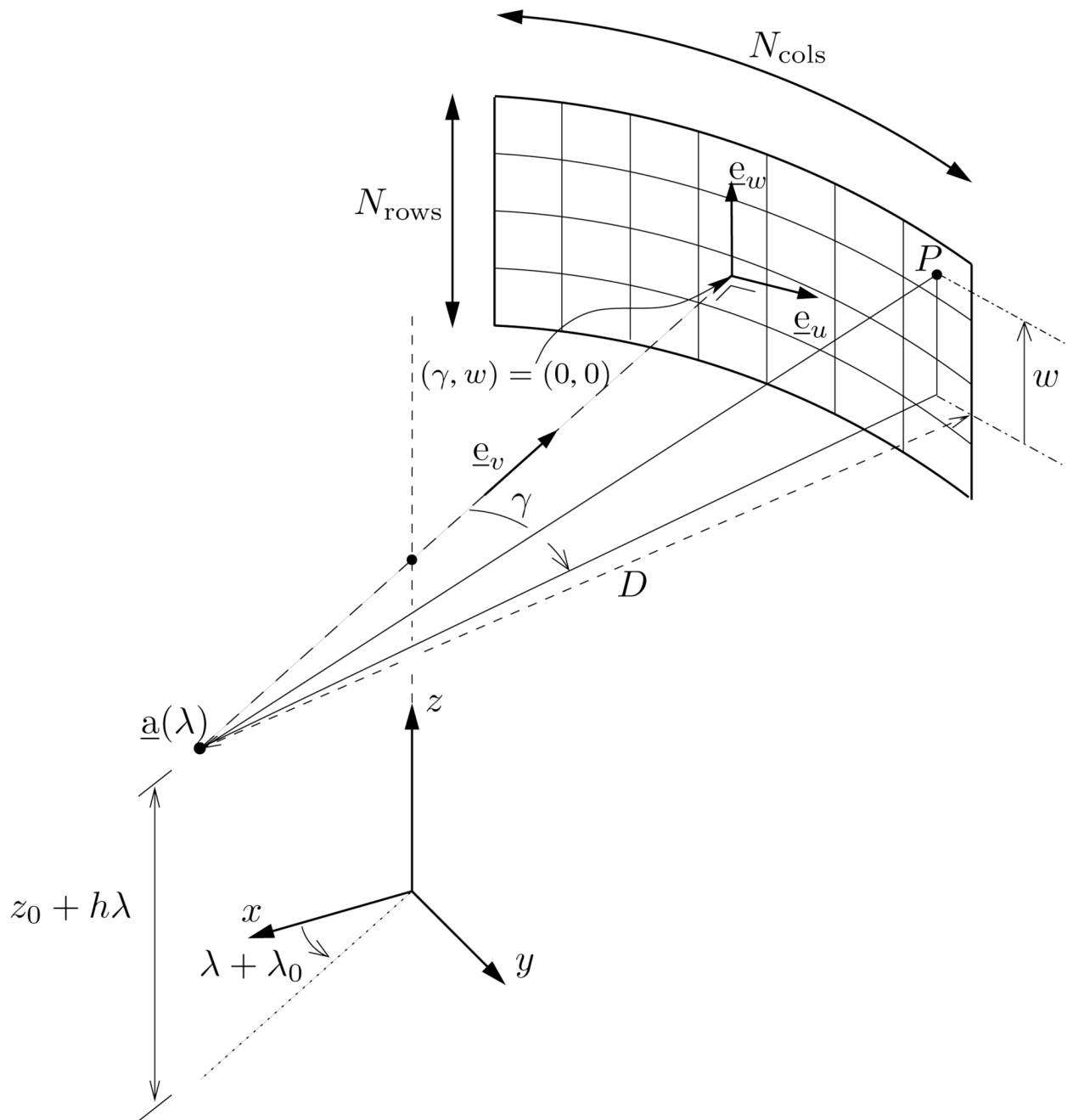


Figure 1. Data acquisition geometry. Our algorithm is designed for the geometry of third generation CT scanners with a multi-row detector.

