Published in final edited form as:
Int J Comput Assist Radiol Surg. 2012 January ; 7(1): 73–86. doi:10.1007/s11548-011-0607-z.

Accurate image reconstruction using real C-arm data from a Circle-plus-arc trajectory

Stefan Hoppe,
University of Erlangen-Nuremberg, Chair of Pattern Recognition, Martensstr. 3, 91058 Erlangen, Germany

Joachim Hornegger,
University of Erlangen-Nuremberg, Chair of Pattern Recognition, Martensstr. 3, 91058 Erlangen, Germany

Frank Dennerlein,
Siemens AG, Health care Sector, H IM AX IN IC, Siemensstr. 1, 91301 Forchheim, Germany

Günter Lauritsch, and
Siemens AG, Health care Sector, H IM AX IN IC, Siemensstr. 1, 91301 Forchheim, Germany

Frédéric Noo
Department of Radiology, University of Utah, 729 Arapeen Drive, Salt Lake City, UT 84108, USA

Stefan Hoppe: hoppe.stefan@online.de; Joachim Hornegger: joachim.hornegger@informatik.uni-erlangen.de; Frank Dennerlein: frank.dennerlein@siemens.com; Günter Lauritsch: guenter.lauritsch@siemens.com; Frédéric Noo: noo@ucair.med.utah.edu

Abstract

Objective—Developing an efficient tool for accurate three-dimensional imaging from projections measured with C-arm systems.

Material and methods—A circle-plus-arc trajectory, which is complete and thus amenable to accurate reconstruction, is used. This trajectory is particularly attractive as its implementation does not require moving the patient. For reconstruction, we use the "M-line method", which allows processing the data in the efficient filtered backprojection mode. This method also offers the advantage of not requiring an ideal data acquisition geometry, i.e., the *M*-line algorithm can account for known deviations in the scanning geometry, which is important given that sizeable deviations are generally encountered in C-arm imaging.

Results—A robust implementation scheme of the "M-line method" that applies straightforwardly to real C-arm data is presented. In particular, a numerically stable technique to compute the view-dependent derivative with respect to the source trajectory parameter is applied, and an efficient way to compute the π -line backprojection intervals via a polygonal weighting mask is presented. Projection data of an anthropomorphic thorax phantom were acquired on a medical C-arm scanner and used to demonstrate the benefit of using a complete data acquisition geometry with an accurate reconstruction algorithm versus using a state-of-the-art implementation of the conventional Feldkamp algorithm with a circular short scan of cone-beam data. A significant image quality improvement based on visual assessment is shown in terms of cone-beam artifacts.

Correspondence to: Frank Dennerlein, frank.dennerlein@siemens.com.

Disclaimer
The concepts and information presented in this paper are based on research and are not commercially available.

Keywords

C-arm; Cone-beam reconstruction; Computed tomography

Introduction

Angiographic C-arm systems are well established in clinical routine for guiding intravascular interventions in radiology and cardiology. They are equipped with a movable source-detector assembly (see Fig. 1), and they offer a large degree of freedom in angulations to allow for optimal viewing angles. It turns out that this flexibility can also be used to collect a set of 2D projection images along a specific trajectory allowing reconstruction of tomographic 3D images. Moreover, the recent advent of flat-panel detectors has improved the quality of X-ray images in such a way that CT-like images are within reach. These features are important and timely because the complexity of interventional procedures has been steadily increasing over the last decade, with a higher and higher demand in 3D imaging capability. Today, 3D imaging has become an essential tool for spine surgery (vertebroplasty), accurate delivery of chemo-embolization of tumors, stenting of stenoses, and treatment of cerebral aneurysms. These applications and the development of future procedures demand high resolution and detectability of low-contrast details pushing for continuous improvements in imaging technology.

For image reconstruction, the C-arm's enhanced flexibility induces some algorithmic requirements. In particular, the C-arm is not able to move the source-detector assembly along an ideal trajectory due to mechanical inaccuracies and physical phenomena. For instance, gravity causes the C-arm to flex or expand during its motion. The non-ideality in the data acquisition process has to be considered during image reconstruction to avoid reconstruction artifacts. State-of-the-art implementations of the Feldkamp (FDK) algorithm often consider geometrical deviations only in the backprojection step, while the remaining filtering steps are implemented as if the geometry was ideal. Accurate reconstruction algorithms, on the other hand, have to consider the precise data acquisition geometry in each processing step.

The goal of this work is the robust implementation of an accurate filtered backprojection (FBP) algorithm such that it is able to process real C-arm data. For data acquisition, we use the circle-plus-arc trajectory. It is complete and especially well suited for C-arm systems since it can be performed purely by rotating the C-arm around the patient without the need to move the patient table. For image reconstruction, we adapt the M-line algorithm originally presented by Pack and Noo [1] to the requirements imposed by our C-arm system. The M-line algorithm has been our first choice for this task because it does not presume an ideal description of the data acquisition geometry. For instance, the filtering step does not require special care, even if the geometry is non-ideal. Other algorithms, such as the algorithm of Katsevich [2], require a different determination of the filtering lines, which may impose additional strategies for a robust implementation, as, for example, a trajectory fitting; see Dennerlein et al. [3]. Note that other implementations of C-arm-based cone-beam reconstruction techniques do exist; see, for example, references [4–6],

The paper is organized as follows. Section "C-arm imaging and geometry" introduces to modern C-arm systems, with special focus on the Artis zee ceiling-mounted C-arm system (Siemens AG, Healthcare Sector). Section "Image reconstruction fundamentals" reviews the implementation steps of the FDK and the M-line algorithm. Section "Towards real cone-beam data" presents strategies to adapt the M-line algorithm to the requirements imposed by real cone-beam data. Experiments are presented in Section "Experiments and results". Section "Summary and conclusions" summarizes our results.

C-arm imaging and geometry

We first give the mathematical formulation of the X-ray attenuation process, which is fundamental for analytical image reconstruction. Next, we describe the C-arm flat-panel detector geometry along with relevant coordinate systems. Then, two trajectories that are well suited for C-arm systems are introduced: the well-known circle (or partial circle) trajectory, which is commonly used for approximate image reconstruction, and the circle-plus-arc trajectory, which fulfills the sufficiency condition for accurate image reconstruction. For both trajectories, we provide a mathematical notation and show how the trajectories can be performed with our particular C-arm system.

Data acquisition

Data acquisition will be discussed using, as an example, the Artis zee ceiling-mounted C-arm system; see Fig. 1. For the acquisition of real cone-beam data, the X-ray beam is focused on a region of interest (ROI) within the inspected object and the source-detector assembly follows a specific trajectory around this ROI. Conventionally, this trajectory is a circle or partial circle, and the acquired cone-beam projections are used for subsequent FDK reconstruction.

For analytical image reconstruction, the measurements are viewed as line integrals of a function $f(\underline{x})$, namely

$$g(\lambda,\underline{\theta})=\int_0^\infty f(\underline{a}(\lambda)+t\underline{\theta})\,\mathrm{d}t. \quad (1)$$

In this notation, $\lambda \in \mathbb{R}$ is the source trajectory parameter (typically the rotation angle of the source-detector assembly); it controls the location of the X-ray source with respect to the source trajectory. The X-ray source is pictured as an ideal point source. The location of this point source is given by the vector $\underline{a}(\lambda)$. The parameter $\underline{\theta}$, with $\|\underline{\theta}\| = 1$, denotes the direction of the emitted X-ray. At fixed λ, we get one value of $g(\lambda, \underline{\theta})$ for each line that connects the point source to a detector pixel; the set of these values, which is essentially the X-ray image, is called a cone-beam projection of f.

Function $f(\underline{x})$ is our reconstruction target; it returns the X-ray linear attenuation coefficient (LAC) of the inspected object at location $\underline{x}$, or more precisely an energy-weighted average of this coefficient, with the weight defined by the energy spectrum of the X-ray source and the energy response of the detector. Logarithmic correction and other common calibration steps are necessary to obtain, from the detector pixel values, measurements that match the model of equation 1 as closely as possible; see e.g., [7,8] for further details). The values taken by $f(\underline{x})$ are expressed in Hounsfield units (HU).

Scanner geometry

The Artis zee ceiling-mounted C-arm system (Siemens AG, Healthcare Sector, Forchheim, Germany) is equipped with a movable C-arm. The C-arm carries a 30 × 40 cm^2 flat-panel detector with a diagonal of 48 cm. The detector elements have a side length of 154 μm, which amounts to a maximal matrix size of 1920 × 2480 pixels. Up to 60 cone-beam projections are acquired per second with 14 bit digitization depth. The source-detector-distance d can be adjusted from 90 cm to 120 cm.

The C-arm flat-panel scanner geometry is depicted in Fig. 2. To describe the scanner geometry, we define four coordinate systems: world coordinate system, source coordinate system, detector coordinate system, and image coordinate system.

The world coordinate system is spanned by orthogonal unit vectors $\underline{e}_x = (1, 0, 0)^\top$, $\underline{e}_y = (0, 1, 0)^\top$, and $\underline{e}_z = (0, 0, 1)^\top$. While the world coordinate system remains fixed during the scan,[1] the source coordinate system is attached to the X-ray source and thus moves with the source detector assembly. The source coordinate system is spanned by orthogonal unit vectors $\underline{e}_u$, $\underline{e}_v$, and $\underline{e}_w$. Vectors $\underline{e}_u$ and $\underline{e}_v$ point in the direction of the detector rows and columns, respectively, and vector $\underline{e}_w$ points from the detector toward the X-ray source. The X-ray beam with direction $\underline{\theta}$ intersects the flat-panel detector at coordinates $(u, v)^\top$, with

$$u=-d(\underline{\theta}\cdot\underline{e}_u)/(\underline{\theta}\cdot\underline{e}_w),\qquad v=-d(\underline{\theta}\cdot\underline{e}_v)/(\underline{\theta}\cdot\underline{e}_w).\quad (2)$$

The computed coordinates u and v refer to the detector coordinate system, the origin of which is at the orthogonal projection of the X-ray source onto the detector plane (called principal point); the u and v axes are parallel to the vectors $\underline{e}_u$ and $\underline{e}_v$.