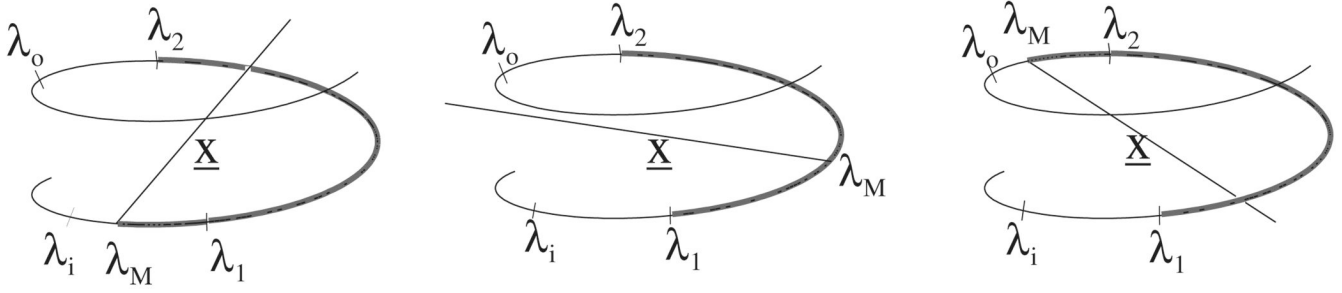


Figure 2.

M-lines and backprojection range. Reconstruction at \underline{x} can be achieved by computing and then inverting the HT on any M-line through \underline{x} , but this procedure requires a different range of projections depending on the position of λ_M (the angle defining the M-line) relative to λ_1 and λ_2 (the angles defining the π -line through \underline{x}). Three cases are possible: λ_M can be located before λ_1 (left diagram), after λ_2 (right diagram) or between λ_1 and λ_2 (middle diagram). In each case, the highlighted portion of the helix depicts the required backprojection range. Note also that the selection of λ_M is restricted to the interval $[\lambda_i, \lambda_o]$ over which no interrupted illumination occurs.

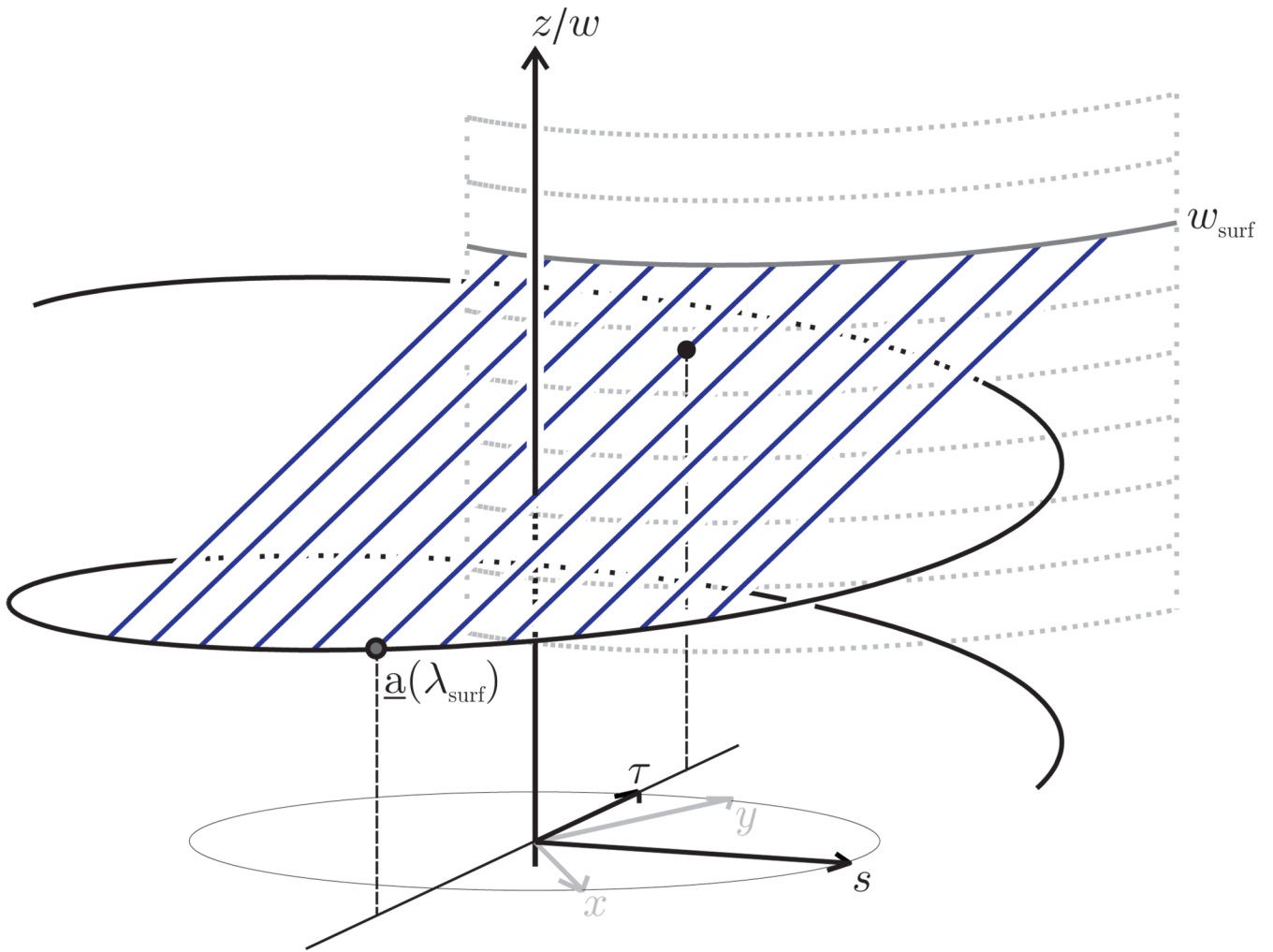


Figure 3.

Depiction of the M-line surface $\mathcal{M}(\lambda_{\text{surf}}, w_{\text{surf}})$. Each M-line on $\mathcal{M}(\lambda_{\text{surf}}, w_{\text{surf}})$ hits the detector at $w = w_{\text{surf}}$, i.e., on a fixed row of the virtual detector onto which the data are mapped by the wedge rebinning. By varying λ_{surf} from $-\infty$ to $+\infty$ at fixed w_{surf} , we get a stack of nutating slices that covers the entire FOV.

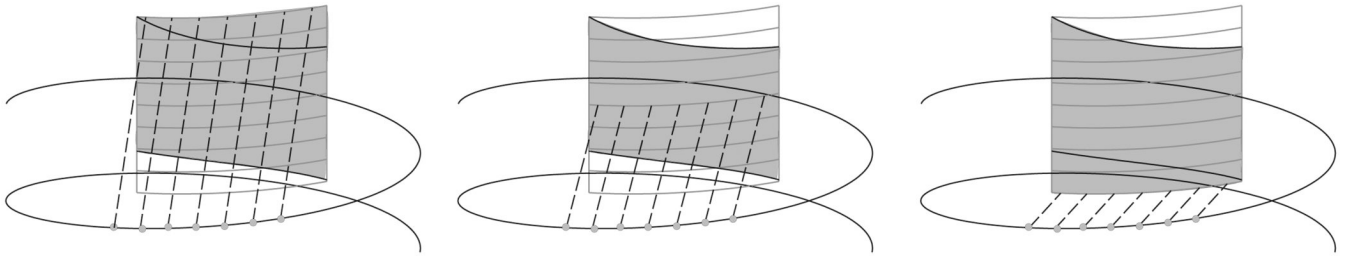


Figure 4. Illustration of the three different kinds of M-line surfaces used in our approach. From left to right: $w_{\text{surf}} = w_{\text{max}}$, $w_{\text{surf}} = 0$ and $w_{\text{surf}} = w_{\text{min}}$. The curved black lines on the rebinned detector denote the boundaries of the TD window; the shaded detector area shows the amount of data needed for reconstruction on each surface.