The image coordinate system is used to access a specific pixel within the cone-beam projection. To express the detector point $(u, v)^\top$ in the image coordinate system, we apply the following affine transformations

$$\widehat{u}=(u+u_0)/(\Delta u),\qquad \widehat{v}=(v+v_0)/(\Delta v).\quad (3)$$

The quantities Δu and Δv denote the pixel width and height, respectively. In general, Δu and Δv may be different from each other. However, the flat-panel detector of the Artis C-arm system has square detector elements and so $\Delta u = \Delta v$. The parameters u_0 and v_0 define the origin of the image coordinate system, with respect to the detector coordinate system; for convenience, we select this origin to be the mid-point of the lower left image pixel, though a different choice could have been made just as well.

Vice versa, the detector coordinates for a given pixel location, $(\hat{u}, \hat{v})^\top$, are $u = \hat{u}\Delta u - u_0$ and $v = \hat{v}\Delta v - v_0$, and the direction of the ray that connects the detector point $(u, v)^\top$ to the X-ray source is given by

$$\underline{\theta}=\frac{u\underline{e}_u+v\underline{e}_v-d\underline{e}_w}{\sqrt{u^2+v^2+d^2}}.\quad (4)$$

Note that all vectors $(\underline{e}_x, \underline{e}_y, \underline{e}_z, \underline{e}_u, \underline{e}_v, \underline{e}_w, \underline{a}(\lambda), \underline{\theta})$ are defined with respect to the world coordinate system.

Trajectories suited for C-arm systems

For rotational image acquisition, the Artis zee ceiling-mounted C-arm system offers two options of system rotation; see Fig. 1. In the rotational mode, the C-arm is propelling. In the orbital mode, the C-arm segments are sliding in each other for a source-detector rotation. The angular range of rotation is 330° and 200° for the rotational and orbital mode, respectively. Theoretically, the X-ray source may move arbitrarily on the spherical surface

[1]In practice, the world coordinate system is defined by a calibration phantom, which is used to calibrate the source trajectory; see, e.g., Hoppe et al. [9] for details.

restricted to the angular range of those two modes. We assume a world coordinate system with its origin located in the iso-center, its *z*-axis points in axial direction along the patient's table long axis and its *x*- and *y*-axes are perpendicular. Let us denote the azimuth and polar angle with λ_1 and λ_2. A source position may be expressed as follows

$$\underline{\tilde{a}}(\lambda_1,\lambda_2)=r\cdot(\cos\lambda_1\cos\lambda_2,\sin\lambda_1\cos\lambda_2,\sin\lambda_2)^\top. \quad (5)$$

The parameter r denotes the source-iso-center distance, i.e., the radius of the scan. The typical source-iso-center distance is $r = 78.5$ cm. If the C-arm system is placed in the head site position, the axial rotation (rotation axis in axial direction) and the polar rotation (rotation axis perpendicular to axial direction) are performed by the rotational and orbital mode, respectively. The C-arm rotational modes are changed by each other if the C-arm system is placed in the right or left site position.

In the following, we give the description of two trajectories, which can be performed with our C-arm system. The circle (or partial circle) trajectory is used for FDK reconstruction. The circle-plus-arc trajectory is an extended (complete) trajectory, which can be utilized for accurate image reconstruction. We restrict our discussion to these two trajectories, although other C-arm suited trajectories for which image reconstruction algorithms exist would also be possible (e.g., the circle-plus-line trajectory [10] and the saddle trajectory [11]).

Circle trajectory—The circle trajectory is routinely used for image reconstruction in the clinical environment; it is performed with a standard, pre-configured data acquisition protocol available on commercial C-arm systems. In fact, the utilized circle trajectory typically describes only a partial circle, termed short scan in this context. The circle trajectory may be performed by rotating the C-arm purely in axial direction. The circle trajectory is given by equation (5) with $\lambda_2 = 0$. Thus

$$\underline{\tilde{a}}(\lambda_1,0)=\underline{a}(\lambda_1)=r\cdot(\cos\lambda_1,\sin\lambda_1,0)^\top. \quad (6)$$

Here, λ_1 is a azimuth angle in the (x, y)-plane.

Circle-plus-arc trajectory—The circle-plus-arc trajectory is a so-called complete trajectory according to Tuy [12] and therefore qualifies for accurate image reconstruction. Although currently not implemented on any commercially available device, the data acquisition protocol can be configured easily on a C-arm system. To do so, we split the trajectory into two segments, circle segment and arc segment. The data along the segments are then acquired independently of one another. For the circle segment, the standard, pre-configured data acquisition protocol is selected and the C-arm is rotated purely in axial direction. For the arc segment, the C-arm is configured to rotate in polar direction. The configured circle-plus-arc trajectory is practically limited to a maximal angular range of ±22° before the source-detector assembly collides with the patient table. Fortunately, the detector size is such that collecting arc data over this limited range is more than sufficient to complete the data from the circle segment and thereby allows a reconstruction free from cone-beam artifacts in the volume covered by the detector area (see [2] for mathematical details on how to evaluate detector-size and arc-length requirements for desired volumes).

The circle segment of the circle-plus-arc trajectory is defined in equation (6). Without loss of generality, assume that the circle scan starts at the positive *x*-axis from location $(r, 0, 0)^\top$. If we select the arc segment to intersect the point $(r, 0, 0)^\top$, we get from equation (5) with $\lambda_1 = 0$

$$\underline{\tilde{a}}(0,\lambda_2)=\underline{a}(\lambda_2)=r\cdot(\cos\lambda_2,0,\sin\lambda_2)^{\top},\ -22\pi/180\le\lambda_2\le 22\pi/180, \quad (7)$$

for the arc segment. According to this definition, the circle-plus- arc trajectory is two-sided, which means that the arc segment reaches to both sides of the (*x, y*)-plane. A one-sided trajectory may be configured by further limiting the polar angle, i.e., by selecting either $-\frac{22}{180}\pi\le\lambda_2\le 0$ or $0\le\lambda_2\le\frac{22}{180}\pi$. However, a two-sided circle-plus-arc trajectory allows an accurate reconstruction of the inspected object below and above the (*x, y*)-plane. A one-sided trajectory allows us to accurately reconstruct the object only on the respective arc side of the (*x, y*)-plane.

Image reconstruction fundamentals

This section reviews two algorithms to solve the image reconstruction task. The approximate FDK algorithm for circular trajectories is state-of-the-art in modern computed tomography and acts as a basis for comparison. The accurate *M*-line algorithm for the circle-plus-arc trajectory is investigated in this work.

FDK algorithm

The description of this algorithm, which was suggested by Feldkamp, Davis, and Kress in 1984 [13], is provided in “Appendix”.

M-line algorithm

The *M*-line reconstruction algorithm was invented in 2005 by Pack and Noo [1] as an accurate image reconstruction method. In general, this reconstruction formula can be applied to any complete source trajectory according to the definition given by Tuy. However, the presented algorithmic steps are specific to its application on the circle-plus-arc trajectory. Figure 3 explains the fundamental setup for a better understanding of those steps.

As can be seen in the left portion of this figure, the method involves a specific point, called *M*-point, on the source trajectory. In general, this point can vary for each $\underline{x}$ inside the ROI, but at a cost in efficiency. For our purposes, we assume the *M*-point is chosen fixed. That means that all object points are associated with the same *M*-point. This *M*-point is preferably located (approximately) in the middle of the circle segment, as done in [1] and Hoppe et al. [14]. This selection mitigates axial data truncation in case of long objects (see also Hoppe et al. [15]). The line connecting $\underline{x}$ with the *M*-point is a so-called *M*-line. In this algorithm, the *M*-line defines the filtering directions for $\underline{x}$. More exactly, for each cone-beam projection, the filtering direction for $\underline{x}$ is defined by the projection of the *M*-line associated with $\underline{x}$ onto the detector plane (see also in “Step 3: forward rebinning”).

The left of Fig. 3 shows a second line passing through $\underline{x}$, a so-called π-line. A π-line is a line that connects two opposite source positions on the trajectory. By definition, a π-line is unique in the sense that there exists only one such line for any given $\underline{x}$. A well-known property of π-lines is that any point on such a line fulfills Tuy’s data completeness condition, since all planes passing through a specific point on that line intersect the source trajectory. The existence and uniqueness of π-lines for the circle-plus-arc trajectory was shown by Katsevich [2]. It was also shown that those π-lines cover a large volume; see the right of Fig. 3. The π-line containing $\underline{x}$ delimitates the portion of the trajectory from where the data must be backprojected onto $\underline{x}$, to reconstruct $f(\underline{x})$ accurately. Backprojection is carried out independently over three segments, and then, the contributions are accumulated. The first segment $\{\lambda_1^-;\lambda_1^+(\underline{x})\}$ defines the portion of the trajectory from the *M*-point to the point where the π-line through $\underline{x}$ intersects the circle segment. The second segment

{ $\lambda_2^-(\underline{x});\lambda_2^+$} reaches from the π-line's other intersection with the arc segment to the foot point of the arc segment. The third segment { $\lambda_3^-;\lambda_3^+$} goes from this foot point back to the M-point.

In the following sections, we review the processing steps of the M-line algorithm in the specific context of reconstruction from a circle-plus-arc trajectory. For a derivation of the algorithm, we refer to [1].

Step 1: view-dependent differentiation—Step 1 is the differentiation of the cone-beam data with respect to the source trajectory parameter (typically the rotation angle of the source-detector assembly) at fixed ray direction $\underline{\theta}$, according to

$$g_1(\lambda,u,v)=\frac{\partial}{\partial\lambda}g(\lambda,\underline{\theta})\Big|_{\underline{\theta}\text{ fixed}}. \quad (8)$$

The implementation of this step may be critical in terms of resolution and image quality and is discussed in Section "View-dependent differentiation".

Step 2: cosine weighting—Perform a cosine weighting, as known from the FDK algorithm

$$g_2(\lambda,u,v)=\frac{d}{\sqrt{u^2+v^2+d^2}}g_1(\lambda,u,v). \quad (9)$$

Step 3: forward rebinning—Generate values on a rebinned detector grid by interpolation of the values from the original detector grid, such that each row of the rebinned detector holds the values for exactly one filtering line; see Fig. 4. The filtering lines are the projections of the M-lines onto the detector plane (cf. Fig. 3, left); as depicted in Fig. 4, they all intersect at the same point, namely the projection of the M-point, located at $(u_M, v_M)^\top$.

Mathematically, the forward rebinning corresponds to a coordinate transform from detector coordinates $(u, v)^\top$ to filtering line coordinates $(u, \eta)^\top$, where η denotes the slope of the filtering line, according to

$$g_3(\lambda,u,\eta)=g_2(\lambda,u,v^*), \qquad v^*=\eta(u-u_M)+v_M \quad (10)$$

Note that the implementation of the Katsevich algorithm for the helix trajectory contains a similar rebinning step, called forward height rebinning there [16].

Step 4: Hilbert filtering—Apply a one-dimensional Hilbert transform along the filtering lines

$$g_4(\lambda,u,\eta)=\int_{-\infty}^{+\infty}h_{hilb}(u-u')g_3(\lambda,u',\eta)du', \quad (11)$$

where $h_{hilb}(u) = 1/(\pi u)$ denotes the Hilbert kernel in the spatial domain.

Step 5: backward rebinning—Generate values on the original detector grid by interpolation of the values from the rebinned detector grid; see Fig. 4.

The backward rebinning corresponds to a coordinate transform from filtering line coordinates $(u, \eta)^\top$ back to detector coordinates $(u, v)^\top$ according to

$$g_5(\lambda, u, v)=g_4(\lambda, u, \eta^*), \qquad \eta^*=\frac{v-v_M}{u-u_M}. \quad (12)$$

Step 6: data selection—Select for each $\underline{x}$ the subset of filtered projections that corresponds to the portion of the trajectory that is delimitated by the intersection points of the π-line through $\underline{x}$ with the source trajectory, i.e., select the data that must be backprojected onto $\underline{x}$; cf. Fig. 3, left. This data selection is done according to

$$g_6(\lambda, u, v)=w_2(\lambda, u, v)g_5(\lambda, u, v). \quad (13)$$

The function $w_2(\lambda, u, v)$ takes only values between zero and one and masks out data that do not need to be backprojected, as shown in Section “Data selection”.

Step 7: backprojection—Backproject selected filtered projections into the image space, according to

$$f(\underline{x})=-\sum_{q=1}^{3}\frac{t_q}{2\pi^2}\int_{\lambda_q^-}^{\lambda_q^+}\frac{g_6(\lambda, u^*, v^*)}{(\underline{x}-\underline{a}(\lambda))\cdot\underline{e}_w}d\lambda, \quad (14)$$

where $(u^*, v^*)^\top$ are the detector coordinates of the projection of $\underline{x}$ onto the detector plane

$$u^*=-d\frac{(\underline{x}-\underline{a}(\lambda))\cdot\underline{e}_u}{(\underline{x}-\underline{a}(\lambda))\cdot\underline{e}_w}, \qquad v^*=-d\frac{(\underline{x}-\underline{a}(\lambda))\cdot\underline{e}_v}{(\underline{x}-\underline{a}(\lambda))\cdot e_w}, \quad (15)$$

and $t_1 = 1$, $t_2 = -1$, $t_3 = -1$; cf. Figure 3, left. Also, $\lambda_1^-=\lambda_1^+/2$, $\lambda_2^+=0$, $\lambda_3^-=\lambda_2^+$, $\lambda_3^+=\lambda_1^-$, and

$$\begin{aligned}\lambda_1^+&=2\pi-\arccos(r_c/r)-\arccos(r_c/(r\cos\lambda^*))\\ \lambda_2^-&=2\arctan(h_c/(r-r_c))\end{aligned} \quad (16)$$

Note that λ_1^+ and λ_2^- do not depend on $\underline{x}$ here and really denote the endpoints of the source trajectory and not the endpoints of the backprojection interval of $\underline{x}$. The correct data selection is guaranteed by the weighting mask $w_2(\lambda, u, v)$.

The value for λ_2^- is the same as s_2^{mx} from Katsevich [2] and corresponds to the minimal angular range of the arc segment, required to reconstruct a cylindrical ROI with radius r_c and height h_c, centered at $(0, 0, 0)^\top$; see [2], equation (4.4). Similarly, one may want to compute the minimal required angular range of the circle segment λ_1^+ for the same ROI by first substituting $h = r_c$ and $x_3^{mn}=h_c$ into equation (4.2) of [2] and solving for $s_2 \le s_2^{mx}$ and then plugging $\lambda^* = s_2$ into the formula for λ_1^+ above. However, doing so appears too cumbersome and is actually not needed. For a practical implementation, we use $\lambda^*=\lambda_2^-$; see Fig. 5. This gives a minimally larger value for λ_1^+, and the angular range of the circle segment becomes a little larger than absolutely required. No consequences on the reconstruction quality are to be expected by doing so, since, as remarked before, the selection of the backprojection interval for each $\underline{x}$ is taken care of by the weighting mask $w_2(\lambda, u, v)$ already in the data selection step (Fig. 6).

Toward real cone-beam data

In this section, we first show how the geometrical parameters required to implement the FDK and M-line reconstruction formulae may be extracted from projection matrices. We assume that these projection matrices have been obtained from the geometrical calibration presented in the study by Hoppe et al. [9]. Other calibration techniques that are based on a direct estimation of the geometrical parameters, such as Noo et al. [17], may not require this extraction. The main focus of this section is dedicated to presenting solutions to two challenging implementation issues for the M-line algorithm such that the algorithm is capable of handling real C-arm data from the circle-plus-arc trajectory. In particular, we apply a novel and numerically stable technique for the differentiation with respect to the source trajectory parameter, and we present an efficient way to compute the π-line backprojection intervals via a polygonal weighting mask.

Parameter extraction

To extract the geometrical parameters from a projection matrix, we follow steps similar to Hartley and Zisserman [18]. Let P be the 3×4 projection matrix for a specific source position $\underline{a}$. This projection matrix can be decomposed into the following product

$$P=\begin{bmatrix} \frac{d}{\Delta u} & s_0 & \widehat{u}_0 \\ 0 & \frac{d}{\Delta v} & \widehat{v}_0 \\ 0 & 0 & 1 \end{bmatrix} [R|-R\underline{a}]=K\,[R|-R\underline{a}]. \quad (17)$$

The upper, 3 × 3, triangular matrix K is called internal parameter matrix. The 3 × 3 rotation matrix R consists of the unit vectors that define the detector coordinate system and thus provides the detector orientation, namely $R = [\underline{e}_u \underline{e}_v - \underline{e}_w]^\top$.

To obtain the complete parameter set, we need to extract K, R, and $\underline{a}$ from P. For that purpose, we reformulate equation (17) as follows

$$P=[K\,R|-K\,R\underline{a}]=[M|-M\underline{a}]. \quad (18)$$

The matrix $M = K\,R$ is the left sub-matrix of P. Since K is upper triangular and R is orthogonal, K and R can be determined from M by using an RQ Decomposition.[2] The decomposition is unique if we require that K has positive diagonal entries. This can be achieved by multiplying K on the right and R on the left by diag(sign(k_{11}), sign(k_{22}), sign(k_{33})), where the k_{ii} 's are the diagonal entries of K.

From (18), we see that $\underline{a}$ can be computed according to $\underline{a} = -M^{-1}\,\underline{p}^4$ where $\underline{p}^4$ denotes the last column of P. The inversion of M can be done in a numerically stable way, e.g., by using a singular value decomposition.

[2]Sometimes scientific computing packages (such as Matlab) do not offer an RQ Decomposition but offer instead a QR Decomposition. In this case, one can first perform a QR decomposition on M^{-1} followed by an inversion of the obtained matrices since $M = (M^{-1})^{-1} = (R^{-1}K^{-1})^{-1} = K\,R$. Note that $R^{-1} = R^\top$.

Given the complete data set that describes the C-arm geometry, we are prepared to implement the required processing steps of the *M*-line algorithm. We need those parameters on many occasions throughout the implementation. For instance, $\hat{u}_0$ and $\hat{v}_0$ are required to perform a coordinate transform from image coordinates to detector coordinates or vice versa. If we want to compute for a given detector coordinate the corresponding X-ray direction, we need $\underline{e}_u$, $\underline{e}_v$, $\underline{e}_w$, and d. And we need $\underline{a}$ to compute the view-dependent derivative, as explained in the following section.

View-dependent differentiation

Many accurate image reconstruction formulae impose the front-end requirement of a view-dependent data differentiation step. This view differentiation step has often been assumed to be suboptimal with respect to the resolution and the quality of the reconstructed image because the discretization along the source trajectory in λ is typically coarse, compared to the much finer sampling on the detector in u and v.

Former differentiation schemes include the direct scheme and the chain-rule scheme. The direct scheme may provide good results when the sampling rate along the source trajectory is high but has been shown to break down in case of coarse view sampling [19]. The chain-rule scheme is more robust to view sampling and can provide higher resolution results than the direct scheme, but not for all source trajectories. In particular, the chain-rule scheme was shown to yield very poor results with data from a circle-plus-line trajectory [19]. For this reason, we adopted a recently proposed view differentiation scheme [19] that seems to be robust to changes in the data acquisition geometry as well as to coarse view sampling. This scheme is summarized below.