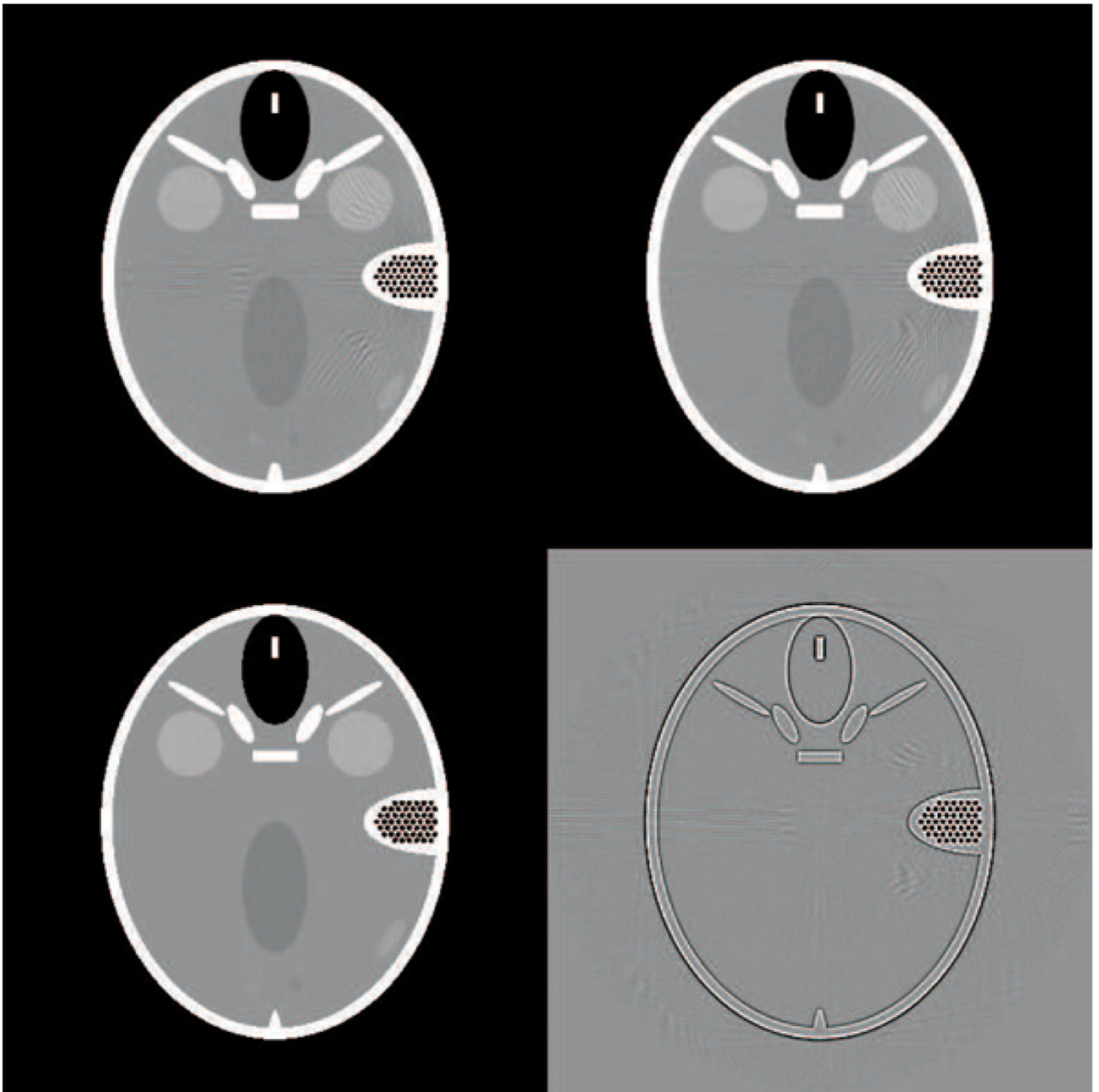


Figure 5. Reconstructions of the FORBILD head phantom (with right ear only). Upper row: DBP-HT reconstructions from data simulated without (left) and with (right) angular FFS. Bottom row: ground truth (left) and difference image between the two reconstructions in the top row. The grayscale window for the ground truth and the reconstructions is $[0, 100]$ HU. The window for the difference image is $[-50, 50]$ HU.