First, note that the differentiation of the cone-beam data $g(\lambda, \underline{\theta})$ with respect to the trajectory parameter λ at fixed viewing direction $\underline{\theta}$ can be discretized into

$$\frac{\partial}{\partial\lambda}g(\lambda,\underline{\theta})\Big|_{\underline{\theta}\text{ fixed}} \simeq \frac{g(\lambda+\varepsilon\Delta\lambda,\underline{\theta})-g(\lambda-\varepsilon\Delta\lambda,\underline{\theta})}{2\varepsilon\,\Delta\lambda}, \quad (19)$$

where ε can be seen as a resolution-control parameter, with $0 < \varepsilon \le 1$. If the trajectory is equidistantly sampled at increments $\Delta\lambda$ and λ is one of the samples, setting $\varepsilon = 1$ into (19) yields a formula that involves available cone-beam data. But this formula is not appealing due to the low sampling rate along the source trajectory. Instead, we would like to use $\varepsilon \ll 1$ to minimize discretization errors and resolution losses.

When $\varepsilon \ll 1$, applying equation (19) requires unavailable cone-beam data, namely $g(\lambda - \varepsilon\Delta\lambda, \underline{\theta})$ and $g(\lambda + \varepsilon\Delta\lambda, \underline{\theta})$. To be able to compute the derivative nevertheless, we use an interpolation scheme that estimates the desired (but unavailable) cone-beam data $g(\lambda - \varepsilon\Delta\lambda, \underline{\theta})$ and $g(\lambda + \varepsilon\Delta\lambda, \underline{\theta})$ from the available cone-beam data $g(\lambda - \Delta\lambda, \underline{\theta})$, $g(\lambda, \underline{\theta})$ and $g(\lambda+\Delta\lambda, \underline{\theta})$. This interpolation is performed using the following equations:

$$g(\lambda-\varepsilon\,\Delta\lambda,\underline{\theta}) \simeq (1-\varepsilon)\,g\,(\lambda,\underline{\alpha}^{-})+\varepsilon\,g\,\left(\lambda-\Delta\lambda,\underline{\beta}^{-}\right) \quad (20)$$

$$g(\lambda+\varepsilon\,\Delta\lambda,\underline{\theta}) \simeq (1-\varepsilon)\,g\,(\lambda,\underline{\alpha}^{+})+\varepsilon\,g\,\left(\lambda+\Delta\lambda,\underline{\beta}^{+}\right), \quad (21)$$

where

$$\underline{\alpha}^{\pm}=\frac{\underline{b}(\lambda \pm \varepsilon\,\Delta\lambda, \underline{\theta})-\underline{a}(\lambda)}{\left\|\underline{b}(\lambda \pm \varepsilon\,\Delta\lambda, \underline{\theta})-\underline{a}(\lambda)\right\|} \quad (22)$$

$$\underline{\beta}^{\pm}=\frac{\underline{b}(\lambda \pm \varepsilon\,\Delta\lambda, \underline{\theta})-\underline{a}(\lambda \pm \Delta\lambda)}{\left\|\underline{b}(\lambda \pm \varepsilon\,\Delta\lambda, \underline{\theta})-\underline{a}(\lambda \pm \Delta\lambda)\right\|} \quad (23)$$

Figure 2 in [19] gives a geometrical description for the unit vectors $\underline{\alpha}^-$, $\underline{\beta}^-$, $\underline{\alpha}^+$, and $\underline{\beta}^+$. Those vectors are defined by the three source positions $\underline{a}(\lambda - \Delta\lambda)$, $\underline{a}(\lambda)$, and $\underline{a}(\lambda + \Delta\lambda)$ as well as by the two points $\underline{b}(\lambda - \Delta\lambda, \underline{\theta})$, and $\underline{b}(\lambda + \Delta\lambda, \underline{\theta})$. The points $\underline{b}(\lambda - \Delta\lambda, \underline{\theta})$ and $\underline{b}(\lambda + \Delta\lambda, \underline{\theta})$ should be chosen such that the integral values along the lines with directions $\underline{\alpha}^+$, $\underline{\beta}^+$, and $\underline{\alpha}^-$, $\underline{\beta}^-$ approximate the desired integrals along the lines with direction $\underline{\theta}$, as good as possible. We assume that this will be the case if $\underline{b}(\lambda - \Delta\lambda, \underline{\theta})$ and $\underline{b}(\lambda + \Delta\lambda, \underline{\theta})$ are the orthogonal projections of the object's center of mass onto the two lines with direction $\underline{\theta}$, radiating from $\underline{a}(\lambda - \varepsilon\Delta\lambda)$ and $\underline{a}(\lambda + \varepsilon\Delta\lambda)$, respectively. Since for real data the true center of mass is rarely known, the origin of the world coordinate system is chosen as an approximation.

The values of $g(\lambda, \underline{\alpha}^-)$, $g(\lambda - \Delta\lambda, \underline{\beta}^-)$, and $g(\lambda, \underline{\alpha}^+)$, $g(\lambda + \Delta\lambda, \underline{\beta}^+)$ are obtained by bilinear interpolation in the detector plane after computing the detector coordinates corresponding to the involved lines using formulae (2).

Data selection

Besides the view differentiation step, there is another processing step in the reconstruction algorithm that requires some care during implementation, namely the data selection step. For the final algorithm, it remains to show how for each object points to practically select the subset of cone-beam projections that need to be backprojected. This data selection can be done analytically for an ideal trajectory. This has been shown, for instance, for the circle-plus-arc [2] and for the circle-plus-line trajectory [10]. Since we are dealing with a non-ideal trajectory, we present an approximation to the analytical formulae in this chapter. This approximation involves the computation of a two-dimensional weighting mask for each cone-beam projection, which must be multiplied with the detector content. This multiplication can be implemented efficiently. The method is not restricted to the circle-plus-arc trajectory but may also be applied to similar complete source trajectories, such as the circle-plus-line trajectory.

The M-line algorithm for the circle-plus-arc trajectory reconstructs the object by backprojecting the filtered cone-beam data onto π-lines, which must cover at least the ROI under investigation. Each object point $\underline{x}$ belongs to a specific π-line, which determines the backprojection interval for $\underline{x}$ and for all other object points on this π-line. More specifically, the backprojection interval is determined by the endpoints of the π-line, which are the two points of intersection of the line with the source trajectory. Data outside the backprojection interval is redundant and must not be backprojected. Otherwise, streak like artifacts that manifest along the π-lines will disturb the reconstructed object, since the redundant data are backprojected but not taken into account by the algorithm. It is therefore important to accurately compute the backprojection intervals. Such a computation was shown in [2] for an ideal circle-plus-arc trajectory. Our non-ideal data acquisition geometry forces us to approximate the analytic formulae. We do this approximation as follows.

First, we note that the decision if the current source position belongs to the backprojection interval of a given object point $\underline{x}$ can be made by projecting one segment of the trajectory onto the detector plane. More specifically, we project the arc segment onto the detector when the source moves along circle segment and we project the circle segment onto the detector when the source moves along the arc segment. We then select only those detector data at the appropriate side of the projected trajectory segment. The principle is illustrated in Fig. 7. Assume the source is on the circle segment. The current source position belongs to the backprojection interval of $\underline{x}$, only if $\underline{x}$ projects on the right-hand side of the projected arc segment. Since this is true for every object point, we can create a two-dimensional weighting mask $w_2(\lambda, u, v)$ that assigns each pixel on the right-hand side a value of one and a value of zero otherwise. The same principle applies if the source is on the arc segment. Here, the current source position belongs to the backprojection interval of $\underline{x}$, if $\underline{x}$ projects above the projected circle segment. Thus, the weighting mask must be one on top of the projected circle segment and zero elsewhere.

In a realistic scenario, we do not have an analytical description of the source trajectory. The source trajectory is described by a sequence of X-ray source positions. To project one trajectory segment, we first project the corresponding source positions onto the detector and then connect the projected points by lines. The resulting polygonal curve then represents the projected trajectory segment.

To avoid artifacts appearing along the π-lines in the final reconstructed object, we generate a smooth transition zone within a small neighborhood to the polygonal curve, by assigning each neighborhood pixel a value between zero and one. We do this by applying a one-dimensional convolution of the weighting mask with a simple averaging filter. When we project the arc segment, the convolution is done along $\underline{e}_u$. When we project the circle segment, we convolve along $\underline{e}_v$. The support of the averaging filter was selected to be three pixels, which we found worked best, at least for our specific C-arm geometry.

We would like to emphasize that the weighting mask $w_2(\lambda, u, v)$ depends only on the projection of $\underline{x}$ onto the detector plane, not on $\underline{x}$ itself. That means that all other points, which lie on the line connecting the current source position with $\underline{x}$, are assigned the same weight. Therefore, this data selection can be implemented efficiently.

Experiments and results

The *M*-line algorithm has been tested on real C-arm data. We used an AXIOM Artis *d*TA C-arm system, which is the precursor of the Artis zee ceiling-mounted system to perform the circle-plus-arc trajectory. The parameters are shown in Table 1. Note that since we reconstructed the volume above and below the circle plane, the arc scan actually consisted of two segments, one reaching above and the another reaching below the circle plane. To compute the view-dependent derivative, the resolution-control parameter ε was set to 2^{-6}, a value that yields highest resolution. The *M*-point was placed (approximately) in the middle of the circle segment. We calibrated the trajectory geometry with the method of Hoppe et al. [9].

Figure 8 show the results of our implementation of the *M*-line approach against a state-of-the-art implementation of the FDK algorithm applied to the data from the circle segment of the circle-plus-arc trajectory. An anthropomorphic thorax phantom was used for the comparison; this phantom consists of a human spine that is embedded in water-equivalent material to emulate real tissue including the heart and the liver. Each column in the figure displays a different slice through the phantom and includes the result from a CT scan to establish the anatomical ground truth. Each slice was obtained as an average of 11 sub-slices separated by 0.1 mm, using the same square pixels of size $0.5\times0.5\text{mm}^2$ for each sub-slice; this averaging was performed to reduce noise at low cost to spatial resolution. The CT images were obtained with a SOMATOM Definition AS (Siemens AG, Healthcare), using a spiral scan performed with the following parameters: 120kV, 29mA, slice thickness of 1 mm, B46f kernel. Registration of the C-arm images to the CT images was done using the syngo 3D image fusion software (Siemens AG, Healthcare Sector) that is based on a mutual information cost function. Note that slight residual registration errors remain, due to the complexity of the registration task. Also, due to differences in X-ray source spectra, truncation errors, and scatter, only an approximate matching in HU units can be expected. Data truncation problems for the M-line reconstruction were handled using the Basic Approach from Hoppe et al. [15]. Data truncation problems for the FDK reconstruction were handled using the modified water cylinder correction of Zellerhoff et al. [8].

The spine in the thorax phantom is a structure that is highly sensitive to data incompleteness in a circular short scan, and allows thereby easy visual assessment of cone-beam artifacts. The results in Fig. 8 show a substantial elimination of cone-beam artifacts from using the M-line method with the circle-plus-arc data, in comparison with employing (incomplete) circular short-scan data along with the FDK method. The typical cone-beam artifacts we see in the FDK reconstruction manifest in dark directed shadows radiating from the bones of the spine whenever there is a noticeable frequency change in z-direction.

Given the more complicated data processing steps being involved in the M-line method, it is also nice to observe very little difference in discretization errors, highlighting the robustness of our implementation scheme. Only one slight streak-shaped artifact appears noticeable (see the bottom image in the middle column of Fig. 8).

We performed a preliminary evaluation of noise and resolution, comparing the M-line method with the circle-plus-arc data to the FDK reconstruction from the data on the circle segment. By comparing various profiles (not shown here) through the slices in Fig. 8, we could observe that the differences in resolution are tiny. To quantify the difference, we determined the standard deviation of a Gaussian kernel that allows minimizing the least-square difference between the modulation transfer function for each reconstruction method, which was obtained by computing the average of the Fourier transform of each image over the polar angle. A standard deviation of 0.3543 pixels was found in favor of the FDK image. To evaluate noise, we first applied the Gaussian kernel to the FDK image, so as to globally equalize the resolution, then we computed the standard deviation inside an homogeneous region of interest of 60 × 60 pixels. The following values were obtained: 69.97 HU for the FDK image and 65.17 HU for the M-line reconstruction. Thus, the M-line image has a lower standard deviation; but the M-line image also uses more projections. Considering that the M-line method uses 543+108/2 = 597 projections, where as the FDK method uses 543 projections only, the FDK result may be expected to be more noisy, by coarsely 4.85%, which is comparable to the observed difference[3]. Hence, it appears at first hand that the M-line method does not amplify noise in the data more than the FDK method; more thorough evaluations of noise will, however, be needed to properly validate this statement.

Summary and conclusions

We have shown how to implement two important processing steps of the M-line approach such that the algorithm is able to handle cone-beam data from a circle-plus-arc trajectory, acquired with a real C-arm system.

As a prerequisite for implementing the algorithm, we reviewed how the necessary geometrical parameters that describe the data acquisition geometry can be extracted from

[3]Assuming the pixels values in the region of interest are statistically independent, the standard deviation of our computed values can be estimated to be about $70HU/\sqrt{2 \times 60 \times 60}=0.825HU$.

projection matrices. This may become important in practice if the calibration procedure outputs projection matrices, as in our case.

For the first critical processing step, we applied a novel and numerically stable way to compute the view-dependent derivative in the context of the *M*-line algorithm for the circle-plus-arc trajectory. The computation is based on a direct discretization of the derivative, which involves a unique resolution-control parameter. For the derivative at λ, this parameter basically determines how much data are used from the neighboring cone-beam projections at $\lambda \pm \Delta\lambda$ and may need to be adjusted on a case-by-case basis for a fine tuning of the derivative computation, especially if $\Delta\lambda$ is big. As already pointed out, the suggested method can be applied to any accurate reconstruction algorithm that involves a view-dependent data differentiation step.

As the last step toward accurate image reconstruction from real C-arm data, we presented an approach to determine for each object point the backprojection interval in a robust and efficient way. We did this by creating for each cone-beam projection a two-dimensional detector weighting mask, which must be multiplied with the detector content prior to the backprojection step. The mask thereby ensures the correct handling of data redundancies and takes care that the backprojection interval borders have a smooth transition zone in order to avoid reconstruction artifacts. The creation of the weighting mask involves the projection of specific trajectory segments onto the detector. We approximated the projections of those segments by a polygonal curve that connects a set of projected X-ray source positions. In former experiments, we have tried to fit an ellipsoidal curve to the projection of the X-ray source positions. This was done by using standard linear algebra. We found out, however, that this fitting cannot be implemented in a robust way, since there are too few source positions for a stable estimate of the ellipsoid. Therefore, we decided to use the polygonal approach, which seems to be a good approximation and is robust. The suggested method can easily be adapted to other complete source trajectories, such as the circle-plus-line trajectory.

Our experiments with real data of a human thorax phantom show that by applying the presented implementation techniques, cone-beam artifacts can be considerably mitigated. This has been confirmed and tested against using the circle segment of the data only along with a state-of-the-art FDK reconstruction. Furthermore, the reconstruction with the *M*-line algorithm appears to have similar noise and resolution. And our implementation of the *M*-line algorithm appears robust enough to avoid discretization errors that would have outweighted the benefits in cone-beam artifacts mitigation.

Acknowledgments

This work was supported by Siemens AG, Healthcare Sector, and was supported in part by the U.S. National Institutes of Health (NIH) under grants R01 EB000627 and R01 EB007236.

References

1. Pack J, Noo F. Cone-beam reconstruction using 1d filtering along the projection of m-lines. Inverse Probl. 2005; 21(3):1105–1120.
2. Katsevich A. Image reconstruction for the circle-and-arc trajectory. Phys Med Biol. 2005; 50(10): 2249–2265. [PubMed: 15876665]
3. Dennerlein F, Katsevich A, Lauritsch G, Hornegger J. Exact and efficient cone-beam reconstruction algorithm for a short-scan circle combined with various lines. Proc SPIE. 2005; 5747:388–399.
4. Grass M, Koppe R, Klotz E, Proksa R, Kuhn MH, Aerts H, Op De Beek J, Kemkers R. Three-dimensional reconstruction of high contrast objects using c-arm image intensifier projection data. Comput Med Imaging Graph. 1999; 23:311–321. [PubMed: 10634143]

5. Riddell C, Trousset Y. Rectification for cone-beam projection and backprojection. IEEE Trans Med Imaging. 2006; 25(7):950–962. [PubMed: 16827495]

6. Wiesent K, Barth K, Navab N, Durlak P, Brunner T, Schütz O, Seissler W. Enhanced 3-d-reconstruction algorithm for c-arm systems suitable for interventional procedures. IEEE Trans Med Imaging. 2000; 19(5):391–403. [PubMed: 11021683]

7. Buzug, TM. Springer; Berlin: 2008. Computed tomography—from photon statistics to modern cone-beam CT.

8. Zellerhoff M, Scholz B, Rührnschopf E-P, Brunner T. Low contrast 3d-reconstruction from c-arm data. Proc SPIE. 2005; 5745:646–655.

9. Hoppe S, Noo F, Dennerlein F, Lauritsch G, Hornegger J. Geometric calibration of the circle-plus-arc trajectory. Phys Med Biol. 2007; 52(23):6943–6960. [PubMed: 18029986]

10. Katsevich A. Image reconstruction for the circle-and-line trajectory. Phys Med Biol. 2004; 49(22): 5059–5072. [PubMed: 15609558]

11. Pack J, Noo F, Kudo H. Investigation of saddle trajectories for cardiac ct imaging in cone-beam geometry. Phys Med Biol. 2004; 49(11):2317–2336. [PubMed: 15248580]

12. Tuy HK. An inversion formula for cone-beam reconstruction. SIAM J Appl Math. 1983; 43(3): 546–552.

13. Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. J Opt Soc Am A. 1984; 1(6): 612–619.

14. Hoppe, S.; Dennerlein, F.; Lauritsch, G.; Hornegger, J.; Noo, F. Cone-beam tomography from short-scan circle-plus-arc data measured on a c-arm system. Nuclear Science Symposium and Medical Imaging Conference; San Diego, CA, USA. Oct. 29–Nov. 4; 2006. p. 2873-2877.

15. Hoppe S, Hornegger J, Lauritsch G, Dennerlein F, Noo F. Truncation correction for oblique filtering lines. Med Phys. 2008; 35:5910–5920. [PubMed: 19175146]

16. Noo F, Pack J, Heuscher D. Exact helical reconstruction using native cone-beam geometries. Phys Med Biol. 2003; 48(23):3787–3818. [PubMed: 14703159]

17. Noo F, Clackdoyle R, Mennessier C, White TA, Roney TJ. Analytic method based on identification of ellipse parameters for scanner calibration in cone-beam tomography. Phys Med Biol. 2000; 45(11):3489–3508. [PubMed: 11098919]

18. Hartley, R.; Zisserman, A. Multiple view geometry in computer vision. 2. Cambridge University Press; Cambridge: 2003.

19. Noo F, Hoppe S, Dennerlein F, Lauritsch G, Hornegger J. A new scheme for view-dependent data differentiation in fanbeam and cone-beam computed tomography. Phys Med Biol. 2007; 52(17): 5393–5414. [PubMed: 17762094]

20. Parker DL. Optimal short scan convolution reconstruction for fanbeam ct. Med Phys. 1982; 9(2): 254–257. [PubMed: 7087912]

21. Silver MD. A method for including redundant data in computed tomography. Med Phys. 2000; 27(4):773–774. [PubMed: 10798699]

22. Rodet T, Noo F, Defrise M. The cone-beam algorithm of feldkamp, davis and kress preserves oblique line integrals. Med Phys. 2004; 31(7):1972–1975. [PubMed: 15305448]

23. Kak, AC.; Slaney, M. Principles of computerized tomographic imaging. 1. The Institute of Electrical and Electronics Engineers Inc; New York: 1988.

Appendix: FDK algorithm

The FDK algorithm was initially designed for a circle trajectory. It was later modified to process cone-beam data also from other trajectories, as, for instance, from a partial circle trajectory [20,21].

Because the FDK algorithm utilizes an incomplete trajectory, it is an approximate image reconstruction algorithm. In fact, the FDK algorithm is based on a 2D image reconstruction formula and may be regarded as an empirical 3D extension of a 2D algorithm [22].