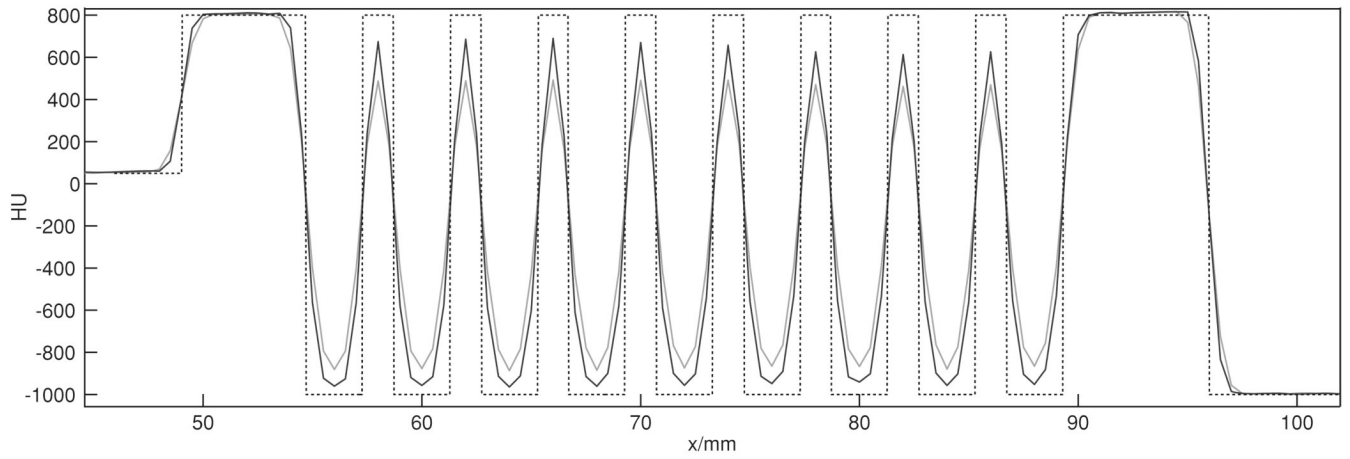


Figure 6.

Profile at $y = 0$ cm through the ear of the reconstructions shown in figure 5; dotted line: ground truth, solid black line: DBP-HT from data simulated with angular FFS, solid gray line: DBP-HT from non-FFS data.