The FDK algorithm reconstructs the attenuation coefficient f accurately only in the plane of the trajectory. Outside of this plane, the algorithm approximates f. One can show, however, that in the special case when the object is constant along the direction orthogonal to the plane of the trajectory, reconstruction is accurate also outside of that plane; cf. Feldkamp et al. [13].

The FDK algorithm applies a filtered backprojection technique to solve the reconstruction task in a computationally efficient way. Due to its efficiency, it has been implemented successfully on almost every commercially available medical CT imaging system and still maintains its state-of-the-art status in modern computed tomography, either in its original form or in various extensions [22].

In the following sections, we review the processing steps of the FDK algorithm. For a derivation of the algorithm, we refer to Feldkamp et al. [13] and to textbooks such as Buzug [7] or Kak and Slaney [23].

Step 1: cosine weighting

Perform a cosine weighting of the projection data g, according to

$$g_2(\lambda, u, v)=\frac{d}{\sqrt{u^2+v^2+d^2}}g_1(\lambda, u, v), \quad (24)$$

where we used $g_1(\lambda, u, v) = g(\lambda, \underline{\theta})$ to express each cone-beam value by its detector coordinates $(u, v)^\top$. The factor $\frac{d}{\sqrt{u^2+v^2+d^2}}$ weights each detector pixel value to normalize the distance between the detector pixel and the X-ray source position; see Fig. 9. This factor is equivalent to the cosine of the cone angle, ω, and one may verify that

$$\cos\omega=-\underline{e}_w\cdot\underline{\theta}=\frac{d}{\sqrt{u^2+v^2+d^2}}, \quad (25)$$

by replacing $\underline{\theta}$ with the right-hand side of equation (4) and noting that $\underline{e}_w \perp \underline{e}_u$ and $\underline{e}_w \perp \underline{e}_v$.

Step 2: Data Redundancy Weighting

Perform a data redundancy weighting, according to

$$g_3(\lambda, u, v)=w_1(\lambda, u)g_2(\lambda, u, v), \quad (26)$$

where the weighting function $w_1(\lambda, u)$ is given by

$$w_1(\lambda, u)=0.5 \quad (27)$$

for a full circle trajectory, and typically

$$w_1(\lambda, u)=\begin{cases}\sin^2\left(\frac{\pi\lambda}{4(\omega_{max}-\lambda^*)}\right) & 0\le\lambda\le 2\omega_{max}-2\lambda^*\\ 1 & 2\omega_{max}-2\lambda^*\le\lambda\le\pi-2\lambda^*\\ \sin^2\left(\frac{\pi(\pi+2\omega_{max}-\lambda)}{4(\omega_{max}+\lambda^*)}\right) & \pi-2\lambda^*\le\lambda\le\pi+2\omega_{max}\end{cases} \quad (28)$$

for a partial circle trajectory, where $2\omega_{max}$ is the fan angle and $\lambda^* = \arctan(u/d)$; this corresponds to the original Parker weighting [20].

The weighting function considers the fact that some line integrals (cone-beam values) through any given point inside the object are measured twice. This is true for points located in the plane of the trajectory, but not for points outside this plane, which are treated as if they were located at their orthogonal projection onto that plane, as appears evident from the function $w_1(\lambda, u)$ being independent of the v-coordinate.

Step 3: Ramp Filtering

Apply a one-dimensional convolution along the filtering lines (i.e., the detector rows) to obtain

$$g_4(\lambda, u, v)=\int_{-\infty}^{+\infty} h_{ramp}(u-u')g_3(\lambda, u', v)du', \quad (29)$$

where $h_{ramp}(u)$ denotes the Ramp kernel in the spatial domain

$$h_{ramp}(u)=\frac{-1}{2\pi^2 u^2}. \quad (30)$$

Here, we assume that $\underline{e}_u$ is parallel to the plane of the trajectory. If this assumption is violated, for instance, due to a slight detector tilt or rotation, those deviations are often assumed to be negligible in practical implementations and filtering is performed in the direction of $\underline{e}_u$ anyway.

Step 4: Backprojection

Backproject the filtered projections into the image space, according to

$$f(\underline{x})=rd\int_{\lambda^-}^{\lambda^+} \frac{g_4(\lambda, u^*, v^*)}{|(\underline{x}-\underline{a}(\lambda)) \cdot \underline{e}_w|^2} d\lambda, \quad (31)$$

where $(u^*, v^*)^\top$ are the detector coordinates of the projection of $\underline{x}$ onto the detector plane:

$$u^*=-d\frac{(\underline{x}-\underline{a}(\lambda)) \cdot \underline{e}_u}{(\underline{x}-\underline{a}(\lambda)) \cdot \underline{e}_w}, \qquad v^*=-d\frac{(\underline{x}-\underline{a}(\lambda)) \cdot \underline{e}_v}{(\underline{x}-\underline{a}(\lambda)) \cdot \underline{e}_w}. \quad (32)$$

The backprojection interval is given by

$$\lambda^-=0, \qquad \lambda^+=2\pi \quad (33)$$

for a full circle trajectory, and typically

$$\lambda^-=0, \qquad \lambda^+=\pi+2\omega_{max} \quad (34)$$

for a partial circle trajectory. For a cylindrical ROI with radius r_c, centered at $(0, 0, 0)^\top$, $\omega_{max} = \arcsin(r_c/r)$; see Kak and Slaney [23].

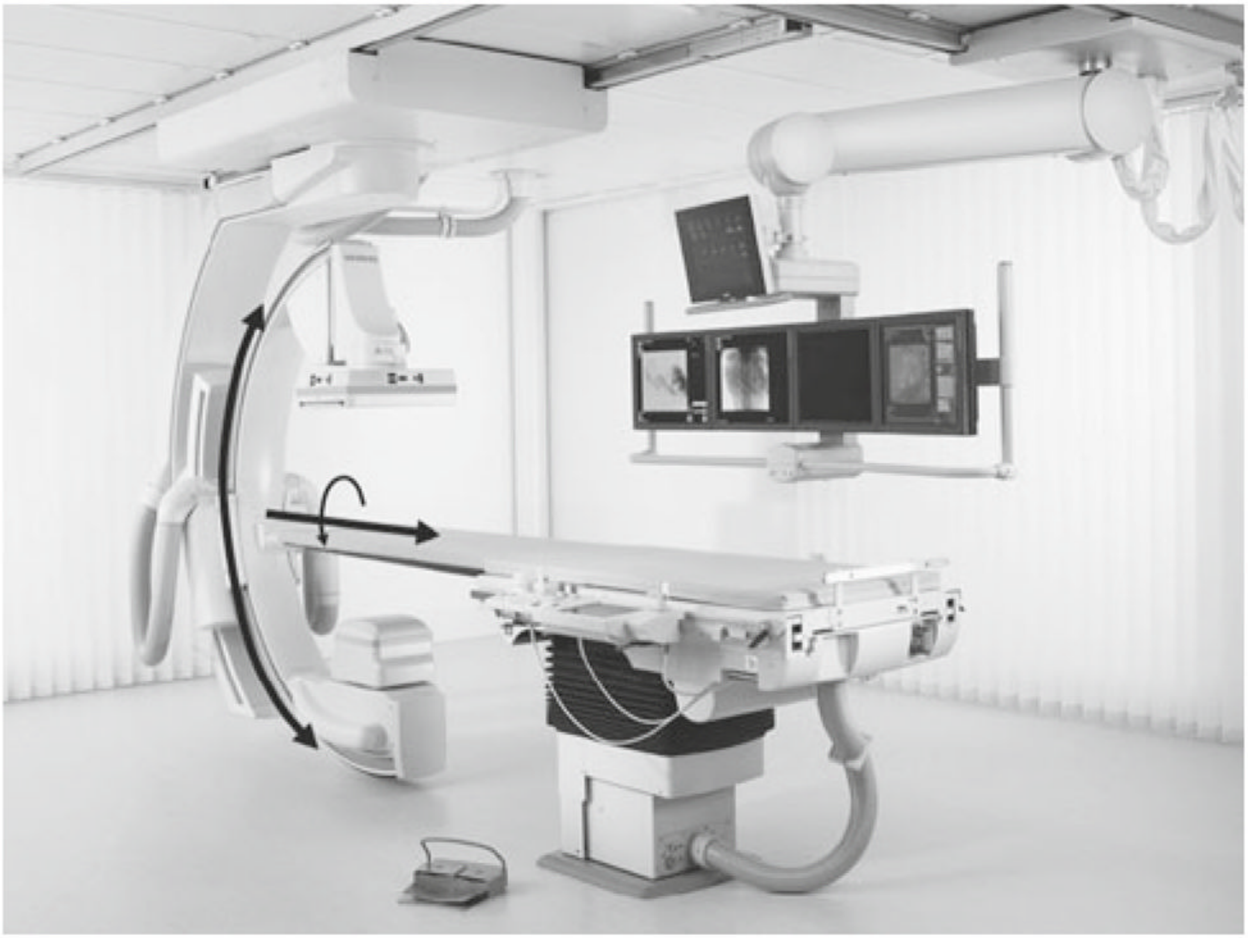

Fig. 1.
As an example, data acquisition is illustrated at the Artis zee ceiling-mounted C-arm system (Siemens AG, Healthcare Sector, Forchheim, Germany). Two modes of rotation are possible for rotational angiography. In the rotational mode, the rotation axis is parallel to the long axis of the patient table. In the orbital mode, rotation is performed by sliding of the C-arm segments. By courtesy of Siemens AG