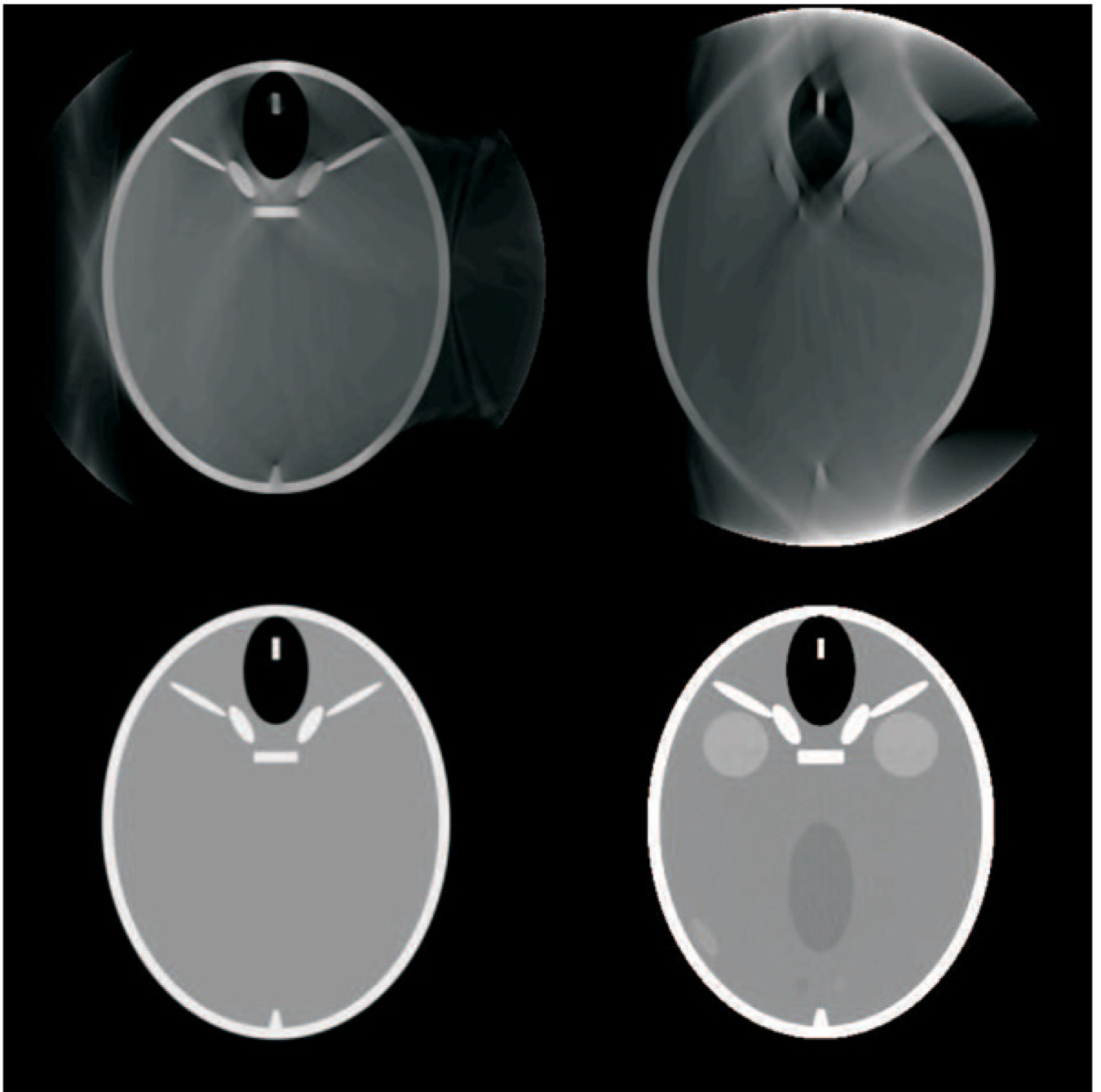


Figure 7. Impact of redundant data in our reconstruction process. The top row shows the reconstruction of the FORBILD head phantom with no ears, as obtained when setting to zero either all data outside the TD window (left) or all data inside the TD window (right). The grayscale is $[-1000, 1000]$ HU. Bottom row: sum of the two images in the top row, displayed with two different grayscale: (left) $[-1000, 1000]$ HU, (right) $[0, 100]$ HU.