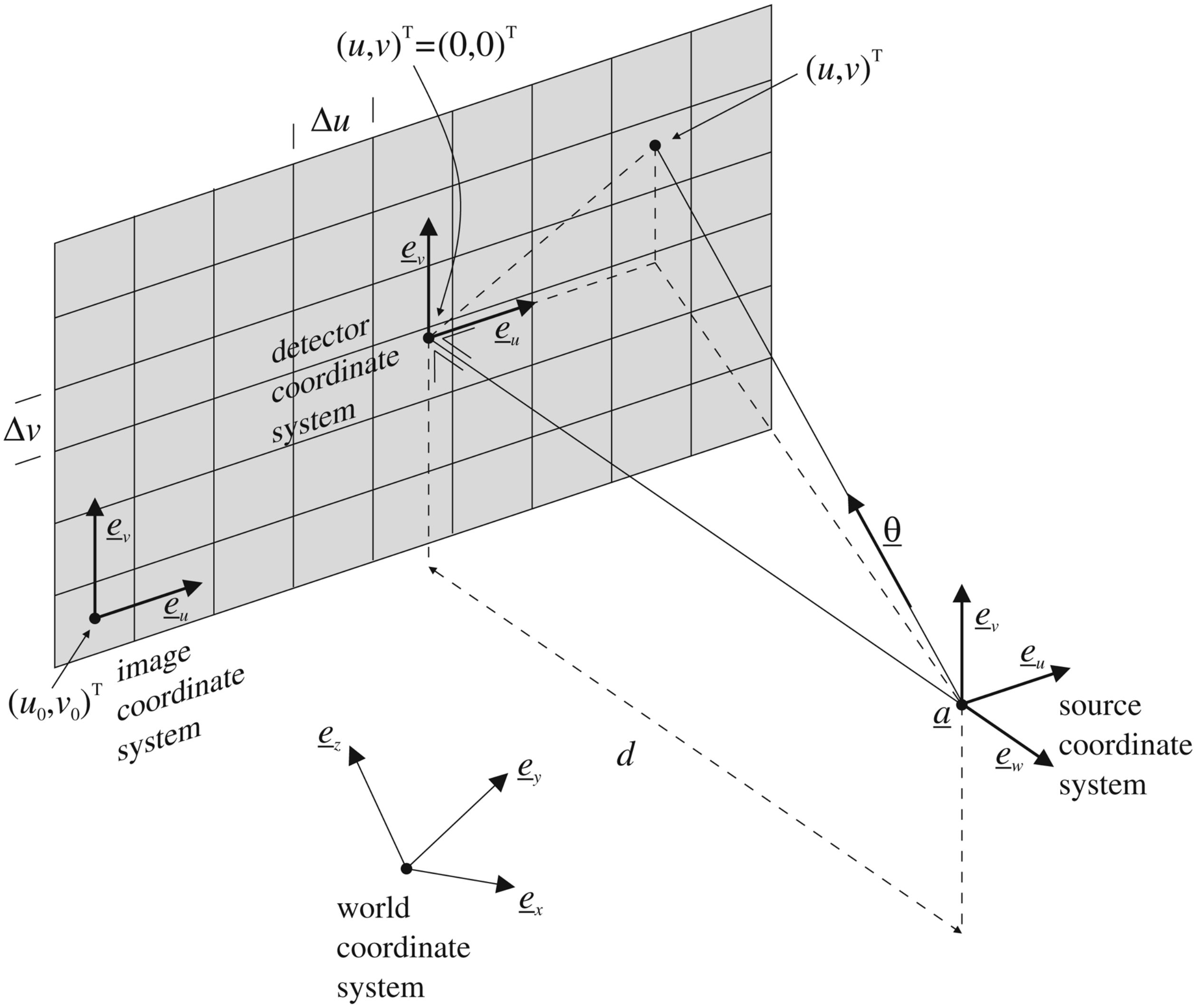

Fig. 2.
C-arm flat-panel scanner geometry

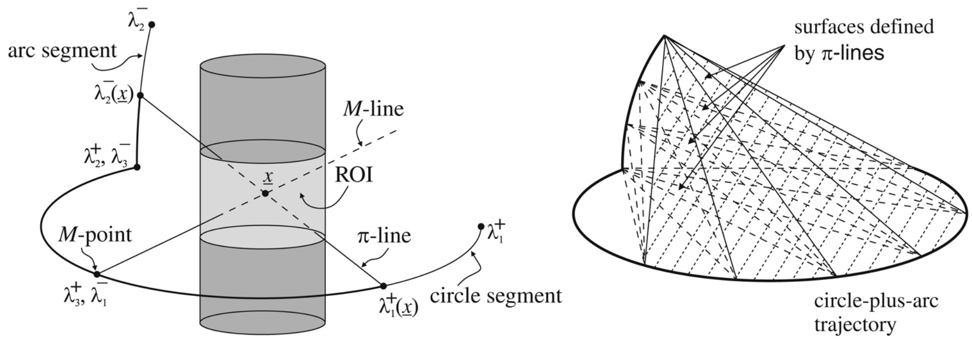

Fig. 3.

(*Left*) For each point $\underline{x}$ inside the object, there is one associated *M*-line and one associated π-line. The *M*-line defines the filtering directions for $\underline{x}$. The π-line delimitates the portion of the trajectory from where the data must be backprojected onto $\underline{x}$, to reconstruct $f(\underline{x})$ accurately. (*Right*) The volume covered by the union of surfaces (see hatched areas) defined by π-lines connecting one source position on the arc segment to all source positions on the circle segment can be reconstructed accurately; see Katsevich [2] for details

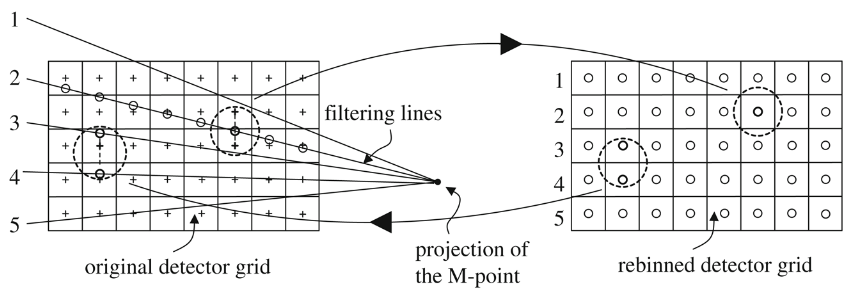

Fig. 4.

During forward rebinning, the values on the rebinned detector grid (*right*) are generated by linear interpolation from the values on the original detector grid (*left*). During backward rebinning, the values on the original detector grid (*left*) are generated by linear interpolation from the values on the rebinned detector grid (*right*). The filtering lines are sampled such that interpolations need to be done only in vertical direction. In horizontal direction, both detector grids coincide. The figure shows a case with five filtering lines. As illustrated with the *top arrow*, the detector values that are at the two positions (crosses '+') within the *dashed circle* on the *left* are interpolated to produce the rebinned detector value within the *dashed circle* on the *right*. Vice versa, the bottom arrow shows that the rebinned detector values at the two positions (circles 'o') within the dashed circle on the right are interpolated to produce the detector value within the *dashed circle* on the *left*

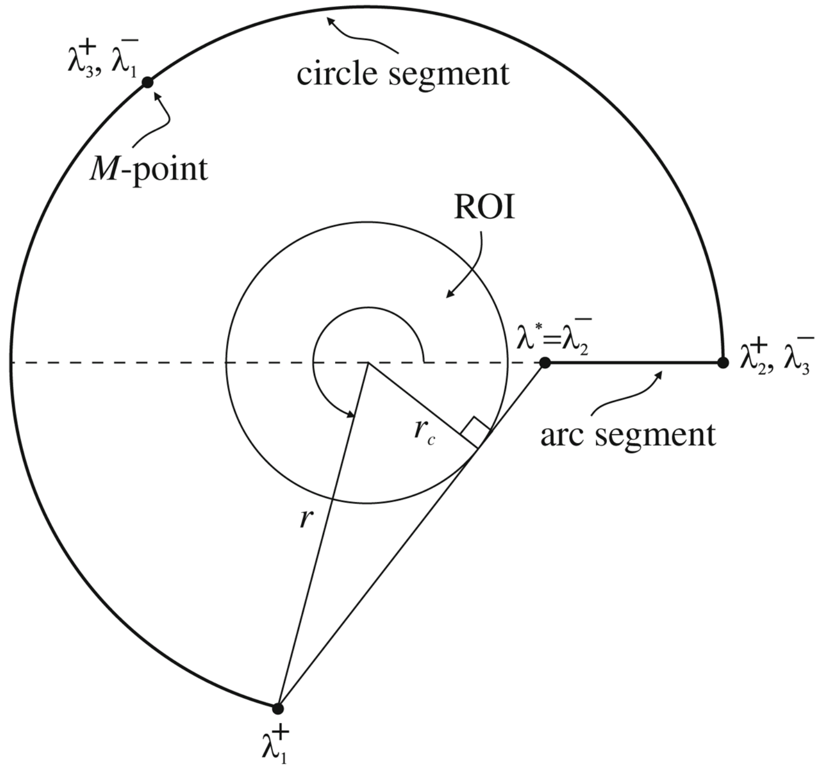

Fig. 5.

Top view onto the circle-plus-arc trajectory. The angular range λ_1^+ of the circle segment is determined by the orthogonal projection of the line, passing through the highest point on the arc segment (the one corresponding to λ_2^-) and being tangent to the ROI, onto the plane of the circle segment

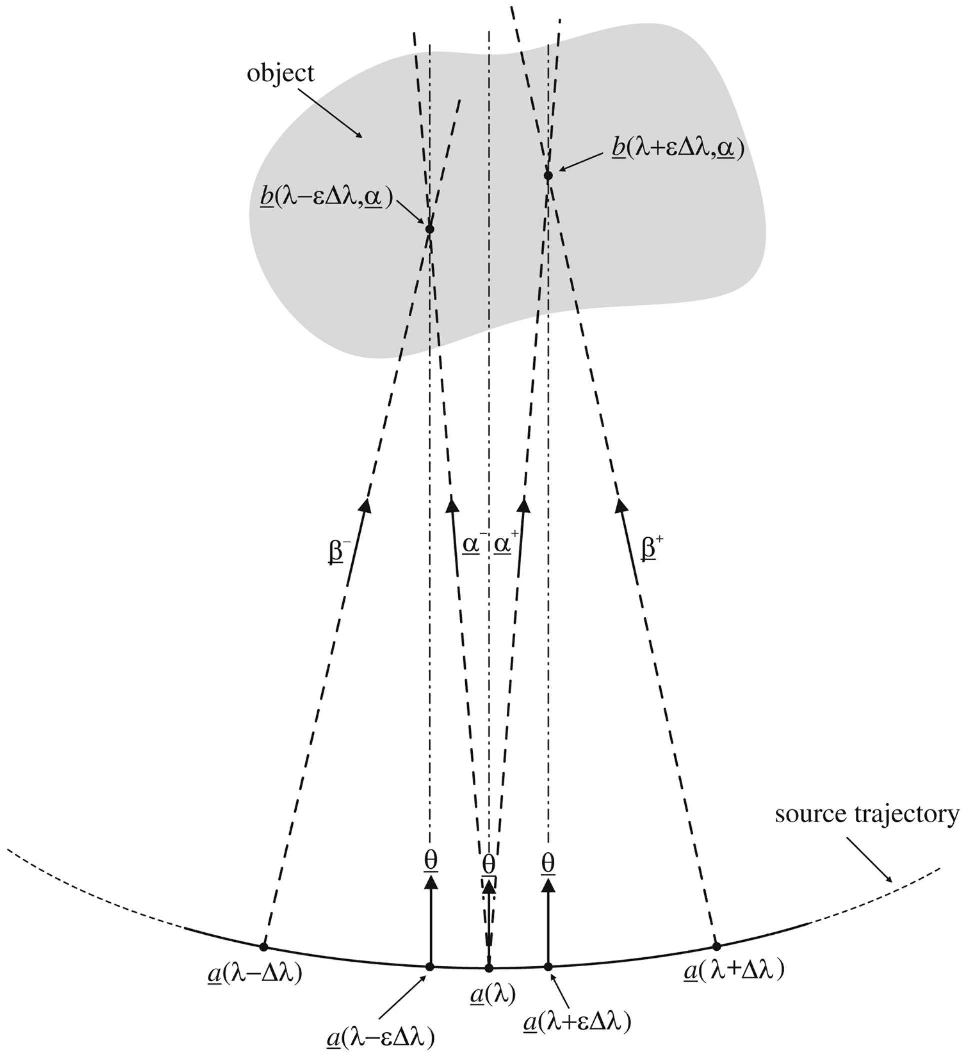

Fig. 6.
The new scheme to compute the view-dependent derivative at position λ on the circle segment of the circle-plus-arc trajectory; see also Noo et al. [19] for further details

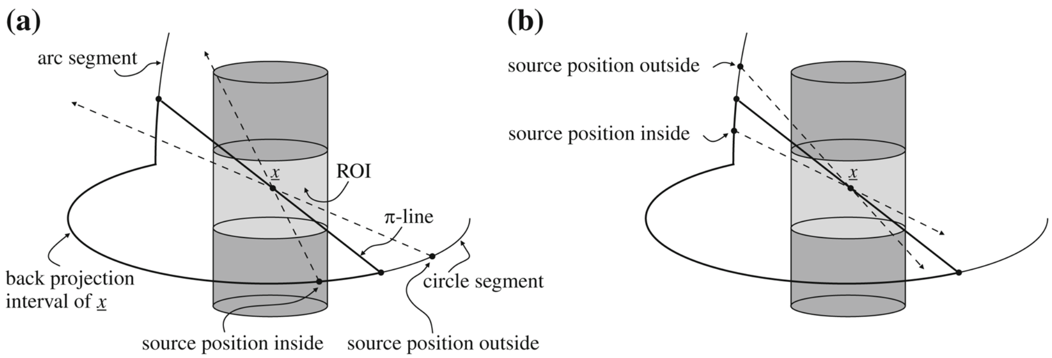

Fig. 7.

Data selection principle for **a** the circle segment and **b** the arc segment. **a** If a source position belongs to the backprojection interval of $\underline{x}$ (shown in *bold*), $\underline{x}$ projects on the right-hand side of the arc segment. Otherwise, $\underline{x}$ projects on the left-hand side of the arc segment. **b** If a source position belongs to the backprojection interval of $\underline{x}$, $\underline{x}$ projects above the circle segment. Otherwise, $\underline{x}$ projects below the circle segment

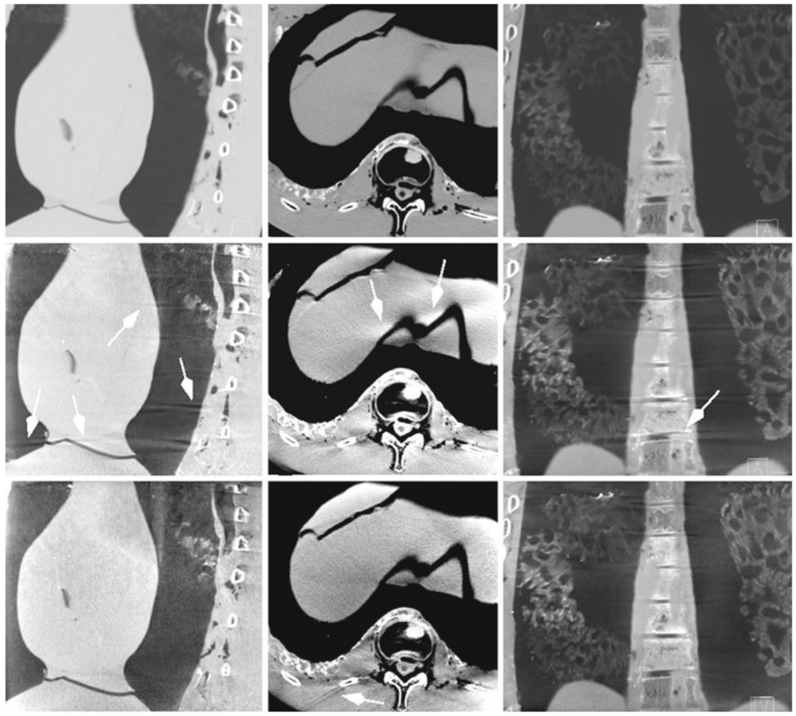

Fig. 8.

Sagittal (*left*), axial (*middle*), and coronal (*right*) slices through the thorax phantom. *Top row*: CT image to be used as ground truth. *Middle row*: FDK reconstruction from the data on the circle segment. *Bottom row*: M-line reconstruction from the circle-and-arc data. In the first column, the white arrows in the FDK image highlight cone-beam artifacts in the form of streaks that are caused by sharp variations in z and that have been eliminated using the circle-and-arc data with the M-line reconstruction technique. In the second column, the *arrows* in the FDK image highlight low-frequency cone-beam artifacts that have also been eliminated, whereas the *arrow* in the M-line image highlights a streak artifact due to discretization errors that does not appear in the FDK image. In the third column, the *arrow* in the FDK image highlights a cone-beam artifact that affects the analysis of the material between the bottom two vertebrae and has been eliminated; notice again in the same image the numerous streak artifacts that originate from the spine

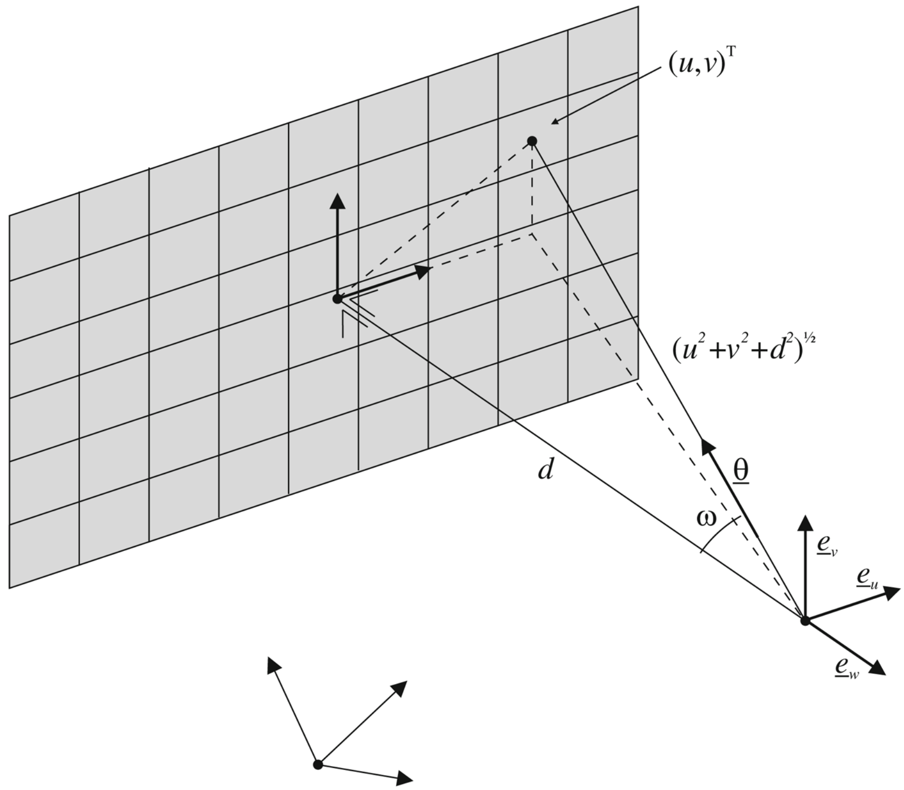

Fig. 9.

The factor $\frac{d}{\sqrt{u^2+v^2+d^2}}$ weights each detector pixel value to normalize the distance between the detector pixel and the X-ray source position. This factor is equivalent to the cosine of the angle ω

Table 1

Parameters used for the experiments

	C-arm parameters
Radius (circle/arc) (r) [mm]	750
Source-detector distance (d) [mm]	1200
Pixel width (Δu) [mm/pixel]	0.308
Pixel height (Δv) [mm/pixel]	0.308
Detector dimension [pixel^2]	1240 × 960
Angular sampling (circle) [°/projection]	0.4
Angular sampling (arc) [°/projection]	0.4
Number of projections (circle)	543 (216.8°)
Number of projections (arc)	109 (43.2°)