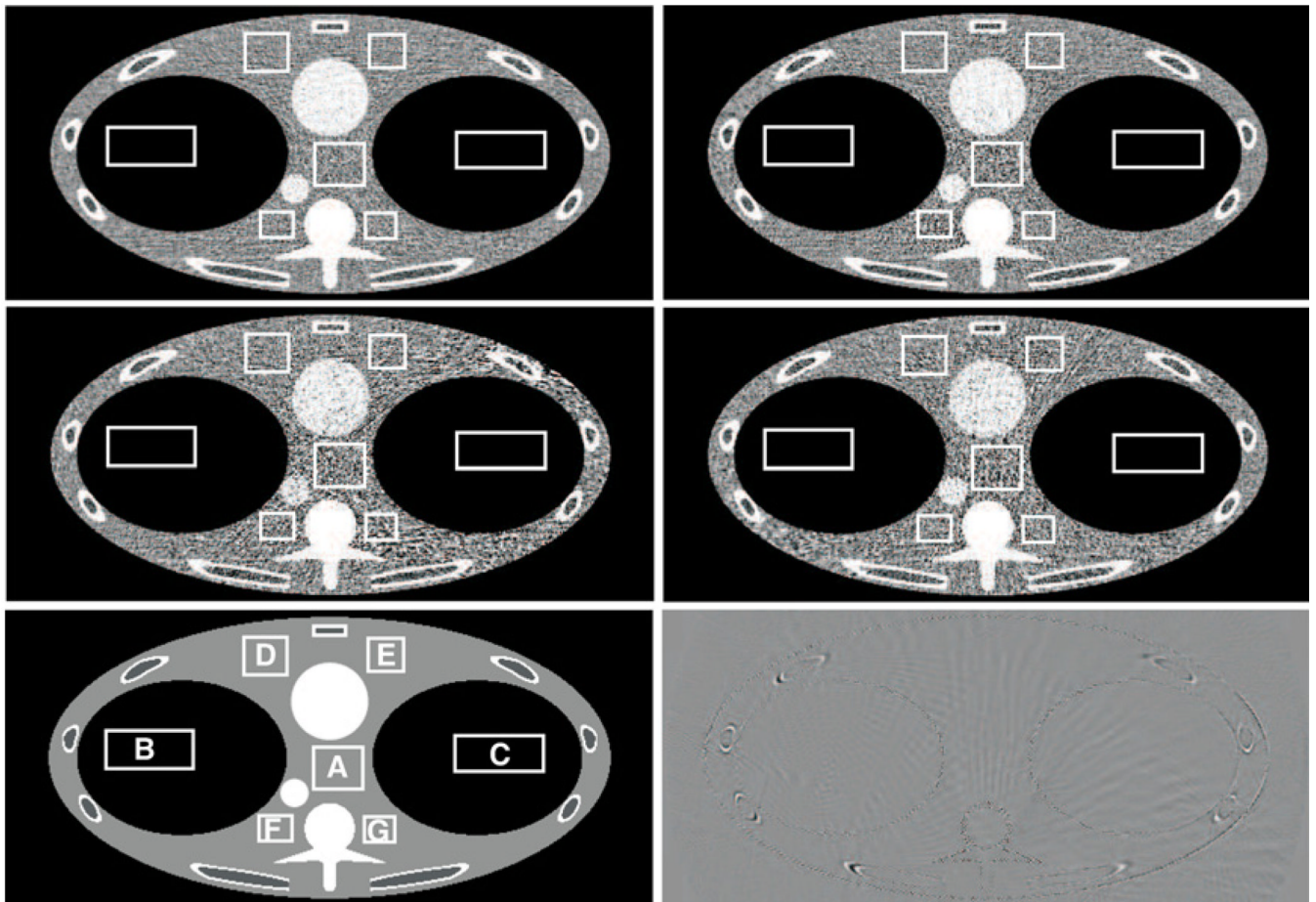


Figure 8.

Reconstruction of the FORBILD thorax phantom with Poisson noise added to the data, assuming an emission of 150 000 photons per ray. The standard deviation measurements reported in table 2 were evaluated in the rectangular areas. Top left: combined reconstruction \bar{V}^* . Top right: reconstruction $V^*(0)$, which is based only on the M-line surfaces defined with $w_{\text{surf}} = 0$. Middle left: reconstruction $V^*(w_{\text{max}})$, which is based only on the surfaces defined with $w_{\text{surf}} = w_{\text{max}}$. Middle right: reconstruction $V^*(w_{\text{min}})$, which is based only on the surfaces defined with $w_{\text{surf}} = w_{\text{min}}$. Bottom left: ground truth. Bottom right: difference image between \bar{V}^* and $V^*(0)$. The following grayscale window was used in all cases: $[-50, 50]$ HU.

Table 1

Data-simulation parameters.

No of detector rows	64
No of detector columns	736
Detector pixel height	0.10947 cm
Detector pixel width	0.12858 cm
Source-to-detector distance (D)	108.56 cm
Height of the focal spot	0.135 cm
Width of the focal spot	0.12 cm
Anode angle	7.0°
Radius of the helix (R_0)	59.5 cm
No of projections per turn (no FFS)	1152
No of projections per turn (with FFS)	2304
Pitch factor	1.35

Table 2

Standard deviation in HU within seven different regions of interest of the reconstructed images displayed in figure 8.

	$V^*(w_{\max})$	$V^*(w_{\min})$	$V^*(0)$	$\frac{-}{V^*}$
A	32.74	32.36	25.56	18.33
B	12.29	13.57	11.94	8.56
C	17.52	14.61	11.34	9.02
D	20.45	21.00	14.44	13.62
E	24.55	25.25	17.73	14.76
F	29.84	25.81	28.29	18.34
G	32.81	27.57	21.32	17.41